

SINUMERIK

SINUMERIK 840D sl Sonderfunktionen

Funktionshandbuch

Gültig für

Steuerung
SINUMERIK 840D sl / 840DE sl

CNC-Software Version 4.91

12/2018
A5E40874099A AB

Vorwort

Grundlegende Sicherheitshinweise	1
F2: Mehrachstransformationen	2
G1: Gantry-Achsen	3
K6: Konturtunnel- Überwachung	4
K7: Kinematische Kette	5
K8: Geometrische Maschinenmodellierung	6
K9: Kollisionsvermeidung, intern	7
K11 Kollisionsvermeidung, extern	8
K12 Transformationsdefinitionen mit kinematischen Ketten	9
M3: Achskopplungen	10
R3: Erweitertes Stillsetzen und Rückziehen	11
S9: Sollwertumschaltung - nur 840D sl	12
T3: Tangentialsteuerung - nur 840D sl	13
T4: Automatisches Nachoptimieren mit AST (Option) - nur 840D sl	14
T5: Intelligente Lastanpassung (Option) - nur 840D sl	15

Fortsetzung nächste Seite

SINUMERIK 840D sl Sonderfunktionen

Funktionshandbuch


Fortsetzung


TE01: Installation und Aktivierung ladbarer Compile-Zyklen	16
TE02: Simulation von Compile-Zyklen (nur HMI Advanced)	17
TE1: Abstandsregelung - nur 840D sl	18
TE3: Drehzahl-/ Drehmomentkopplung, Master-Slave	19
TE4: Transformationspaket Handling - nur 840D sl	20
TE6: MKS-Kopplung - nur 840D sl	21
TE7: Wiederaufsetzen (Retrace Support) - nur 840D sl	22
TE8: Taktunabhängige bahnsynchrone Schaltsignalausgabe - nur 840D sl	23
TE9: Achspaar-Kollisionsschutz	24
V2: Vorverarbeitung	25
W5: 3D-Werkzeugradiuskorrektur - nur 840D sl	26
W6: Weglängenauswertung - nur 840D sl	27
Z3: NC/PLC-Nahtstellensignale	28
Anhang	A


Rechtliche Hinweise

Warnhinweiskonzept

Dieses Handbuch enthält Hinweise, die Sie zu Ihrer persönlichen Sicherheit sowie zur Vermeidung von Sachschäden beachten müssen. Die Hinweise zu Ihrer persönlichen Sicherheit sind durch ein Warndreieck hervorgehoben, Hinweise zu alleinigen Sachschäden stehen ohne Warndreieck. Je nach Gefährdungsstufe werden die Warnhinweise in abnehmender Reihenfolge wie folgt dargestellt.

 GEFAHR
bedeutet, dass Tod oder schwere Körperverletzung eintreten wird , wenn die entsprechenden Vorsichtsmaßnahmen nicht getroffen werden.

 WARNUNG
bedeutet, dass Tod oder schwere Körperverletzung eintreten kann , wenn die entsprechenden Vorsichtsmaßnahmen nicht getroffen werden.

 VORSICHT
bedeutet, dass eine leichte Körperverletzung eintreten kann, wenn die entsprechenden Vorsichtsmaßnahmen nicht getroffen werden.

ACHTUNG
bedeutet, dass Sachschaden eintreten kann, wenn die entsprechenden Vorsichtsmaßnahmen nicht getroffen werden.


Beim Auftreten mehrerer Gefährdungsstufen wird immer der Warnhinweis zur jeweils höchsten Stufe verwendet. Wenn in einem Warnhinweis mit dem Warndreieck vor Personenschäden gewarnt wird, dann kann im selben Warnhinweis zusätzlich eine Warnung vor Sachschäden angefügt sein.

Qualifiziertes Personal

Das zu dieser Dokumentation zugehörige Produkt/System darf nur von für die jeweilige Aufgabenstellung **qualifiziertem Personal** gehandhabt werden unter Beachtung der für die jeweilige Aufgabenstellung zugehörigen Dokumentation, insbesondere der darin enthaltenen Sicherheits- und Warnhinweise. Qualifiziertes Personal ist auf Grund seiner Ausbildung und Erfahrung befähigt, im Umgang mit diesen Produkten/Systemen Risiken zu erkennen und mögliche Gefährdungen zu vermeiden.

Bestimmungsgemäßer Gebrauch von Siemens-Produkten

Beachten Sie Folgendes:

 WARNUNG
Siemens-Produkte dürfen nur für die im Katalog und in der zugehörigen technischen Dokumentation vorgesehenen Einsatzfälle verwendet werden. Falls Fremdprodukte und -komponenten zum Einsatz kommen, müssen diese von Siemens empfohlen bzw. zugelassen sein. Der einwandfreie und sichere Betrieb der Produkte setzt sachgemäßen Transport, sachgemäße Lagerung, Aufstellung, Montage, Installation, Inbetriebnahme, Bedienung und Instandhaltung voraus. Die zulässigen Umgebungsbedingungen müssen eingehalten werden. Hinweise in den zugehörigen Dokumentationen müssen beachtet werden.

Marken

Alle mit dem Schutzrechtsvermerk ® gekennzeichneten Bezeichnungen sind eingetragene Marken der Siemens AG. Die übrigen Bezeichnungen in dieser Schrift können Marken sein, deren Benutzung durch Dritte für deren Zwecke die Rechte der Inhaber verletzen kann.

Haftungsausschluss

Wir haben den Inhalt der Druckschrift auf Übereinstimmung mit der beschriebenen Hard- und Software geprüft. Dennoch können Abweichungen nicht ausgeschlossen werden, so dass wir für die vollständige Übereinstimmung keine Gewähr übernehmen. Die Angaben in dieser Druckschrift werden regelmäßig überprüft, notwendige Korrekturen sind in den nachfolgenden Auflagen enthalten.

Vorwort

SINUMERIK-Dokumentation

Die SINUMERIK-Dokumentation ist in folgende Kategorien gegliedert:

- Allgemeine Dokumentation/Kataloge
- Anwender-Dokumentation
- Hersteller-/Service-Dokumentation

Weiterführende Informationen

Unter folgender Adresse (<https://support.industry.siemens.com/cs/de/de/view/108464614>) finden Sie Informationen zu den Themen:

- Dokumentation bestellen/Druckschriftenübersicht
- Weiterführende Links für den Download von Dokumenten
- Dokumentation online nutzen (Handbücher/Informationen finden und durchsuchen)

Bei Fragen zur technischen Dokumentation (z. B. Anregungen, Korrekturen) senden Sie eine E-Mail an folgende Adresse (<mailto:docu.motioncontrol@siemens.com>).

mySupport/Dokumentation

Unter folgender Adresse (<https://support.industry.siemens.com/My/ww/de/documentation>) finden Sie Informationen, wie Sie Ihre Dokumentation auf Basis der Siemensinhalte individuell zusammenstellen und für die eigene Maschinendokumentation anpassen.

Training

Unter folgender Adresse (<http://www.siemens.de/sitrain>) finden Sie Informationen zu SITRAIN - dem Training von Siemens für Produkte, Systeme und Lösungen der Antriebs- und Automatisierungstechnik.

FAQs

Frequently Asked Questions finden Sie in den Service&Support-Seiten unter Produkt Support (<https://support.industry.siemens.com/cs/de/de/ps/faq>).

SINUMERIK

Informationen zu SINUMERIK finden Sie unter folgender Adresse (<http://www.siemens.de/sinumerik>).

Zielgruppe

Die vorliegende Druckschrift wendet sich an:

- Projekteure
- Technologen (von Maschinenherstellern)
- Inbetriebnehmer (von Systemen/Maschinen)
- Programmierer

Nutzen

Das Funktionshandbuch beschreibt die Funktionen, so dass die Zielgruppe die Funktionen kennt und auswählen kann. Es befähigt die Zielgruppe, die Funktionen in Betrieb zu nehmen.

Standardumfang

In der vorliegenden Dokumentation ist die Funktionalität des Standardumfangs beschrieben. Ergänzungen oder Änderungen, die durch den Maschinenhersteller vorgenommen werden, werden vom Maschinenhersteller dokumentiert.

Es können in der Steuerung weitere, in dieser Dokumentation nicht erläuterte Funktionen ablauffähig sein. Es besteht jedoch kein Anspruch auf diese Funktionen bei der Neulieferung bzw. im Servicefall.

Ebenso enthält diese Dokumentation aus Gründen der Übersichtlichkeit nicht sämtliche Detailinformationen zu allen Typen des Produkts und kann auch nicht jeden denkbaren Fall der Aufstellung, des Betriebes und der Instandhaltung berücksichtigen.

Hinweis zur Datenschutzgrundverordnung

Siemens beachtet die Grundsätze des Datenschutzes, insbesondere die Gebote der Datenminimierung (privacy by design). Für dieses Produkt bedeutet dies:

Das Produkt verarbeitet/speichert keine personenbezogenen Daten, lediglich technische Funktionsdaten (z. B. Zeitstempel). Verknüpft der Anwender diese Daten mit anderen Daten (z. B. Schichtpläne) oder speichert er personenbezogene Daten auf dem gleichen Medium (z. B. Festplatte) und stellt so einen Personenbezug her, hat er die Einhaltung der datenschutzrechtlichen Vorgaben selbst sicherzustellen.

Technical Support

Landesspezifische Telefonnummern für technische Beratung finden Sie im Internet unter folgender Adresse (<https://support.industry.siemens.com/sc/ww/de/sc/2090>) im Bereich "Kontakt".

Informationen zu Struktur und Inhalt

Aufbau

Das vorliegende Funktionshandbuch ist wie folgt aufgebaut:

- Innentitel (Seite 3) mit dem Titel des Funktionshandbuchs, den SINUMERIK-Steuerungen sowie der Software und Version, für die diese Ausgabe des Funktionshandbuchs gültig ist, und der Übersicht der einzelnen Funktionsbeschreibungen.
- Funktionsbeschreibungen in der Reihenfolge ihrer alphanumerischen Kurzzeichen (z. B. A2, A3, B1 etc.)
- Anhang mit:
 - Abkürzungsverzeichnis
 - Dokumentationsübersicht
- Begriffsverzeichnis

Hinweis

Ausführliche Daten- und Alarm-Beschreibungen siehe:

- für Maschinen- und Settingdaten:
Ausführliche Maschinendaten-Beschreibung
 - für NC/PLC-Nahtstellensignale:
Listenhandbuch NC-Variablen und Nahtstellensignale
 - für Alarme:
Diagnosehandbuch
-

Schreibweise von Systemdaten

Für Systemdaten gelten in dieser Dokumentation folgende Schreibweisen:

Signal/Datum	Schreibweise	Beispiel
NC/PLC-Nahtstellensignale	... NC/PLC-Nahtstellensignal: <Signaladresse> (<Signalname>)	Ist die neue Getriebestufe eingelegt, dann werden vom PLC-Programm die folgenden NC/PLC-Nahtstellensignale gesetzt: DB31, ... DBX16.0-2 (Istgetriebestufe A bis C) DB31, ... DBX16.3 (Getriebe ist umgeschaltet)
Maschinendatum	... Maschinendatum: <Typ><Nummer> <Vollständiger Bezeichner> (<Bedeutung>)	Masterspindel ist die abgelegte Spindel im Maschinendatum: MD20090 \$MC_SPIND_DEF_MASTER_SPIND (Löschstellung Masterspindel im Kanal)
Settingdatum	... Settingdatum: <Typ><Nummer> <Vollständiger Bezeichner> (<Bedeutung>)	Die logische Masterspindel ist enthalten im Settingdatum: SD42800 \$SC_SPIND_ASSIGN_TAB[0] (Spindelnummernumsetzer)

Hinweis**Signaladresse**

Die Funktionsbeschreibungen enthalten als <Signaladresse> eines NC/PLC-Nahtstellensignals nur die für SINUMERIK 840D sl gültige Adresse. Die Signaladresse für SINUMERIK 828D ist den Datenlisten "Signale an/von ..." am Ende der jeweiligen Funktionsbeschreibung zu entnehmen.

Mengengerüst

Erläuterungen bezüglich der NC/PLC-Nahtstelle gehen von der absoluten maximalen Anzahl folgender Komponenten aus:

- Betriebsartengruppen (DB11)
- Kanäle (DB21, ...)
- Achsen/Spindeln (DB31, ...)

Datentypen

In der Steuerung stehen zur Programmierung in Teileprogrammen folgenden Datentypen zur Verfügung:

Typ	Bedeutung	Wertebereich
INT	Ganzzahlige Werte mit Vorzeichen	-2.147.483.648 ... +2.147.483.647
REAL	Zahlen mit Dezimalpunkt	$\approx \pm 5,0 \cdot 10^{-324} \dots \approx \pm 1,7 \cdot 10^{308}$
BOOL	Wahrheitswerte	TRUE ($\neq 0$), FALSE (0)
CHAR	Zeichen ASCII bzw. Byte	0 ... 255 bzw. -128 ... 127
STRING	Zeichenkette, Null-terminiert	maximal 400 Zeichen + /0 (keine Sonderzeichen)
AXIS	Achsennamen	alle in der Steuerung vorhandenen Achsnamen
FRAME	geometrische Angaben für Verschieben, Drehen, Skalieren, Spiegeln	---

Felder

Felder können nur aus gleichen elementaren Datentypen gebildet werden. Es sind maximal 3-dimensionale Arrays möglich.

Beispiel: `DEF INT FELD[2, 3, 4]`

Zahlensysteme

Es stehen folgende Zahlensysteme zur Verfügung:

- Dezimal: `DEF INT Zahl = 1234` oder `DEF REAL Zahl = 1234.56`
- Hexadezimal: `DEF INT Zahl = 'H123ABC'`
- Binär: `DEF INT Zahl = 'B10001010010'`

Abfrage von REAL-Variablen

Es wird empfohlen, die Abfragen von REAL- bzw. DOUBLE-Variablen in NC-Programmen und Synchronaktionen als Grenzwertbetrachtung zu programmieren.

Beispiel: Abfrage des Istwerts einer Achse auf einen bestimmten Wert

Programmcode	Kommentar
DEF REAL AXPOS = 123.456	
IF (\$VA_IM[<Achse>] - 1ex-6) <= AXPOS <= (\$VA_IM[<Achse>] + 1ex-6)	; Istposition
...	== AXPOS
ELSE	
...	<> AXPOS
ENDIF	

Inhaltsverzeichnis

	Vorwort	5
1	Grundlegende Sicherheitshinweise	31
1.1	Allgemeine Sicherheitshinweise.....	31
1.2	Gewährleistung und Haftung für Applikationsbeispiele.....	31
1.3	Industrial Security.....	32
2	F2: Mehrachstransformationen	35
2.1	Kurzbeschreibung.....	35
2.1.1	Allgemeine Vorgaben.....	35
2.1.2	5-Achs-Transformation.....	35
2.1.3	3- und 4-Achs-Transformation.....	37
2.1.4	Orientierungstransformation mit schwenkbarer Linearachse.....	39
2.1.5	Kardanischer Fräskopf.....	40
2.1.6	Orientierungsachsen.....	41
2.1.7	Kartesisches manuelles Verfahren.....	42
2.1.8	Kartesisches PTP-Fahren.....	42
2.1.9	Generische 5-Achstransformation.....	42
2.1.10	Online-Werkzeuglängenkorrektur.....	43
2.1.11	Aktivierung über Teileprogramm/Softkey.....	43
2.1.12	Komprimierung der Orientierung.....	43
2.2	5-Achs-Transformation.....	44
2.2.1	Kinematische Transformation.....	44
2.2.2	Maschinentypen für 5-Achs-Transformation.....	45
2.2.3	Konfiguration einer Maschine für 5-Achs-Transformation.....	46
2.2.4	Werkzeugorientierung.....	51
2.2.5	Singuläre Stellen und ihre Behandlung.....	56
2.3	3- und 4-Achs-Transformationen.....	58
2.4	Transformation mit geschwenkter Linearachse.....	60
2.5	Kardanischer Fräskopf.....	66
2.5.1	Grundlagen kardanischer Fräskopf.....	66
2.5.2	Parametrierung.....	68
2.5.3	Verfahren des kardanischen Fräskopfes im JOG-Betrieb.....	69
2.6	Programmierung der 3- bis 5-Achs-Transformation.....	70
2.7	Generische 5-Achs-Transformation und Varianten.....	71
2.7.1	Funktionalität.....	71
2.7.2	Beschreibung der Maschinenkinematik.....	72
2.7.3	Generische Orientierungstransformations-Varianten.....	73
2.7.4	Daten orientierbarer Werkzeugträger parametrieren.....	75
2.7.5	Erweiterung der Generischen Transformation auf 6 Achsen - nur 840D sl.....	79
2.7.6	Erweiterung der Generischen Transformation auf 7 Achsen - nur 840D sl.....	82
2.7.7	Kartesisches Handverfahren bei Generischer Transformation.....	87

2.8	Einschränkungen für Kinematiken und Interpolationen.....	88
2.8.1	Singularitäten der Orientierung.....	89
2.9	Orientierung.....	91
2.9.1	Grundorientierung.....	91
2.9.2	Orientierungsbewegungen mit Achsgrenzen.....	93
2.9.3	Komprimierung der Orientierung.....	94
2.9.4	Glättung des Orientierungsverlaufs.....	99
2.9.4.1	Funktion.....	99
2.9.4.2	Inbetriebnahme.....	99
2.9.4.3	Glättung des Orientierungsverlaufs ein-/ausschalten (ORISON, ORISOF).....	100
2.9.5	Bahnrelative Orientierung (ORIPATH, ORIPATHS, ORIROTC).....	101
2.9.6	Programmierung von Orientierungspolynomen.....	107
2.9.7	Systemvariable für Werkzeugorientierung.....	111
2.10	Orientierungsachsen.....	113
2.10.1	Betriebsart JOG.....	115
2.10.2	Programmierung bei Orientierungstransformation.....	116
2.10.3	Programmierbarer Offset für Orientierungsachsen.....	118
2.10.4	Orientierungstransformation und orientierbare Werkzeugträger.....	119
2.10.5	Modulo-Anzeige von Orientierungsachsen.....	120
2.11	Orientierungsvektoren.....	121
2.11.1	Polynominterpolation von Orientierungsvektoren.....	121
2.11.2	Drehungen des Orientierungsvektors.....	125
2.11.3	Erweiterte Interpolationen von Orientierungen.....	129
2.12	Online-Werkzeuglängenkorrektur.....	133
2.13	Beispiele.....	137
2.13.1	Beispiel für eine 5-Achs-Transformation.....	137
2.13.2	Beispiel für eine 3- und 4-Achs-Transformation.....	140
2.13.2.1	Beispiel für eine 3-Achs-Transformation.....	140
2.13.2.2	Beispiel für eine 4-Achs-Transformation.....	140
2.13.3	Beispiel Kardanischer Fräskopf.....	141
2.13.4	Beispiel für Orientierungsachsen.....	142
2.13.5	Beispiele zu Orientierungsvektoren.....	144
2.13.5.1	Beispiel für Polynominterpolation von Orientierungsvektoren.....	144
2.13.5.2	Beispiel für Drehungen des Orientierungsvektors.....	145
2.13.6	Beispiele für generische Achstransformationen.....	146
2.13.6.1	Beispiel für eine Generische 6-Achs-Transformation.....	147
2.13.6.2	Beispiel für eine generische 7-Achs-Transformation.....	148
2.13.6.3	Beispiel für die Modifikation der Rundachsbewegung.....	149
2.14	Datenlisten.....	150
2.14.1	Maschinendaten.....	150
2.14.1.1	Allgemeine Maschinendaten.....	150
2.14.1.2	Kanal-spezifische Maschinendaten.....	150
2.14.2	Settingdaten.....	154
2.14.2.1	Allgemeine Settingdaten.....	154
2.14.2.2	Kanal-spezifische Settingdaten.....	154
2.14.3	Signale.....	155
2.14.3.1	Signale an Kanal.....	155
2.14.3.2	Signale von Kanal.....	155

3	G1: Gantry-Achsen.....	157
3.1	Kurzbeschreibung.....	157
3.2	Funktion "Gantry-Achsen".....	158
3.2.1	Definition eines Gantry-Verbundes.....	158
3.2.2	Überwachung der Synchronlaufdifferenz.....	159
3.2.3	Erweiterte Überwachung der Synchronlaufdifferenz.....	160
3.2.4	Referenzieren und Synchronisieren der Gantry-Achsen.....	160
3.2.5	Regeldynamik.....	161
3.2.6	Gantry-Verbund auftrennen.....	161
3.3	Referenzieren und Synchronisieren von Gantry-Achsen.....	162
3.3.1	Einführung.....	162
3.3.2	Automatisches Synchronisieren.....	168
3.3.3	Besonderheiten.....	169
3.4	Inbetriebnahme der Gantry-Achsen.....	172
3.5	Parametrierung: Verhalten im Fehlerfall.....	177
3.6	PLC-Nahtstellensignale bei Gantry-Achsen.....	177
3.7	Sonstiges bei Gantry-Achsen.....	179
3.8	Beispiele.....	182
3.8.1	Gantry-Verband erstellen.....	182
3.8.2	Einstellung der NC-PLC Nahtstelle.....	183
3.8.3	Beginn der Inbetriebnahme.....	184
3.8.4	Warn- und Fehlergrenzen einstellen.....	185
3.9	Datenlisten.....	187
3.9.1	Maschinendaten.....	187
3.9.1.1	Achs-/Spindel-spezifische Maschinendaten.....	187
3.9.2	Signale.....	188
3.9.2.1	Signale von BAG.....	188
3.9.2.2	Signale von Kanal.....	188
3.9.2.3	Signale an Achse/Spindel.....	188
3.9.2.4	Signale von Achse/Spindel.....	188
4	K6: Konturtunnel-Überwachung.....	191
4.1	Kurzbeschreibung.....	191
4.1.1	Konturtunnel-Überwachung - nur 840D sl.....	191
4.1.2	Programmierbare Konturgenauigkeit.....	192
4.2	Konturtunnel-Überwachung - nur 840D sl.....	193
4.3	Programmierbare Konturgenauigkeit.....	194
4.4	Randbedingungen.....	198
4.5	Datenlisten.....	198
4.5.1	Maschinendaten.....	198
4.5.1.1	Kanal-spezifische Maschinendaten.....	198
4.5.1.2	Achs-/Spindel-spezifische Maschinendaten.....	198
4.5.2	Settingdaten.....	199
4.5.2.1	Kanal-spezifische Settingdaten.....	199

5	K7: Kinematische Kette.....	201
5.1	Funktionsbeschreibung.....	201
5.1.1	Merkmale.....	201
5.2	Inbetriebnahme.....	205
5.2.1	Allgemein.....	205
5.2.1.1	Übersicht.....	205
5.2.1.2	Aufbau der Systemvariablen.....	206
5.2.2	Maschinendaten.....	207
5.2.2.1	Maximale Anzahl Elemente.....	207
5.2.2.2	Root-Element.....	207
5.2.2.3	Maximale Anzahl Schalter.....	207
5.2.3	Systemvariablen.....	208
5.2.3.1	Übersicht.....	208
5.2.3.2	\$NK_NAME.....	209
5.2.3.3	\$NK_NEXT.....	210
5.2.3.4	\$NK_PARALLEL.....	210
5.2.3.5	\$NK_TYPE.....	211
5.2.3.6	Typabhängige Variablen bei \$NK_TYPE = "AXIS_LIN".....	212
5.2.3.7	Typabhängige Variablen bei \$NK_TYPE = "AXIS_ROT".....	215
5.2.3.8	Typabhängige Variablen bei \$NK_TYPE = "ROT_CONST".....	219
5.2.3.9	Typabhängige Variablen bei \$NK_TYPE = "OFFSET".....	221
5.2.3.10	Typabhängige Variablen bei \$NK_TYPE = "SWITCH".....	222
5.2.3.11	\$NK_SWITCH.....	224
5.3	Programmierung.....	225
5.3.1	Löschen von Komponenten (DELOBJ).....	225
5.3.2	Indexermittlung per Namen (NAMETOINT).....	228
5.4	Beispiel.....	229
5.4.1	Vorgaben.....	229
5.4.2	Teileprogramm des Maschinenmodells.....	231
5.5	Datenlisten.....	234
5.5.1	Maschinendaten.....	234
5.5.1.1	NC-spezifische Maschinendaten.....	234
5.5.2	Systemvariablen.....	234
6	K8: Geometrische Maschinenmodellierung.....	235
6.1	Funktionsbeschreibung.....	235
6.1.1	Merkmale.....	235
6.1.2	Automatische Werkzeugschutzbereiche.....	238
6.2	Inbetriebnahme.....	241
6.2.1	Allgemein.....	241
6.2.1.1	Übersicht.....	241
6.2.1.2	Aufbau der Systemvariablen.....	241
6.2.1.3	Farbtafel.....	243
6.2.2	Maschinendaten.....	244
6.2.2.1	Maximale Anzahl von Schutzbereichen.....	244
6.2.2.2	Maximale Anzahl von Schutzbereichselementen für Maschinenschutzbereiche.....	244
6.2.2.3	Maximale Anzahl von Schutzbereichselementen für automatische Werkzeugschutzbereiche.....	244

6.2.2.4	Maximale Anzahl von NC/PLC-Nahtstellensignalen zur Voraktivierung von Schutzbereichen.....	244
6.2.2.5	Maximale Anzahl von Dreiecken für Maschinenschutzbereiche.....	244
6.2.2.6	Maximale Anzahl von Dreiecken für automatisch Werkzeugschutzbereiche.....	245
6.2.2.7	Erzeugungsmodus für automatische Werkzeugschutzbereiche.....	245
6.2.3	Systemvariablen: Schutzbereiche.....	245
6.2.3.1	Übersicht.....	245
6.2.3.2	\$NP_PROT_NAME.....	246
6.2.3.3	\$NP_CHAIN_ELEM.....	247
6.2.3.4	\$NP_PROT_TYPE.....	248
6.2.3.5	\$NP_1ST_PROT.....	249
6.2.3.6	\$NP_PROT_COLOR.....	250
6.2.3.7	\$NP_PROT_D_LEVEL.....	252
6.2.3.8	\$NP_BIT_NO.....	253
6.2.3.9	\$NP_INIT_STAT.....	254
6.2.3.10	\$NP_INDEX.....	255
6.2.4	Systemvariablen: Schutzbereichselemente für Maschinenschutzbereiche.....	257
6.2.4.1	Übersicht.....	257
6.2.4.2	\$NP_NAME.....	258
6.2.4.3	\$NP_NEXT.....	259
6.2.4.4	\$NP_NEXTP.....	260
6.2.4.5	\$NP_COLOR.....	261
6.2.4.6	\$NP_D_LEVEL.....	263
6.2.4.7	\$NP_USAGE.....	264
6.2.4.8	\$NP_TYPE.....	265
6.2.4.9	\$NP_FILENAME.....	269
6.2.4.10	\$NP_PARA.....	274
6.2.4.11	\$NP_OFF.....	275
6.2.4.12	\$NP_DIR.....	276
6.2.4.13	\$NP_ANG.....	278
6.2.5	Systemvariablen: Schutzbereichselemente für automatische Werkzeugschutzbereiche ...	279
6.2.6	Randbedingungen.....	280
6.3	Datenlisten.....	281
6.3.1	Maschinendaten.....	281
6.3.1.1	NC-spezifische Maschinendaten.....	281
6.3.2	Systemvariablen.....	281
7	K9: Kollisionsvermeidung, intern.....	283
7.1	Funktionsbeschreibung.....	283
7.1.1	Optionen.....	283
7.1.2	Merkmale.....	283
7.1.3	Reaktion der Steuerung bei Kollisionsgefahr.....	285
7.1.4	Zustandsdiagramm: Schutzbereich.....	288
7.1.5	Werkzeuge.....	289
7.1.6	Randbedingungen.....	292
7.2	Inbetriebnahme.....	294
7.2.1	Allgemein.....	294
7.2.1.1	Übersicht.....	294
7.2.1.2	Aufbau der Systemvariablen.....	294
7.2.2	Maschinendaten.....	295
7.2.2.1	Kollisionstoleranz.....	295

7.2.2.2	Sicherheitsabstand.....	296
7.2.2.3	Maximaler Speicherplatz.....	296
7.2.2.4	Maximale Anzahl an Kollisionspaaren.....	297
7.2.2.5	Schutzstufen für Kollisionsvermeidung Ein/Aus.....	298
7.2.3	Systemvariablen.....	298
7.2.3.1	Übersicht.....	298
7.2.3.2	\$NP_COLL_PAIR.....	299
7.2.3.3	\$NP_SAFETY_DIST.....	300
7.2.4	Erweiternde Systemvariablen.....	301
7.2.4.1	Übersicht.....	301
7.2.4.2	Zustandsdaten.....	302
7.2.4.3	Speicherplatzbedarf.....	303
7.2.4.4	Bremswegschätzungen.....	303
7.3	Programmierung.....	304
7.3.1	Prüfen auf Kollisionspaar (COLLPAIR).....	304
7.3.2	Neuberechnung des Maschinenmodells der Kollisionsvermeidung anfordern (PROTA).....	305
7.3.3	Schutzbereichszustand setzen (PROTS).....	306
7.3.4	Abstandsbestimmung zweier Schutzbereiche (PROTD).....	307
7.4	Beispiel.....	309
7.4.1	Vorgaben.....	309
7.4.2	Teilprogramm des Maschinenmodells.....	313
7.5	Datenlisten.....	321
7.5.1	Maschinendaten.....	321
7.5.1.1	NC-spezifische Maschinendaten.....	321
7.5.2	Systemvariablen.....	321
7.5.3	Signale.....	322
7.5.3.1	Signale an NC.....	322
7.5.3.2	Signale von NC.....	322
7.5.3.3	Signale von Kanal.....	323
7.5.3.4	Signale von Achse.....	323
8	K11 Kollisionsvermeidung, extern.....	325
8.1	Funktionsbeschreibung.....	325
8.1.1	Optionen.....	325
8.1.2	Merkmale.....	325
8.2	Inbetriebnahme.....	325
8.2.1	Maschinendaten.....	325
8.2.1.1	Funktionen der externen Kollisionsvermeidung.....	325
8.2.1.2	Vorschauzeit.....	326
8.2.1.3	Vorschau-Zeitschritt.....	326
8.2.1.4	Timeout-Zeit.....	326
9	K12 Transformationsdefinitionen mit kinematischen Ketten.....	327
9.1	Funktionsbeschreibung.....	327
9.1.1	Merkmale.....	327
9.1.2	Definition kinematischer Transformationen.....	328
9.1.3	Dynamische Orientierungstransformation TRAORI_DYN.....	330
9.1.4	Stirnseitentransformation TRANSMIT.....	333
9.1.5	Zylindermanteltransformation TRACYL.....	337
9.1.6	Schiefwinkeltransformation (Schräge Achse) TRAANG_K.....	340

9.1.7	Statische Orientierungstransformation TRAORI_STAT.....	341
9.1.8	Transformationen verketteten (TRACON_K).....	342
9.1.9	Bei Reset wirksame Transformationen.....	343
9.1.10	Persistente Transformationen.....	343
9.1.11	Migration von kinematischen Transformationen.....	343
9.1.12	Frames bei kinematischen Transformationen.....	345
9.1.13	Werkzeuglängen.....	345
9.2	Inbetriebnahme.....	346
9.2.1	Allgemein.....	346
9.2.1.1	Übersicht.....	346
9.2.1.2	Aufbau der Systemvariablen.....	346
9.2.2	Maschinendaten.....	347
9.2.2.1	Maximale Anzahl von Transformationen mit Kinematischen Ketten.....	347
9.2.2.2	Name der Resettransformation.....	347
9.2.2.3	Aktivierungsgrenze der Echtzeitdynamiküberwachung (Linearachsen).....	348
9.2.2.4	Aktivierungsgrenze der Echtzeitdynamiküberwachung (Rundachsen).....	348
9.2.2.5	Korrekturwert für Offsetvektoren bei CORRTRAFO.....	348
9.2.2.6	Winkelabweichung für Richtungsvektoren bei CORRTRAFO.....	348
9.2.3	Verwendung der Systemvariablen bei kinematischen Transformationen.....	348
9.2.4	Systemvariablen für allgemeine Transformationstypen.....	349
9.2.4.1	Übersicht.....	349
9.2.4.2	\$NT_NAME.....	351
9.2.4.3	\$NT_TRAFO_INDEX.....	352
9.2.4.4	\$NT_TRAFO_TYPE.....	353
9.2.4.5	\$NT_T_CHAIN_FIRST_ELEM.....	354
9.2.4.6	\$NT_P_CHAIN_FIRST_ELEM.....	356
9.2.4.7	\$NT_T_CHAIN_LAST_ELEM.....	357
9.2.4.8	\$NT_P_CHAIN_LAST_ELEM.....	358
9.2.4.9	\$NT_T_REF_ELEM.....	359
9.2.4.10	\$NT_GEO_AX_NAME.....	360
9.2.4.11	\$NT_ROT_AX_NAME.....	361
9.2.4.12	\$NT_ROT_AX_OFFSET.....	363
9.2.4.13	\$NT_CLOSE_CHAIN_P.....	364
9.2.4.14	\$NT_CLOSE_CHAIN_T.....	365
9.2.4.15	\$NT_ROT_OFFSET_FROM_FRAME.....	366
9.2.4.16	\$NT_TRAFO_INCLUDES_TOOL.....	367
9.2.4.17	\$NT_AUX_POS.....	368
9.2.4.18	\$NT_IDENT.....	369
9.2.4.19	\$NT_CNTRL.....	370
9.2.4.20	\$NT_ROT_AX_CNT.....	373
9.2.4.21	\$NT_BASE_TOOL_COMP.....	374
9.2.5	Additive Systemvariablen für Orientierungstransformation.....	375
9.2.5.1	Übersicht.....	375
9.2.5.2	\$NT_BASE_ORIENT.....	376
9.2.5.3	\$NT_BASE_ORIENT_NORMAL.....	377
9.2.5.4	\$NT_ROT_AX_POS.....	378
9.2.5.5	\$NT_POLE_LIMIT.....	379
9.2.5.6	\$NT_POLE_TOL.....	380
9.2.5.7	\$NT_IGNORE_TOOL_ORIENT.....	381
9.2.5.8	\$NT_CORR_ELEM_T.....	382
9.2.5.9	\$NT_CORR_ELEM_P.....	383
9.2.6	Additive Systemvariablen für Stirnflächentransformationen (TRANSMIT).....	384

9.2.6.1	Übersicht.....	384
9.2.6.2	\$NT_POLE_SIDE_FIX.....	385
9.3	Programmierung.....	386
9.3.1	Transformation einschalten (TRAFOON).....	386
9.3.2	Orientierungstransformation nach Maschinenvermessung modifizieren (CORRTRAFO)....	387
9.4	Beispiele.....	394
9.4.1	Vorgaben für TRAORI_DYN.....	394
9.4.2	Teilprogramm für TRAORI_DYN.....	397
9.4.3	Teilprogramm für TRANSMIT.....	401
9.4.4	Teilprogramm für TRACYL.....	404
9.4.5	Teilprogramm für TRAANG.....	407
10	M3: Achskopplungen.....	409
10.1	Mitschleppen.....	409
10.1.1	Kurzbeschreibung.....	409
10.1.1.1	Funktion.....	409
10.1.1.2	Voraussetzungen.....	409
10.1.2	Allgemeine Funktionalität.....	410
10.1.3	Programmierung.....	414
10.1.3.1	Definition und Einschalten eines Mitschleppverbandes (TRAILON).....	414
10.1.3.2	Ausschalten (TRAILOF).....	414
10.1.4	Wirksamkeit der PLC-Nahtstellensignale.....	415
10.1.5	Kopplungsstatus.....	416
10.1.6	Dynamikbegrenzung.....	417
10.1.7	Randbedingungen.....	417
10.1.8	Beispiele.....	418
10.2	Kurventabellen - nur 840D sl.....	419
10.2.1	Kurzbeschreibung.....	419
10.2.1.1	Funktion.....	419
10.2.1.2	Voraussetzungen.....	420
10.2.2	Allgemeine Funktionalität.....	420
10.2.3	Speicherorganisation.....	421
10.2.4	Inbetriebnahme.....	423
10.2.4.1	Speicherkonfiguration.....	423
10.2.4.2	Werkzeugradiuskorrektur.....	424
10.2.4.3	Vorgabe des Speichertyps.....	424
10.2.5	Programmierung.....	425
10.2.6	Zugriff auf Tabellenpositionen und Tabellensegmente.....	431
10.2.7	Aktivierung/Deaktivierung.....	435
10.2.8	Sonderfall Modulo-Leitachse.....	436
10.2.9	Verhalten in den Betriebsarten AUTOMATIK, MDA, JOG.....	437
10.2.10	Wirksamkeit der PLC-Nahtstellensignale.....	437
10.2.11	Diagnose und Optimierung der Ressourcennutzung.....	437
10.2.12	Randbedingungen.....	441
10.2.13	Beispiele.....	441
10.3	Leitwertkopplung - nur 840D sl.....	443
10.3.1	Kurzbeschreibung.....	443
10.3.1.1	Funktion.....	443
10.3.1.2	Voraussetzungen.....	444
10.3.2	Allgemeine Funktionalität.....	444

10.3.3	Programmierung.....	447
10.3.4	Verhalten in den Betriebsarten AUTOMATIK, MDA, JOG.....	452
10.3.5	Wirksamkeit der PLC-Nahtstellensignale.....	454
10.3.6	Besonderheiten der Funktion Achs-Leitwertkopplung.....	454
10.3.7	Randbedingungen.....	455
10.4	Elektronisches Getriebe (EG).....	455
10.4.1	Kurzbeschreibung.....	455
10.4.1.1	Funktion.....	455
10.4.1.2	Voraussetzungen.....	456
10.4.2	Elektronisches Getriebe (EG).....	456
10.4.3	Definition eines EG-Achsverbandes.....	464
10.4.4	Einschalten eines EG-Achsverbandes.....	465
10.4.5	Ausschalten eines EG-Achsverbandes.....	470
10.4.6	Löschen eines EG-Achsverbandes.....	470
10.4.7	Zusammenwirken des Umdrehungsvorschubs (G95) mit dem Elektronischen Getriebe....	471
10.4.8	Verhalten bei POWER ON, RESET, BA-Wechsel, Suchlauf.....	471
10.4.9	Systemvariablen für Elektronisches Getriebe.....	472
10.4.10	Beispiele.....	473
10.4.10.1	Beispiel mit linearen Kopplungen.....	473
10.4.10.2	Erweitertes Beispiel mit nicht linearen Anteilen.....	477
10.5	Generische Kopplung.....	483
10.5.1	Kurzbeschreibung.....	483
10.5.1.1	Funktion.....	483
10.5.1.2	Voraussetzungen.....	484
10.5.2	Grundlagen.....	487
10.5.2.1	Koppelmodul.....	487
10.5.2.2	Schlüsselwörter und Kopplungseigenschaften.....	489
10.5.2.3	Systemvariablen.....	491
10.5.3	Koppelmodule anlegen/löschen.....	492
10.5.3.1	Koppelmodul anlegen (CPDEF).....	492
10.5.3.2	Koppelmodul löschen (CPDEL).....	493
10.5.3.3	Leitachsen definieren (CPLDEF bzw. CPDEF+CPLA).....	494
10.5.3.4	Leitachsen löschen (CPLDEL bzw. CPDEL+CPLA).....	495
10.5.4	Kopplung ein-/ausschalten.....	496
10.5.4.1	Koppelmodul einschalten (CPON).....	496
10.5.4.2	Koppelmodul ausschalten (CPOF).....	497
10.5.4.3	Leitachsen eines Koppelmoduls einschalten (CPLON).....	498
10.5.4.4	Leitachsen eines Koppelmoduls ausschalten (CPLOF).....	499
10.5.4.5	Implizites Anlegen und Löschen von Koppelmodulen.....	499
10.5.5	Kopplungseigenschaften programmieren.....	500
10.5.5.1	Koppelgesetz (CPLNUM, CPLDEN, CPLCTID).....	500
10.5.5.2	Kopplungsbezug (CPLSETVAL).....	502
10.5.5.3	Koordinatenbezug (CPFRS).....	503
10.5.5.4	Satzwechselverhalten (CPBC).....	504
10.5.5.5	Synchronposition der Folgeachse beim Einschalten (CPFPOS+CPON).....	506
10.5.5.6	Synchronposition der Leitachse beim Einschalten (CPLPOS).....	507
10.5.5.7	Synchronisationsmodus (CPFMSON).....	508
10.5.5.8	Verhalten der Folgeachse beim Einschalten (CPFMON).....	510
10.5.5.9	Verhalten der Folgeachse beim Ausschalten (CPFMOF).....	511
10.5.5.10	Position der Folgeachse beim Ausschalten (CPFPOS+CPOF).....	512
10.5.5.11	Zustand bei RESET (CPMRESET).....	512

10.5.5.12	Zustand bei Teileprogrammstart (CPMSTART).....	514
10.5.5.13	Zustand bei Teileprogrammstart unter Suchlauf via Programmtest (CPMPRT).....	515
10.5.5.14	Verschiebung / Skalierung (CPLINTR, CPLINSC, CPLOUTTR, CPLOUTSC).....	516
10.5.5.15	Synchronlaufüberwachung Stufe 1 (CPSYNCOV, CPSYNFIP, CPSYNCOV, CPSYNFIV).....	519
10.5.5.16	Synchronlaufüberwachung Stufe 2 (CPSYNCOV2, CPSYNFIP2).....	523
10.5.5.17	Reaktion auf Stopp-Signale und -kommandos (CPMBRAKE).....	526
10.5.5.18	Reaktion auf bestimmte NC/PLC-Nahtstellensignale (CPMVDI)	528
10.5.5.19	Alarmunterdrückung (CPMALARM).....	532
10.5.6	Kopplungskaskadierung.....	533
10.5.7	Kompatibilität.....	534
10.5.7.1	Anpasszyklen.....	534
10.5.7.2	Kopplungstypen (CPSETTYPE).....	536
10.5.7.3	Projektierte Kopplung (CPRES).....	540
10.5.8	Kanalübergreifende Kopplung, Achstausch.....	541
10.5.9	Verhalten bei Rundachsen.....	542
10.5.10	Verhalten bei POWER ON, RESET,	544
10.5.11	CP-SW-Limit-Überwachung.....	544
10.5.11.1	Funktion.....	544
10.5.11.2	Parametrierung.....	547
10.5.11.3	Programmierung.....	547
10.5.11.4	Randbedingungen.....	547
10.5.11.5	Beispiele.....	549
10.5.12	Störverhalten.....	550
10.5.12.1	Schnellstopp.....	550
10.5.13	Nachführen der Synchronlaufabweichung.....	551
10.5.13.1	Grundlagen.....	551
10.5.13.2	Synchronlaufabweichung messen.....	552
10.5.13.3	Synchronlaufabweichung direkt angeben.....	555
10.5.13.4	Synchronlaufkorrektur.....	555
10.5.13.5	Diagnose der Synchronlaufkorrektur.....	556
10.5.13.6	Ablöschen einer Synchronlaufkorrektur.....	556
10.5.13.7	Randbedingungen.....	558
10.5.14	Beispiele.....	559
10.5.14.1	Programmierbeispiele.....	559
10.5.14.2	Anpasszyklus anpassen.....	560
10.6	Dynamikverhalten der Folgeachse.....	561
10.6.1	Parametrierte Dynamikgrenzen.....	561
10.6.2	Programmierte Dynamikgrenzen.....	562
10.6.2.1	Programmierung (VELOLIMA, ACCLIMA).....	562
10.6.2.2	Beispiele.....	564
10.6.2.3	Systemvariablen.....	565
10.7	Allgemeine Randbedingungen.....	565
10.8	Datenlisten.....	566
10.8.1	Maschinendaten.....	566
10.8.1.1	NC-spezifische Maschinendaten.....	566
10.8.1.2	Kanal-spezifische Maschinendaten.....	566
10.8.1.3	Achs-/Spindel-spezifische Maschinendaten.....	567
10.8.2	Settingdaten.....	567
10.8.2.1	Achs-/Spindel-spezifische Settingdaten.....	567
10.8.3	Systemvariablen.....	568

10.8.4	Signale.....	571
10.8.4.1	Signale an Achse/Spindel.....	571
10.8.4.2	Signale von Achse/Spindel.....	571
11	R3: Erweitertes Stillsetzen und Rückziehen.....	573
11.1	Kurzbeschreibung.....	573
11.2	Steuerungsgeführtes ESR - nur 840D sl.....	573
11.2.1	Grundlagen.....	573
11.2.2	Antriebsautarke Reaktionen.....	575
11.2.3	Überwachung der Zwischenkreisspannung.....	575
11.2.4	Erweitertes Stillsetzen.....	576
11.2.5	Rückziehen.....	578
11.2.5.1	Funktion.....	578
11.2.5.2	Parametrierung: Maschinendaten.....	579
11.2.5.3	Parametrierung: Systemvariable.....	580
11.2.5.4	Programmierung (POLF, POLFA, POLFMASK, POLFMLIN).....	580
11.2.5.5	Randbedingungen.....	582
11.2.6	Auslösequellen.....	584
11.2.7	Verknüpfungslogik: Quellen-/Reaktionsverknüpfung.....	585
11.2.8	Aktivierung.....	586
11.2.9	Projektierungshilfe für ESR.....	586
11.2.10	Steuerungsverhalten.....	589
11.2.10.1	Achsverhalten in Abhängigkeit der Freigaben.....	589
11.2.10.2	POWER OFF/POWER ON.....	591
11.2.10.3	Betriebsartenwechsel, NC-Stop, Reset.....	591
11.2.10.4	Teileprogrammstart, NC-Start.....	592
11.2.10.5	Alarmverhalten.....	592
11.2.10.6	Satzsuchlauf, REPOS.....	593
11.3	Antriebsautarkes ESR.....	593
11.3.1	Grundlagen.....	593
11.3.2	Stillsetzen im Antrieb projektieren.....	594
11.3.3	Rückziehen im Antrieb projektieren.....	596
11.3.4	Generatorbetrieb im Antrieb projektieren.....	597
11.3.5	ESR über Systemvariable freigeben.....	599
11.3.6	ESR über Systemvariable auslösen.....	599
11.3.7	Rückmeldung des ESR-Status.....	600
11.3.8	ESR-Reaktionen quittieren.....	600
11.3.9	ESR im Teileprogramm projektieren.....	601
11.3.9.1	Stillsetzen (ESRS).....	602
11.3.9.2	Rückziehen (ESRR).....	602
11.3.9.3	Randbedingungen.....	604
11.3.10	ESR und Safety Integrated (840D sl).....	605
11.3.11	ESR und Safety Integrated (828D).....	606
11.4	Randbedingungen.....	607
11.5	Beispiele.....	608
11.5.1	NC-geführte Reaktionen.....	608
11.5.2	Rückziehen während Gewindeschneiden.....	610
11.5.3	Schnellabheben über ASUP und schnellen Eingang.....	610
11.5.4	Schnellabheben absolut und inkrementell.....	611
11.5.5	Schnellabheben im linearen Zusammenhang der Achsen.....	611

11.6	Datenlisten.....	612
11.6.1	Maschinendaten.....	612
11.6.1.1	Kanal-spezifische Maschinendaten.....	612
11.6.1.2	Achs-/Spindel-spezifische Maschinendaten.....	612
11.6.2	Systemvariablen.....	612
11.6.3	Signale.....	613
11.6.3.1	Signale an Kanal.....	613
11.6.3.2	Signale an Achse/Spindel.....	613
11.6.3.3	Signale von Achse/Spindel.....	613
12	S9: Sollwertumschaltung - nur 840D sl.....	615
12.1	Kurzbeschreibung.....	615
12.2	Inbetriebnahme.....	617
12.3	Ablaufdiagramm.....	620
12.4	Randbedingungen.....	621
12.5	Datenlisten.....	622
12.5.1	Maschinendaten.....	622
12.5.1.1	Achs-/Spindel-spezifische Maschinendaten.....	622
13	T3: Tangentialsteuerung - nur 840D sl.....	623
13.1	Inbetriebnahme.....	624
13.2	Programmierung.....	625
13.2.1	Kopplung definieren (TANG).....	625
13.2.2	Zwischensatzerzeugung einschalten (TLIFT).....	627
13.2.3	Kopplung einschalten (TANGON).....	629
13.2.4	Kopplung ausschalten (TANGOF).....	630
13.2.5	Kopplung löschen (TANGDEL).....	630
13.3	Grenzwinkel.....	632
13.4	Randbedingungen.....	633
13.5	Beispiele.....	633
13.6	Datenlisten.....	636
13.6.1	Maschinendaten.....	636
13.6.1.1	NC-spezifische Maschinendaten.....	636
13.6.1.2	Achs-/Spindel-spezifische Maschinendaten.....	636
13.6.2	Settingdaten.....	636
13.6.2.1	Kanal-spezifische Settingdaten.....	636
13.6.3	Systemvariablen.....	636
14	T4: Automatisches Nachoptimieren mit AST (Option) - nur 840D sl.....	637
14.1	Funktion.....	637
14.2	Inbetriebnahme.....	639
14.3	Programmierung.....	640
14.3.1	Übersicht.....	640
14.3.2	CYCLE751 - Optimierungssitzung öffnen/ausführen/schließen.....	641
14.3.3	CYCLE752 - Achse zu einer Optimierungssitzung hinzufügen.....	642
14.3.4	CYCLE753 - Optimierungsmodus auswählen.....	645

14.3.5	CYCLE754 - Datensatz hinzufügen/entfernen.....	646
14.3.6	CYCLE755 - Daten sichern/wiederherstellen.....	647
14.3.7	CYCLE756 - Optimierungsergebnisse aktivieren.....	648
14.3.8	CYCLE757 - Optimierungsdaten speichern.....	650
14.3.9	CYCLE758 - Parameterwert ändern.....	653
14.3.10	CYCLE759 - Parameterwert lesen.....	655
14.3.11	Liste der Parameter zur automatischen Servo Optimierung.....	656
14.4	Diagnose.....	661
14.5	Randbedingungen.....	662
14.6	Beispiele.....	662
14.6.1	Beispiel 1: Vermessen einer Achse mit AST-StandardEinstellungen.....	662
14.6.2	Beispiel 2: Drehzahlregler einer Achse neu optimieren.....	664
14.6.3	Beispiel 3: Drehzahlregler und Lageregler einer Achse neu optimieren.....	667
14.6.4	Beispiel 4: Nachoptimieren der Bahninterpolation.....	672
14.6.5	Beispiel 5: Nachoptimieren des Drehzahlregelkreises zur Eliminierung bekannter periodischer Störfrequenzen.....	677
14.6.6	Beispiel 6: Vermessen einer Achse ohne Optimierung.....	679
15	T5: Intelligente Lastanpassung (Option) - nur 840D sl.....	683
15.1	Funktion.....	683
15.2	Parametrierung.....	684
15.3	Beispiel.....	687
16	TE01: Installation und Aktivierung ladbarer Compile-Zyklen.....	689
16.1	Laden von Compile-Zyklen.....	691
16.1.1	Laden eines Compile-Zyklus mit SINUMERIK Operate.....	691
16.1.2	Laden eines Compile-Zyklus mit HMI Advanced.....	691
16.1.3	Laden eines Compile-Zyklus von einem externen Rechner mit WinSCP3.....	692
16.2	Kompatibilität der Interfaceversionen.....	693
16.3	Software-Version eines Compile-Zyklus.....	694
16.4	Aktivieren der Technologiefunktionen im NC.....	695
16.5	Funktionsspezifische Inbetriebnahme.....	696
16.6	Anlegen von Alarmtexten.....	696
16.6.1	Alarmtexte anlegen mit SINUMERIK Operate.....	696
16.6.2	Alarmtexte anlegen mit HMI Advanced.....	697
16.7	Hochrüsten eines Compile-Zyklus.....	698
16.8	Löschen eines Compile-Zyklus.....	699
16.9	Datenlisten.....	700
16.9.1	Maschinendaten.....	700
16.9.1.1	NC-spezifische Maschinendaten.....	700
17	TE02: Simulation von Compile-Zyklen (nur HMI Advanced).....	701
17.1	Kurzbeschreibung.....	701
17.1.1	Funktion.....	701
17.2	OEM-Transformationen.....	701

18	TE1: Abstandsregelung - nur 840D sl.....	703
18.1	Kurzbeschreibung.....	703
18.1.1	Kurzbeschreibung.....	703
18.1.2	Funktionsbeschreibung.....	704
18.2	Abstandsreglung.....	706
18.2.1	Regeldynamik.....	706
18.2.2	Geschwindigkeitsvorsteuerung.....	708
18.2.3	Regelkreisstruktur.....	709
18.2.4	Kompensationsvektor.....	710
18.3	Technologische Eigenschaften der Abstandsreglung.....	713
18.4	Sensor-Kollisionsüberwachung.....	714
18.5	Inbetriebnahme.....	715
18.5.1	Aktivieren der Technologiefunktion.....	715
18.5.2	Speicherkonfiguration durchführen.....	715
18.5.3	Parametrierung der Eingangssignale.....	715
18.5.4	Parametrierung des programmierbaren Kompensationsvektors.....	716
18.5.5	Parametrierung der Abstandsregelung.....	718
18.5.6	Inbetriebnahme der Abstandsregelung.....	719
18.6	Programmierung.....	721
18.6.1	Ein- und Ausschalten der Abstandsregelung (CLC).....	721
18.6.2	Regelkreisverstärkung (CLC_GAIN).....	726
18.6.3	Begrenzung des Regelbereichs (CLC_LIM).....	728
18.6.4	Richtungsabhängiges Sperren der Verfahrbewegung.....	730
18.6.5	Satzweise vorgebbarer Spannungsoffset (CLC_VOFF).....	732
18.6.6	Per Synchronaktion vorgebbarer Spannungsoffset.....	732
18.6.7	Auswahl der aktiven Sensorkennlinie (CLC_SEL).....	733
18.7	Funktionsspezifische Anzeigedaten.....	734
18.7.1	Kanalspezifische GUD-Variable.....	735
18.7.2	BTSS-Variable.....	737
18.8	Funktionsspezifische Alarmtexte.....	738
18.9	Randbedingungen.....	738
18.9.1	Peripheriebaugruppen.....	738
18.9.1.1	Externe Glättungsfilter.....	739
18.9.2	Funktionsspezifische Randbedingungen.....	740
18.10	Datenlisten.....	742
18.10.1	Maschinendaten.....	742
18.10.1.1	NC-spezifischen Maschinendaten.....	742
18.10.1.2	Kanal-spezifische Maschinendaten.....	742
18.10.1.3	Achs-/Spindel-spezifische Maschinendaten.....	743
18.10.2	Parameter Antrieb (SINAMICS S120).....	744
18.10.3	Signale.....	744
18.10.3.1	Signale an Kanal.....	744
18.10.3.2	Signale von Kanal.....	744
19	TE3: Drehzahl-/Drehmomentkopplung, Master-Slave.....	745
19.1	Kurzbeschreibung.....	745

19.2	Kopplungsschaltbild.....	746
19.3	Konfiguration einer Kopplung.....	747
19.4	Momentenausgleichsregler.....	749
19.5	Verspannmoment.....	752
19.6	Ein-/Ausschalten einer Kopplung.....	755
19.7	Ein-/Ausschaltverhalten.....	757
19.8	Randbedingungen.....	761
19.8.1	Funktionale Randbedingungen.....	761
19.8.2	Axiale NC/PLC-Nahtstellensignale.....	763
19.8.3	Zusammenspiel mit anderen Funktionen.....	764
19.9	Beispiele.....	770
19.9.1	Master-Slave-Kopplung zwischen AX1=Master und AX2=Slave.....	770
19.9.2	Kopplung schließen über PLC.....	770
19.9.3	Kopplung schließen/trennen über Teileprogramm bei SINUMERIK 840D sl.....	771
19.9.4	Mechanische Bremse öffnen.....	772
19.10	Datenlisten.....	773
19.10.1	Maschinendaten.....	773
19.10.1.1	Achs-/Spindel-spezifische Maschinendaten.....	773
19.10.2	Systemvariablen.....	773
19.10.3	Signale.....	774
19.10.3.1	Signale an Achse/Spindel.....	774
19.10.3.2	Signale von Achse/Spindel.....	774
20	TE4: Transformationspaket Handling - nur 840D sl.....	775
20.1	Kurzbeschreibung.....	775
20.2	Kinematische Transformation.....	776
20.3	Begriffsbestimmungen.....	776
20.3.1	Einheiten und Richtungen.....	776
20.3.2	Positions- und Orientierungsbeschreibung mit Hilfe von Frames.....	777
20.3.3	Gelenkdefinition.....	778
20.4	Konfiguration der kinematischen Transformation.....	779
20.4.1	Allgemeine Maschinendaten.....	780
20.4.2	Parametrierung über Geometriedaten.....	781
20.5	Kinematikbeschreibungen.....	794
20.5.1	3-Achs-Kinematiken.....	794
20.5.2	4-Achs-Kinematiken.....	800
20.5.3	5-Achs-Kinematiken.....	807
20.5.4	6-Achs-Kinematiken.....	811
20.5.5	Sonderkinematiken.....	811
20.6	Werkzeugorientierung.....	816
20.6.1	Orientierungsprogrammierung bei 4-Achsern.....	818
20.6.2	Orientierungsprogrammierung bei 5-Achsern.....	819
20.7	Singuläre Stellungen und ihre Behandlung.....	821
20.8	Aufruf und Anwendung der Transformation.....	821

20.9	Istwertanzeige.....	822
20.10	Werkzeugprogrammierung.....	823
20.11	Kartesische PTP-Fahren mit Transformationspaket Handling.....	824
20.12	Inbetriebnahme.....	824
20.12.1	Allgemeine Inbetriebnahme.....	824
20.12.2	Funktionsspezifische Inbetriebnahme.....	824
20.13	Randbedingungen.....	826
20.14	Datenlisten.....	827
20.14.1	Maschinendaten.....	827
20.14.1.1	Allgemeine Maschinendaten.....	827
20.14.1.2	Kanal-spezifische Maschinendaten.....	827
20.14.1.3	Kanal-spezifische Maschinendaten für Compile-Zyklen.....	827
20.14.2	Signale.....	829
20.14.2.1	Signale an Kanal.....	829
20.14.2.2	Signale von Kanal.....	829
21	TE6: MKS-Kopplung - nur 840D sl.....	831
21.1	Kurzbeschreibung.....	831
21.2	Funktionsbeschreibung MKS-Kopplung.....	832
21.2.1	Kopplungspaare definieren.....	832
21.2.2	Kopplung EIN-/AUS schalten.....	833
21.2.3	Toleranzfenster.....	834
21.2.4	Speicherkonfiguration: Satzspeicher.....	835
21.3	Funktionsbeschreibung Kollisionsschutz.....	835
21.3.1	Schutzpaare definieren.....	835
21.3.2	Kollisionsschutz EIN-/AUSschalten.....	835
21.3.3	Projektierungsbeispiel.....	836
21.4	Anwenderspezifische Projektierungen.....	837
21.5	Besondere Betriebszustände.....	838
21.6	Randbedingungen.....	839
21.7	Datenlisten.....	840
21.7.1	Maschinendaten.....	840
21.7.1.1	Kanal-spezifische Maschinendaten.....	840
21.7.1.2	Achs-/Spindel-spezifische Maschinendaten.....	840
21.7.2	Signale.....	840
21.7.2.1	Signale an Achse/Spindel.....	840
21.7.2.2	Signale von Achse/Spindel.....	840
22	TE7: Wiederaufsetzen (Retrace Support) - nur 840D sl.....	843
22.1	Kurzbeschreibung.....	843
22.2	Funktionsbeschreibung.....	844
22.2.1	Funktion.....	844
22.2.2	Begriffsdefinitionen.....	846
22.2.3	Funktionsablauf (Prinzip).....	847
22.2.4	Maximaler RESU-fähiger Konturbereich.....	850

22.3	Inbetriebnahme.....	851
22.3.1	Aktivierung.....	851
22.3.2	Festlegung der RESU-Arbeitsebene.....	851
22.3.3	Speicherkonfiguration: Satzspeicher.....	852
22.3.4	Speicherkonfiguration: Heap-Speicher.....	853
22.3.5	Speicherbereich des RESU-Hauptprogramms.....	854
22.3.6	Ablage der RESU-Unterprogramme.....	855
22.3.7	ASUP-Freigabe.....	856
22.3.8	PLC-Anwenderprogramm.....	856
22.4	Programmierung.....	857
22.4.1	RESU-Start/Stop/Reset (CC_PREPRE).....	857
22.5	RESU-spezifische Teileprogramme.....	858
22.5.1	Übersicht.....	858
22.5.2	Hauptprogramm (CC_RESU.MPF).....	859
22.5.3	INI-Programm (CC_RESU_INI.SPF).....	860
22.5.4	END-Programm (CC_RESU_END.SPF).....	862
22.5.5	Wiederaufsetz-ASUP (CC_RESU_BS_ASUP.SPF).....	863
22.5.6	RESU-ASUP (CC_RESU_ASUP.SPF).....	863
22.6	Wiederaufsetzen.....	864
22.6.1	Allgemeine Informationen.....	864
22.6.2	Satzsuchlauf mit Berechnung an der Kontur.....	865
22.6.3	Repositionieren.....	865
22.6.4	Zeitliche Bedingungen bezüglich NC-Start.....	866
22.6.5	Satzsuchlauf ab letztem Hauptsatz.....	867
22.7	Funktionsspezifische Anzeigedaten.....	868
22.7.1	Kanalspezifische GUD-Variable.....	868
22.8	Funktionsspezifische Alarmtexte.....	869
22.9	Randbedingungen.....	869
22.9.1	Funktionsspezifische Randbedingungen.....	869
22.9.1.1	Wiederaufsetzen innerhalb von Unterprogrammen.....	869
22.9.1.2	Wiederaufsetzen innerhalb von Programmschleifen.....	870
22.9.1.3	Wiederaufsetzen an Vollkreisen.....	870
22.9.1.4	Automatisch generierte Konturelemente.....	870
22.9.2	Randbedingungen bezüglich Standardfunktionen.....	871
22.9.2.1	Achstausch.....	871
22.9.2.2	Verfahrbewegungen von Kanalachsen.....	871
22.9.2.3	Satznummern.....	871
22.9.2.4	Satzsuchlauf.....	872
22.9.2.5	Transformationen.....	872
22.9.2.6	Kompensationen.....	872
22.9.2.7	Frames.....	873
22.9.2.8	Werkzeugkorrekturen.....	873
22.10	Datenlisten.....	874
22.10.1	Maschinendaten.....	874
22.10.1.1	Allgemeine Maschinendaten.....	874
22.10.1.2	Kanal-spezifische Maschinendaten.....	874
22.10.2	Signale.....	874
22.10.2.1	Signale an Kanal.....	874


22.10.2.2	Signale von Kanal.....	875
23	TE8: Taktunabhängige bahnsynchrone Schaltsignalausgabe - nur 840D sl.....	877
23.1	Kurzbeschreibung.....	877
23.2	Funktionsbeschreibung.....	878
23.2.1	Allgemeine Informationen.....	878
23.2.2	Ermittlung der Schaltpositionen.....	878
23.2.2.1	Satzbezogene Schaltsignalausgabe.....	878
23.2.2.2	Bahnlängenbezogene Schaltsignalausgabe.....	880
23.2.3	Berechnung der Schaltzeitpunkte.....	881
23.2.4	Schaltfrequenz und Schaltpositionsabstand.....	881
23.2.5	Angenäherte Schaltposition.....	882
23.2.6	Programmierte Schaltpositionsverschiebung.....	883
23.2.7	Verhalten bei Teileprogramm-Unterbrechung.....	883
23.3	Inbetriebnahme.....	884
23.3.1	Aktivierung.....	884
23.3.2	Speicherkonfiguration.....	884
23.3.3	Parametrierung der digitalen Onboard-Ausgänge.....	884
23.3.4	Parametrierung des Schaltsignals.....	885
23.3.5	Parametrierung der Geometrieachsen.....	885
23.4	Programmierung.....	886
23.4.1	Einschalten der satzbezogenen Schaltsignalausgabe (CC_FASTON).....	886
23.4.2	Einschalten der bahnlängenbezogenen Schaltsignalausgabe (CC_FASTON_CONT).....	887
23.4.3	Ausschalten (CC_FASTOFF).....	888
23.5	Funktionsspezifische Alarmtexte.....	889
23.6	Randbedingungen.....	889
23.6.1	Satzsuchlauf.....	889
23.6.2	Transformationen.....	890
23.6.3	Kompensationen.....	891
23.6.4	Werkzeugradiuskorrektur (WRK).....	891
23.6.5	Bahnsteuerbetrieb.....	891
23.6.6	Softwaresnocken.....	892
23.7	Datenlisten.....	892
23.7.1	Maschinendaten.....	892
23.7.1.1	Allgemeine Maschinendaten.....	892
23.7.1.2	Kanal-spezifische Maschinendaten.....	892
24	TE9: Achspaar-Kollisionsschutz.....	893
24.1	Kurzbeschreibung.....	893
24.2	Funktionsbeschreibung.....	893
24.3	Inbetriebnahme.....	894
24.3.1	Freigabe der Technologiefunktion (Option).....	894
24.3.2	Aktivierung der Technologiefunktion.....	895
24.3.3	Aktivierung der Zusatzfunktionen.....	895
24.3.4	Definition eines Achspaares.....	896
24.3.5	Freifahrrichtung.....	896
24.3.6	Offset der Maschinenkoordinatensysteme.....	896
24.3.7	Schutzfenster.....	897


24.3.8	Orientierung.....	898
24.3.9	Schutzfenster-Erweiterung.....	898
24.3.10	Aktivierung der Schutzfunktion.....	899
24.3.11	Achsspezifische Beschleunigung.....	900
24.3.12	Überwachungsstatus (GUD).....	900
24.3.13	PLC-Schnittstelle: Achsspezifische Bremsvorgänge.....	901
24.3.14	PLC-Schnittstelle: Achspaarspezifische Aktivierung der Schutzfunktion.....	902
24.4	Randbedingungen.....	902
24.4.1	Achsen.....	902
24.4.2	Achscontainer.....	903
24.4.3	Link-Achsen.....	904
24.4.4	Interpolatorische Kopplungen.....	904
24.5	Beispiele.....	905
24.5.1	Kollisionsschutz.....	905
24.5.2	Kollisionsschutz und Abstandsbegrenzung.....	906
24.6	Datenlisten.....	908
24.6.1	Optionsdaten.....	908
24.6.2	Maschinendaten.....	908
24.6.2.1	NC-spezifische Maschinendaten.....	908
24.6.2.2	Achs/Spindel-spezifische Maschinendaten.....	908
24.6.3	Signale.....	909
24.6.3.1	Signale an Achse/Spindel.....	909
24.6.4	Anwenderdaten.....	909
24.6.4.1	Globale Anwenderdaten (GUD).....	909
25	V2: Vorverarbeitung.....	911
25.1	Kurzbeschreibung.....	911
25.2	Programmhandling.....	913
25.3	Programmaufruf.....	916
25.4	Randbedingungen.....	918
25.5	Beispiele.....	919
25.5.1	Vorverarbeitung einzelner Dateien.....	919
25.5.2	Vorverarbeitung im dynamischen NC-Speicher.....	921
25.6	Datenlisten.....	921
25.6.1	Maschinendaten.....	921
25.6.1.1	Allgemeine Maschinendaten.....	921
25.6.1.2	Kanal-spezifische Maschinendaten.....	922
26	W5: 3D-Werkzeugradiuskorrektur - nur 840D sl.....	923
26.1	Funktion.....	923
26.1.1	Einleitung.....	923
26.1.2	Umfangsfräsen.....	924
26.1.2.1	Ecken beim Umfangsfräsen.....	926
26.1.2.2	Verhalten an Außenecken.....	927
26.1.2.3	Verhalten an Innenecken.....	928
26.1.2.4	Überwachung der Bahnkrümmung.....	930
26.1.3	Stirnfräsen.....	930
26.1.3.1	Werkzeugformen und Werkzeugdaten für Stirnfräsen.....	931

26.1.3.2	Stirnfräsen mit Angabe des Flächennormalenvektors.....	932
26.1.3.3	Korrektur auf der Bahn.....	934
26.1.3.4	Ecken beim Stirnfräsen.....	935
26.1.3.5	Verhalten an Außenecken.....	936
26.1.3.6	Verhalten an Innenecken.....	937
26.1.3.7	Überwachung der Bahnkrümmung.....	939
26.2	Inbetriebnahme.....	939
26.2.1	Parametrierung.....	939
26.3	Programmierung.....	940
26.3.1	3D-Werkzeugradiuskorrektur für das 3D-Umfangsfräsen anwählen (CUT3DC, CUT3DCD, ISD).....	940
26.3.2	3D-Werkzeugradiuskorrektur für das 3D-Stirnfräsen anwählen (CUT3DF, CUT3DFS, CUT3DFF, CUT3DFD).....	945
26.3.3	3D-Umfangsfräsen unter Berücksichtigung einer Begrenzungsfläche (CUT3DCC, CUT3DCCD).....	951
26.4	Randbedingungen.....	955
26.5	Beispiele.....	956
26.5.1	Beispiel 1: Orientierungsänderung an Außenecke beim 3D-Umfangsfräsen.....	956
26.5.2	Beispiel 2: Orientierungsänderung an Innenecke beim 3D-Umfangsfräsen.....	958
26.6	Datenlisten.....	959
26.6.1	Allgemeine Maschinendaten.....	959
26.6.2	Kanal-spezifische Maschinendaten.....	960
27	W6: Weglängenauswertung - nur 840D sl.....	961
27.1	Kurzbeschreibung.....	961
27.2	Daten.....	962
27.3	Parametrierung.....	962
27.3.1	Allgemeine Aktivierung.....	962
27.3.2	Datengruppen.....	962
27.4	Beispiele.....	963
27.4.1	Verfahrweg pro Teileprogramm.....	963
27.5	Datenlisten.....	964
27.5.1	Maschinendaten.....	964
27.5.1.1	NC-spezifische Maschinendaten.....	964
27.5.1.2	Achs-/Spindel-spezifische Maschinendaten.....	964
28	Z3: NC/PLC-Nahtstellensignale.....	965
A	Anhang.....	967
A.1	Liste der Abkürzungen.....	967
A.2	Dokumentationsübersicht.....	976
	Glossar.....	977
	Index.....	999

Grundlegende Sicherheitshinweise

1.1 Allgemeine Sicherheitshinweise

 WARNUNG
Lebensgefahr bei Nichtbeachtung von Sicherheitshinweisen und Restrisiken
Bei Nichtbeachtung der Sicherheitshinweise und Restrisiken in der zugehörigen Hardware-Dokumentation können Unfälle mit schweren Verletzungen oder Tod auftreten.
<ul style="list-style-type: none">• Halten Sie die Sicherheitshinweise der Hardware-Dokumentation ein.• Berücksichtigen Sie bei der Risikobeurteilung die Restrisiken.

 WARNUNG
Fehlfunktionen der Maschine infolge fehlerhafter oder veränderter Parametrierung
Durch fehlerhafte oder veränderte Parametrierung können Fehlfunktionen an Maschinen auftreten, die zu Körperverletzungen oder Tod führen können.
<ul style="list-style-type: none">• Schützen Sie die Parametrierungen vor unbefugtem Zugriff.• Beherrschen Sie mögliche Fehlfunktionen durch geeignete Maßnahmen, z. B. NOT-HALT oder NOT-AUS.

1.2 Gewährleistung und Haftung für Applikationsbeispiele

Applikationsbeispiele sind unverbindlich und erheben keinen Anspruch auf Vollständigkeit hinsichtlich Konfiguration und Ausstattung sowie jeglicher Eventualitäten.

Applikationsbeispiele stellen keine kundenspezifischen Lösungen dar, sondern sollen lediglich Hilfestellung bieten bei typischen Aufgabenstellungen.

Als Anwender sind Sie für den sachgemäßen Betrieb der beschriebenen Produkte selbst verantwortlich. Applikationsbeispiele entheben Sie nicht der Verpflichtung zu sicherem Umgang bei Anwendung, Installation, Betrieb und Wartung.

1.3 Industrial Security

Hinweis

Industrial Security

Siemens bietet Produkte und Lösungen mit Industrial Security-Funktionen an, die den sicheren Betrieb von Anlagen, Systemen, Maschinen und Netzwerken unterstützen.

Um Anlagen, Systeme, Maschinen und Netzwerke gegen Cyber-Bedrohungen zu sichern, ist es erforderlich, ein ganzheitliches Industrial Security-Konzept zu implementieren (und kontinuierlich aufrechtzuerhalten), das dem aktuellen Stand der Technik entspricht. Die Produkte und Lösungen von Siemens formen nur einen Bestandteil eines solchen Konzepts.

Der Kunde ist dafür verantwortlich, unbefugten Zugriff auf seine Anlagen, Systeme, Maschinen und Netzwerke zu verhindern. Systeme, Maschinen und Komponenten sollten nur mit dem Unternehmensnetzwerk oder dem Internet verbunden werden, wenn und soweit dies notwendig ist und entsprechende Schutzmaßnahmen (z. B. Nutzung von Firewalls und Netzwerksegmentierung) ergriffen wurden.

Zusätzlich sollten die Empfehlungen von Siemens zu entsprechenden Schutzmaßnahmen beachtet werden. Weiterführende Informationen über Industrial Security finden Sie unter:

Industrial Security (<http://www.siemens.com/industrialsecurity>)

Die Produkte und Lösungen von Siemens werden ständig weiterentwickelt, um sie noch sicherer zu machen. Siemens empfiehlt ausdrücklich, Aktualisierungen durchzuführen, sobald die entsprechenden Updates zur Verfügung stehen und immer nur die aktuellen Produktversionen zu verwenden. Die Verwendung veralteter oder nicht mehr unterstützter Versionen kann das Risiko von Cyber-Bedrohungen erhöhen.

Um stets über Produkt-Updates informiert zu sein, abonnieren Sie den Siemens Industrial Security RSS Feed unter:

Industrial Security (<http://www.siemens.com/industrialsecurity>)

Weitere Informationen finden Sie im Internet:

Projektierungshandbuch Industrial Security (<https://support.industry.siemens.com/cs/ww/de/view/108862708>)

**WARNUNG****Unsichere Betriebszustände durch Manipulation der Software**

Manipulationen der Software, z. B. Viren, Trojaner, Malware oder Würmer, können unsichere Betriebszustände in Ihrer Anlage verursachen, die zu Tod, schwerer Körperverletzung und zu Sachschäden führen können.

- Halten Sie die Software aktuell.
- Integrieren Sie die Automatisierungs- und Antriebskomponenten in ein ganzheitliches Industrial Security-Konzept der Anlage oder Maschine nach dem aktuellen Stand der Technik.
- Berücksichtigen Sie bei Ihrem ganzheitlichen Industrial Security-Konzept alle eingesetzten Produkte.
- Schützen Sie die Dateien in Wechselspeichermedien vor Schadsoftware durch entsprechende Schutzmaßnahmen, z. B. Virens Scanner.
- Schützen Sie den Antrieb vor unberechtigten Änderungen, indem Sie die Umrichterfunktion „Know-How-Schutz“ aktivieren.

F2: Mehrachstransformationen

2.1 Kurzbeschreibung

2.1.1 Allgemeine Vorgaben

Transformationsdatensatz

Eine Transformation wird mit einer bestimmten Anzahl von Maschinendaten beschrieben. Alle Maschinendaten eines Transformationsdatensatzes sind durch den gleichen Präfix <x> gekennzeichnet. Mit <x> = 1, 2, 3, ... maximale Anzahl möglicher Transformationen im Kanal.

Beispiel anhand des Transformationstyps einer Transformation: MD24100
\$MC_TRAFO_TYPE_<x>

- Transformationstyp der ersten Transformation: MD24100 \$MC_TRAFO_TYPE_1
- Transformationstyp der zweiten Transformation: MD24100 \$MC_TRAFO_TYPE_2
- ...

Achsamen

Während eine Transformation aktiv ist, müssen die Kanal-, Geometrie- und Maschinenachsenamen innerhalb eines Kanal unterschiedlich sein.

- MD10000 \$MN_AXCONF_MACHAX_NAME_TAB (Maschinenachsenname)
- MD20080 \$MC_AXCONF_CHANAX_NAME_TAB (Kanalachsenname)
- MD20060 \$MC_AXCONF_GEOAX_NAME_TAB (Geometrieachsenname)

2.1.2 5-Achs-Transformation

Funktion

Das Bearbeitungspaket "5-Achs-Transformation" ist für die Bearbeitung räumlich gekrümmter Flächen vorgesehen, die außer den drei Linearachsen X, Y und Z noch über zwei Rundachsen verfügen. Damit kann ein rotationssymmetrisches Werkzeug (Fräser, Laserstrahl) im Bearbeitungsraum beliebig zum Werkstück orientiert werden.

Die Programmierung der Bahn und der Bahngeschwindigkeit erfolgt wie bei der 3-Achs-Programmierung. Die Werkzeugausrichtung wird in den Bewegungssätzen zusätzlich programmiert.

Die Echtzeit-Transformation übernimmt die Berechnung der resultierenden Bewegungen aller 5 Achsen. Damit sind die erstellten Bearbeitungsprogramme maschinenunabhängig. Der Einsatz kinematikspezifischer Postprozessoren für die 5-Achs-Bearbeitung entfällt.

Zur Anpassung der Steuerung an verschiedene Maschinenkinematiken besteht die Auswahl aus verschiedenen Transformationen. Zwischen zwei bei der Inbetriebnahme parametrisierten Transformationen kann zur Laufzeit über Teileprogrammbeefehle umgeschaltet werden.

Damit werden die drei möglichen, grundlegenden Maschinenkonfigurationen abgedeckt, die sich in der Orientierung von Werkzeug und Werkstück unterscheiden:

- Orientierung von Werkzeug mit Zweiachsen-Schwenkkopf (Maschinentyp 1)
- Orientierung von Werkstück mit Zweiachsen-Drehtisch (Maschinentyp 2)
- Orientierung von Werkstück und Werkzeug mit Einachs-Drehtisch und -Schwenkkopf (Maschinentyp 3)

Bei der Bearbeitung wird eine Werkzeuglängenkorrektur eingerechnet.

Durch die Ablage der Orientierung zur Werkstückoberfläche in einem eigenen FRAME ist auch ein senkrecht zum Werkstück orientierter Werkzeugrückzug möglich.

Werkzeug-Orientierung

Die Werkzeugorientierung kann auf zwei Arten angegeben werden:

- **Maschinenbezogene Orientierung**
Die maschinenbezogene Orientierung ist von der Maschinenkinematik abhängig.
- **Werkstückbezogene Orientierung**
Die werkstückbezogene Orientierung ist unabhängig von der Maschinenkinematik.
Sie wird programmiert mit:
 - Eulerwinkeln
 - RPY-Winkeln
 - Vektorkomponenten

Mit der werkstückbezogenen Orientierung wird die Richtung des Werkzeugs im Werkstückkoordinatensystem beschrieben. Es kann eine bestimmte Komponente des Werkzeugs in ihrer Ausrichtung zum Werkstück programmiert werden. In den meisten Fällen wird dies eine Längsachse des Werkzeugs mit der Werkzeugspitze (TCP-Tool Center Point) sein, was auch unter dem Begriff der TCP-Programmierung bekannt ist.

Systemvariablen für die Orientierung

Teileprogramme und Synchronaktionen können lesend auf die Systemvariablen zugreifen, die Auskunft geben über:

- Endorientierung des Satzes (Vorlaufwert)
- Sollorientierung
- Istorientierung
- Winkel zwischen Soll- und Istorientierung
- Status für die Variable der Istorientierung

Sonderfälle der 5-Achs-Transformation

Die folgenden Transformationen sind als Sonderfälle der allgemeinen 5-Achs-Transformation aufzufassen:

- **3- und 4-Achs-Transformation**
Es existieren 2 oder 3 Linearachsen und eine Rundachse.
- **Schwenkbare Linearachse**
Eine der Rundachsen dreht die 3. Linearachse.
- **Kardanischer Fräskopf**
Die beiden Rundachsen stehen in einem projektierbaren Winkel zueinander.

Für all diese Transformationen werden die Kenntnisse der allgemeinen 5-Achs-Transformation vorausgesetzt.

2.1.3 3- und 4-Achs-Transformation

Funktion

Die 3- und 4-Achstransformationen zeichnen sich durch folgende Merkmale aus:

Transformation	Merkmale
3-Achs-Transformation	2 translatorische Achsen 1 rotatorische Achse
4-Achs-Transformation	3 translatorische Achsen 1 rotatorische Achse

Beide Transformationsarten gehören zu den Orientierungstransformationen. Die Orientierung des Werkzeuges muss explizit programmiert werden. Die Orientierung des Werkzeuges erfolgt in einer Ebene senkrecht zur rotatorischen Achse.

2.1 Kurzbeschreibung

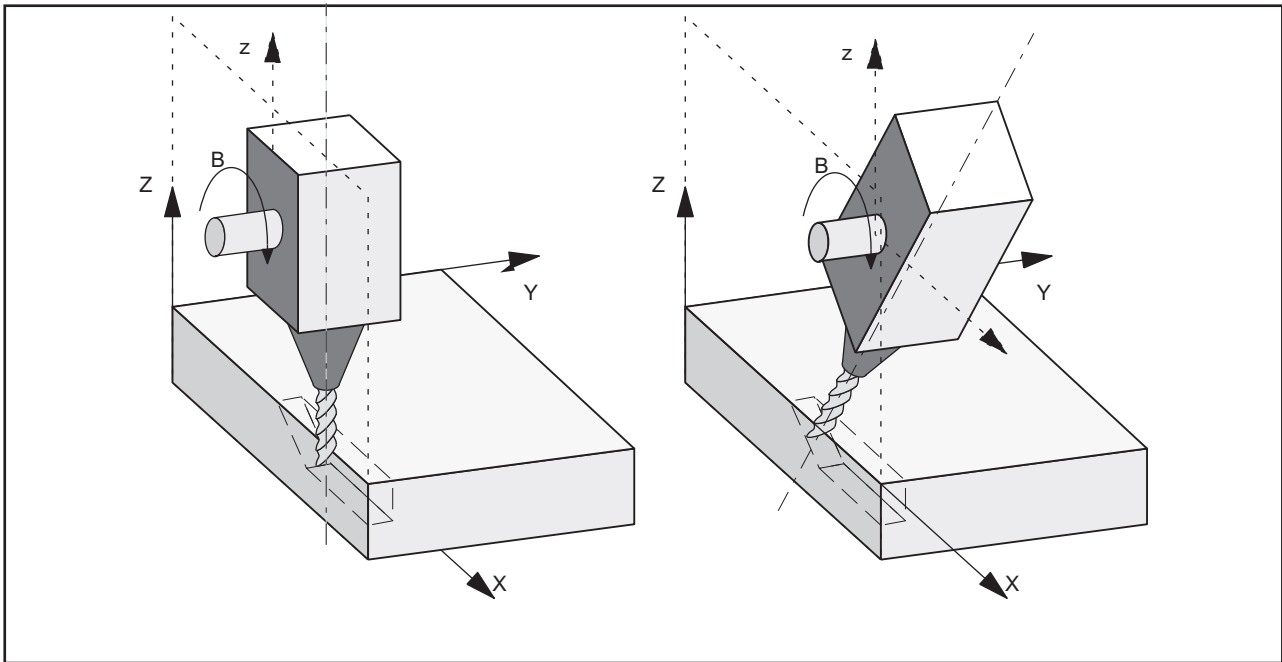


Bild 2-1 Schematische Darstellung einer 3-Achs-Transformation

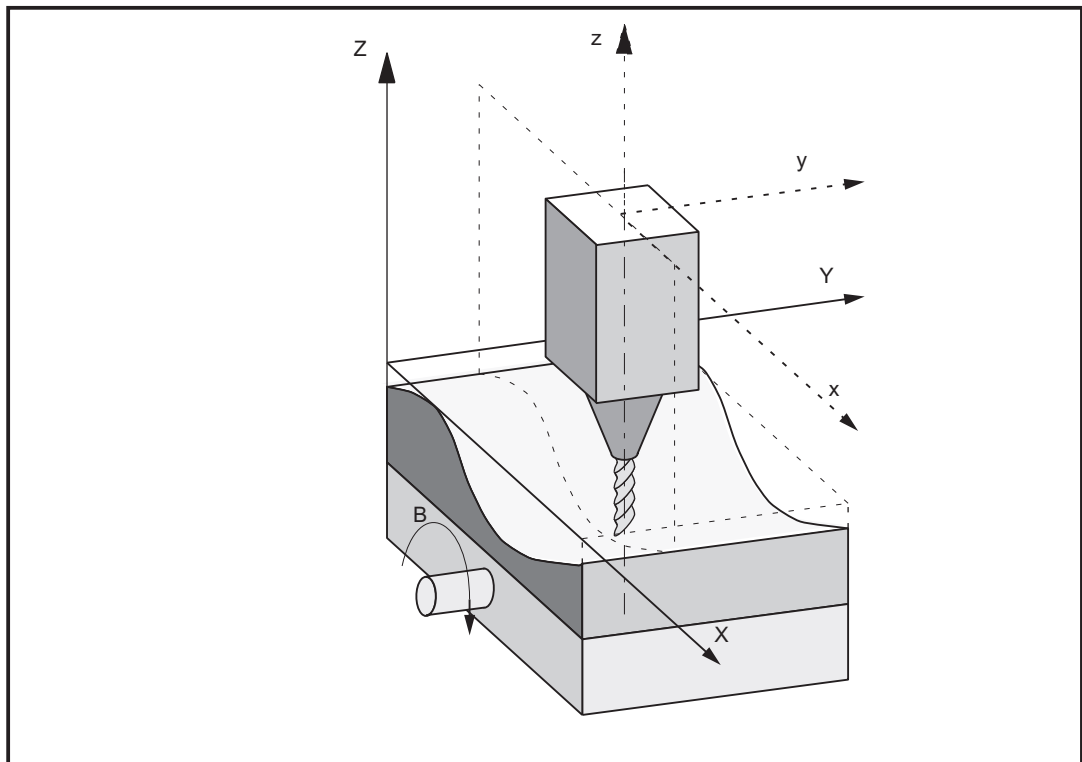


Bild 2-2 Schematische Darstellung einer 4-Achs-Transformation mit beweglichem Werkstück

2.1.4 Orientierungstransformation mit schwenkbarer Linearachse

Funktion

Die Orientierungstransformation mit schwenkbarer Linearachse ähnelt der 5-Achstransformation für Maschinentyp 3, jedoch steht die 3. Linearachse nicht immer senkrecht auf der durch die anderen beiden Linearachsen aufgespannten Ebene.

Merkmale der Kinematik

- Kinematik mit drei Linearachsen und zwei orthogonalen Rundachsen
- Die Rundachsen sind parallel zu zwei der drei Linearachsen
- Die erste Rundachse wird von zwei kartesischen Linearachsen bewegt. Sie dreht die dritte Linearachse, welche das Werkzeug bewegt. Das Werkzeug ist parallel zur dritten Linearachse ausgerichtet.
- Die zweite Rundachse dreht das Werkstück.
- Es ist eine Kinematik mit bewegtem Werkstück und bewegtem Werkzeug.

Das folgende Bild verdeutlicht die Zusammenhänge für eine der zulässigen Achsfolgen, für welche die Transformation möglich ist.

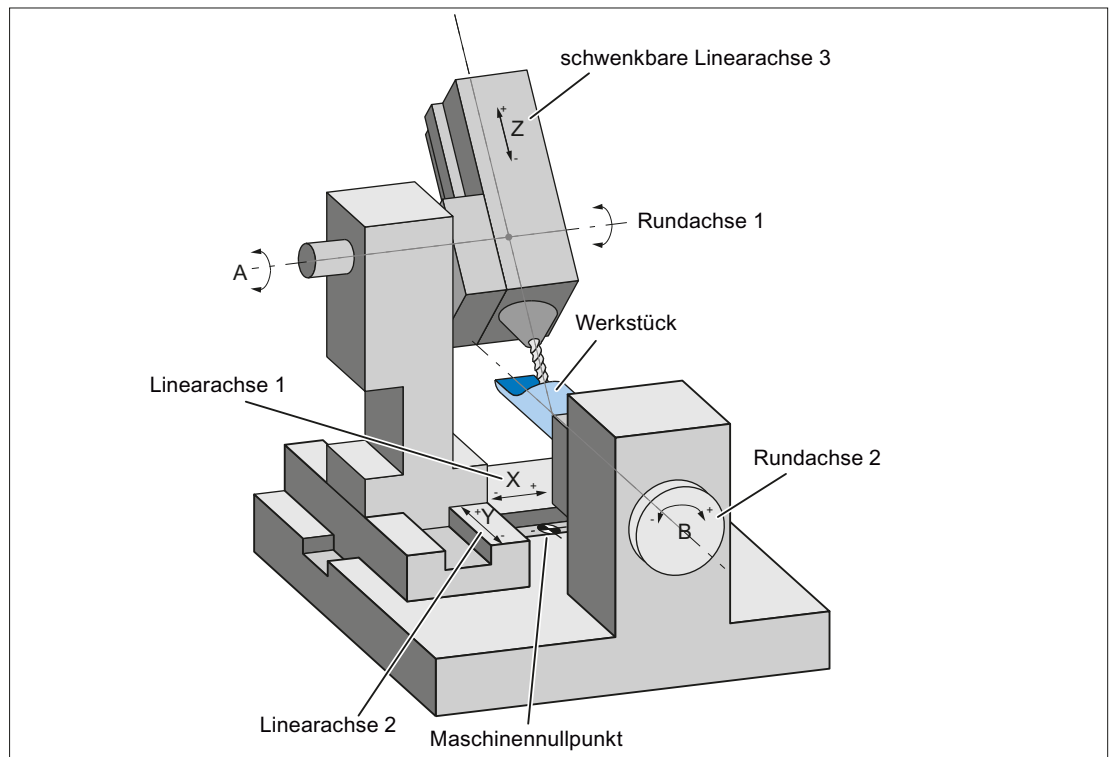


Bild 2-3 Schematisierte Darstellung einer Maschine mit schwenkbarer Linearachse

2.1.5 Kardanischer Fräskopf

Funktion

Eine Werkzeugmaschine mit kardanischem Fräskopf hat mindestens 5 Achsen:

- 3 translatorische Achsen
 - für geradlinige Bewegungen [X, Y, Z]
 - bewegen den Arbeitspunkt an jede beliebige Position im Arbeitsraum
- 2 rotatorische Schwenkachsen
 - sind unter einem projektierbaren Winkel (meist 45 Grad) angeordnet
 - ermöglichen dem Werkzeug, Orientierungen im Raum einzunehmen (beschränken sich bei einer 45-Grad-Anordnung auf eine Halbkugel)

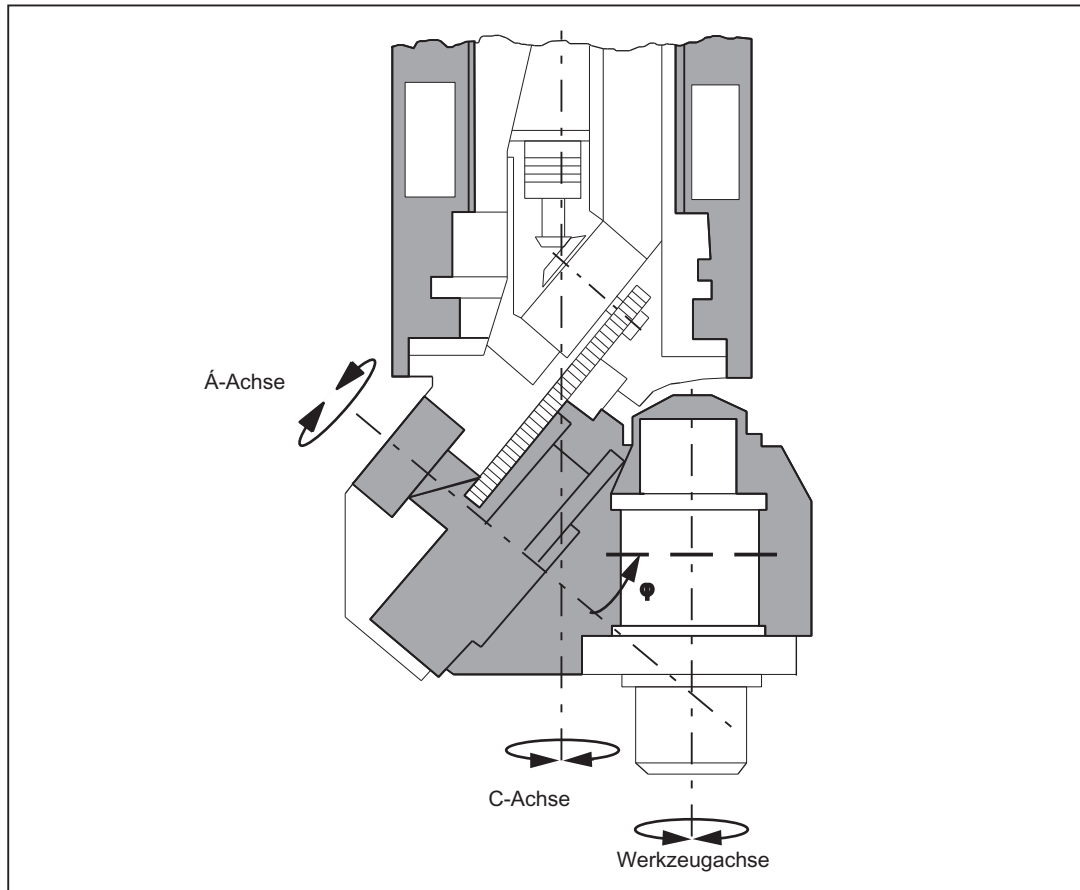


Bild 2-4 Schematische Darstellung einer Werkzeugmaschine mit kardanischem Fräskopf

2.1.6 Orientierungsachsen

Modell zur Beschreibung der Orientierungsänderung

Bei Roboter-, Hexapoden- oder Nutatorkinematiken existiert kein derart einfacher Zusammenhang zwischen Achsbewegung und Orientierungsänderung wie es bei konventionellen 5-Achs-Maschinen der Fall ist.

Deshalb wird ein Modell zur Beschreibung der Orientierungsänderung geschaffen, das unabhängig von der tatsächlichen Maschine ist. Dieses Modell definiert drei virtuelle Orientierungsachsen, die man sich als Drehungen um die Koordinatenachsen eines rechtwinkligen Koordinatensystems vorstellen kann.

Für die 6-Achs-Transformation wird ein weiterer dritter Freiheitsgrad für die Orientierung eingeführt, der die Drehung des Werkzeugs um sich selbst beschreibt.

Echtzeittransformation

Über die Echtzeittransformation werden die kartesischen Koordinaten vom Basiskoordinatensystem ins Maschinenkoordinatensystem umgerechnet.

Die kartesischen Koordinaten bestehen aus:

- **Geometrieachsen**
Geometrieachsen beschreiben den Arbeitspunkt.
- **Orientierungsachsen**
Orientierungsachsen beschreiben die Orientierung des Werkzeugs im Raum.

Orientierung des Werkzeugs

Die Orientierung des Werkzeugs im Raum kann durch Lineare Interpolation, Großkreisinterpolation und mittels Orientierungsvektoren wie folgt definiert werden:

- Direkte Programmierung der Rundachspositionen A, B, C
Bei 5-Achs-Transformation durch Programmierung:
 - der Euler- oder RPY-Winkel in Grad über A_2, B_2, C_2
oder
 - des Richtungsvektors über A_3, B_3, C_3
- Programmierung über Voreilwinkel `LEAD` und Seitwärtswinkel `TILT`

2.1.7 Kartesisches manuelles Verfahren

Funktion

Mit der Funktion "Kartesisches manuelles Verfahren" ist es möglich, als Bezugssystem für die JOG-Bewegung eines der folgenden Koordinatensysteme einzustellen (sowohl für die Translation als auch für die Orientierung getrennt):

- Basiskoordinatensystem (BKS)
- Werkstückkoordinatensystem (WKS)
- Werkzeugkoordinatensystem (TCS)

2.1.8 Kartesisches PTP-Fahren

Funktion

Mit der Funktion "Kartesisches PTP-Fahren" [PTP = Punkt zu Punkt Bewegung (**P**oint to **P**oint)] ist es möglich, eine Position in einem kartesischen Koordinatensystem (Werkstückkoordinatensystem) zu programmieren. Die Bewegung der Maschine erfolgt jedoch in Maschinenkoordinaten.

Die Funktion kann beispielsweise beim Durchfahren einer Singularität angewendet werden. Kartesische Positionen, welche ein CAD-System liefert, müssen nicht in Maschinenachswerte umgerechnet werden.

Außerdem ist das Verfahren der Achsen im kartesischen Koordinatensystem mit aktiver Transformation und programmierten Vorschub zeitintensiver als das direkte Verfahren der Maschinenachsen.

2.1.9 Generische 5-Achstransformation

Funktion

Die "Generische 5-Achstransformation" unterscheidet sich von bisher verfügbaren 5-Achstransformationen dadurch, dass sie keinen Einschränkungen mehr bezüglich der Richtungen der Rundachsen unterliegt.

Die Grundorientierung des Werkzeugs ist im Gegensatz zu den bisher realisierten Orientierungstransformationen nicht mehr durch Maschinendaten fest vorgegeben, sondern frei programmierbar.

2.1.10 Online-Werkzeuglängenkorrektur

Funktion

Über die Systemvariable \$AA_TOFF[] können die effektiven Werkzeuglängen dreidimensional in Echtzeit überlagert werden. Diese Korrekturen sind bei einer aktiven Orientierungstransformation (TRAORI) oder einem aktiven Werkzeugträger in den jeweiligen Werkzeugrichtungen wirksam.

Bei einer Orientierungsänderung des Werkzeugs werden die aufgebauten Werkzeuglängenkorrekturen mitgedreht, so dass sich der Drehpunkt für die Orientierungsbewegung immer auf die korrigierte Werkzeugspitze bezieht.

2.1.11 Aktivierung über Teileprogramm/Softkey

Die für die kinematischen Transformationen relevanten Maschinendaten werden bisher größtenteils über POWER ON wirksam gesetzt.

Transformations-Maschinendaten können auch über Teileprogramm/Softkey wirksam gesetzt werden, wodurch das Booten der Steuerung entfallen kann.

Literatur:

Funktionshandbuch Erweiterungsfunktionen; Kinematische Transformationen (M1), Kapitel: Kartesisches PTP-Fahren

2.1.12 Komprimierung der Orientierung

Bei der Abarbeitung von NC-Programmen, die Sätze mit relativ kurzen Verfahrenswegen enthalten, kann es abhängig vom eingestellten Interpolatortakt zu einer Reduzierung der Bahngeschwindigkeit mit entsprechender Verlängerung der Bearbeitungszeit kommen.

COMPON, COMPCURV, COMPCAD

Durch die Aktivierung einer der "Kompressoren" COMPON, COMPCURV oder COMPCAD können auch NC-Programme mit kurzen Verfahrenswegen ohne Reduzierung der Bahngeschwindigkeit abgearbeitet werden. Weiterhin werden durch den Kompressor die programmierten Bewegungen geglättet, was zu einem glatteren Verlauf der Bahngeschwindigkeit führt.

Programmierung von Richtungsvektoren

Eine Programmierung der Werkzeugorientierung, die von der Kinematik unabhängig ist, erreicht man durch Programmierung von Richtungsvektoren. NC-Programme mit solchen Richtungsvektoren können mit den Kompressoren COMPON, COMPCURV und COMPCAD abgearbeitet werden.

2.2 5-Achs-Transformation

2.2.1 Kinematische Transformation

Aufgabe der Orientierungstransformation

Aufgabe der Orientierungstransformation ist es, Bewegungen der Werkzeugspitze, die sich aus Orientierungsänderungen ergeben, durch entsprechende Ausgleichsbewegungen der Geometrieachsen zu kompensieren. Die Orientierungsbewegung wird dadurch von der Bewegung auf der Werkstückkontur entkoppelt. Die unterschiedlichen Maschinenkinematiken benötigen jeweils eine eigene Orientierungstransformation.

Einsatzgebiet

Das Bearbeitungspaket "5-Achs-Transformation" ist für Werkzeugmaschinen vorgesehen, die außer über drei Linearachsen X, Y und Z über zwei zusätzliche Rundachsen (Drehung um die Linearachsen) verfügen: Damit kann ein rotationssymmetrisches Werkzeug (Fräser, Laserstrahl) in jedem Punkt des Bearbeitungsraumes beliebig zum Werkstück orientiert werden.

Das Werkstück wird immer im rechtwinkligen Werkstückkoordinatensystem programmiert, eventuell programmierte oder eingestellte Frames drehen und verschieben dieses System auf das Basissystem. Die kinematische Transformation setzt diese Informationen dann in Bewegungsvorgaben der realen Maschinenachsen um.

Die kinematische Transformation benötigt Angaben über den Aufbau (Kinematik) der Maschine, die in Maschinendaten hinterlegt sind.

Die kinematische Transformation wirkt nicht auf Positionierachsen.

2.2.2 Maschinentypen für 5-Achs-Transformation

Maschinen werden abhängig von der Orientierbarkeit von Werkzeug und Werkstück eingeteilt in die Maschinentypen 1, 2 und 3:

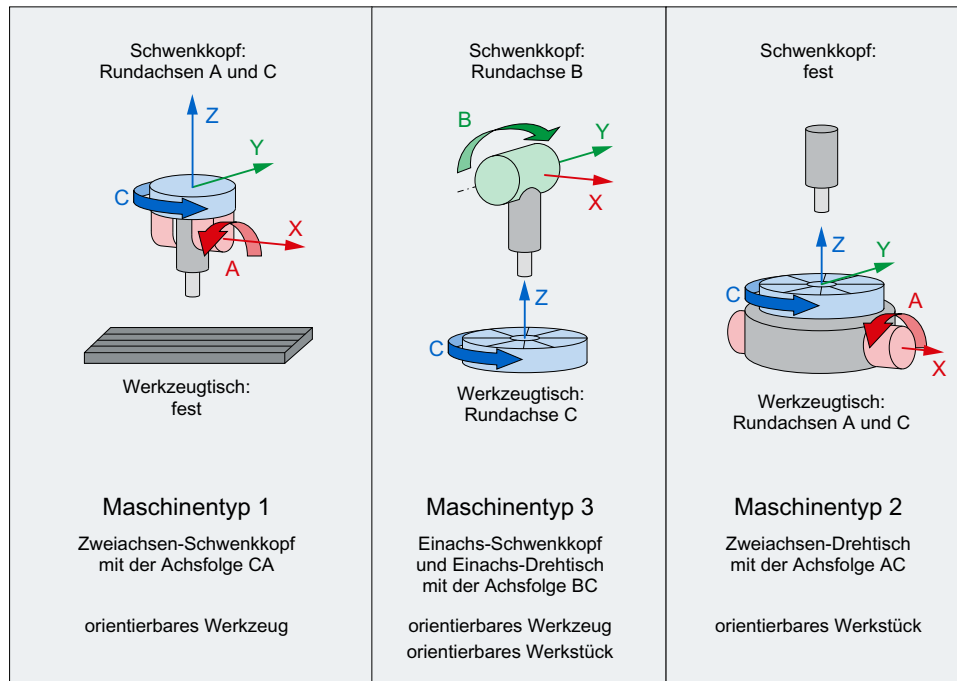


Bild 2-5 Maschinentypen für 5-Achs-Transformation

Gemeinsame Eigenschaften von Maschinentyp 1, 2 und 3

- Die drei Linearachsen bilden ein rechtshändiges kartesisches Koordinatensystem
- Die Orientierungen der Rundachsen sind parallel zu den Linearachsen
- Die Orientierungen der Rundachsen stehen senkrecht aufeinander
- Die Werkzeug-Grundstellung ist in negativer Z-Richtung

Weitere Eigenschaften von Maschinentyp 1 und 2

- Rundachse 1 ist die 4. Maschinenachse der Transformation
- Rundachse 1 verändert die Orientierung von Rundachse 2
- Rundachse 2 ist die 5. Maschinenachse der Transformation
- Rundachse 2 verändert die Orientierung der von Rundachse 1 nicht

Weitere Eigenschaften von Maschinentyp 3

- Rundachse 1 ist die 4. Maschinenachse der Transformation
- Rundachse 1 dreht das Werkzeug
- Rundachse 2 ist die 5. Maschinenachse der Transformation
- Rundachse 2 dreht das Werkstück

Hinweis

Transformationen, die nicht die hier genannten Bedingungen erfüllen, sind in eigenen Kapiteln beschrieben.

2.2.3 Konfiguration einer Maschine für 5-Achs-Transformation

Maximale Anzahl 5-Achs-Transformationen pro Kanal

Aktuell können pro Kanal maximal vier 5-Achs-Transformationen parametrierbar werden.

Transformationstyp

Im Maschinendatum wird der vom Maschinentyp abhängige Transformationstyp eingestellt:

MD24100 \$MC_TRAFO_TYPE_<x> = <Transformationstyp>

mit x = 1, 2, ... maximale Anzahl Transformationen

Achsfolge der Maschine	Maschinentyp 1 (schwenk-/drehbares Werkzeug)	Maschinentyp 2 (schwenk-/drehbares Werkstück)	Maschinentyp 3 (schwenk-/drehbares Werkzeug/Werkstück)
	Transformationstyp	Transformationstyp	Transformationstyp
AB	16	32	48
AC	--- ¹⁾	33	49
BA	18	34	50
BC	--- ¹⁾	35	51
CA	20	--- ¹⁾	--- ¹⁾
CB	21	--- ¹⁾	--- ¹⁾

¹⁾ Eine nicht sinnvolle Kombinationen, da eine Drehung der C-Achse nur einer Drehung des Werkzeugs um seine Längsachse (Symmetrieachse) entspräche.

Kennzeichnung der Achsfolge

Die Kennzeichnung der Achsfolge erfolgt auf folgende Weise:

- AB bedeutet: A ist 4. Achse, B ist 5. Achse der Transformation
- Bei Maschinentyp 3 ist die Schwenkachse des Werkzeugs die 4. Achse und die Drehachse des Werkstücks die 5. Achse der Transformation.

Achszuordnung

Im Maschinendatum wird die Achszuordnung am Eingang der 5-Achs-Transformation legt fest, d.h. welche Achse von der Transformation auf welche Kanalachse abgebildet wird:

MD24110 \$MC_TRAFO_AXES_IN_<x> (Achszuordnung für Transformation x)

mit $x = 1, 2, \dots$ maximale Anzahl Transformationen

Geometrie-Informationen

Für die Berechnung der Achswerte durch die 5-Achs-Transformation sind Angaben über die Geometrie der Maschine notwendig. Sie finden sich in den Maschinendaten (hier für die erste Transformation im Kanal):

- $\$MC_TRAF05_PART_OFFSET_<x>$ (Werkstückorientierte Verschiebung) mit $x = 1, 2, \dots$ maximale Anzahl Transformationen im Kanal
 - Maschinentyp 1:
Vektor vom Maschinenbezugspunkt zum Nullpunkt des Tisches (i. A. Nullvektor)
 - Maschinentyp 2:
Vektor vom letzten Drehgelenk des Tisches zum Nullpunkt des Tisches

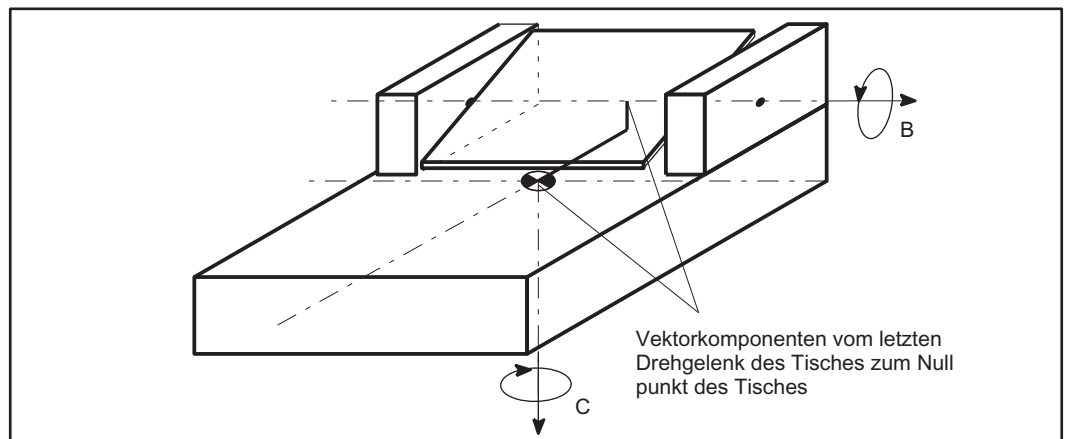
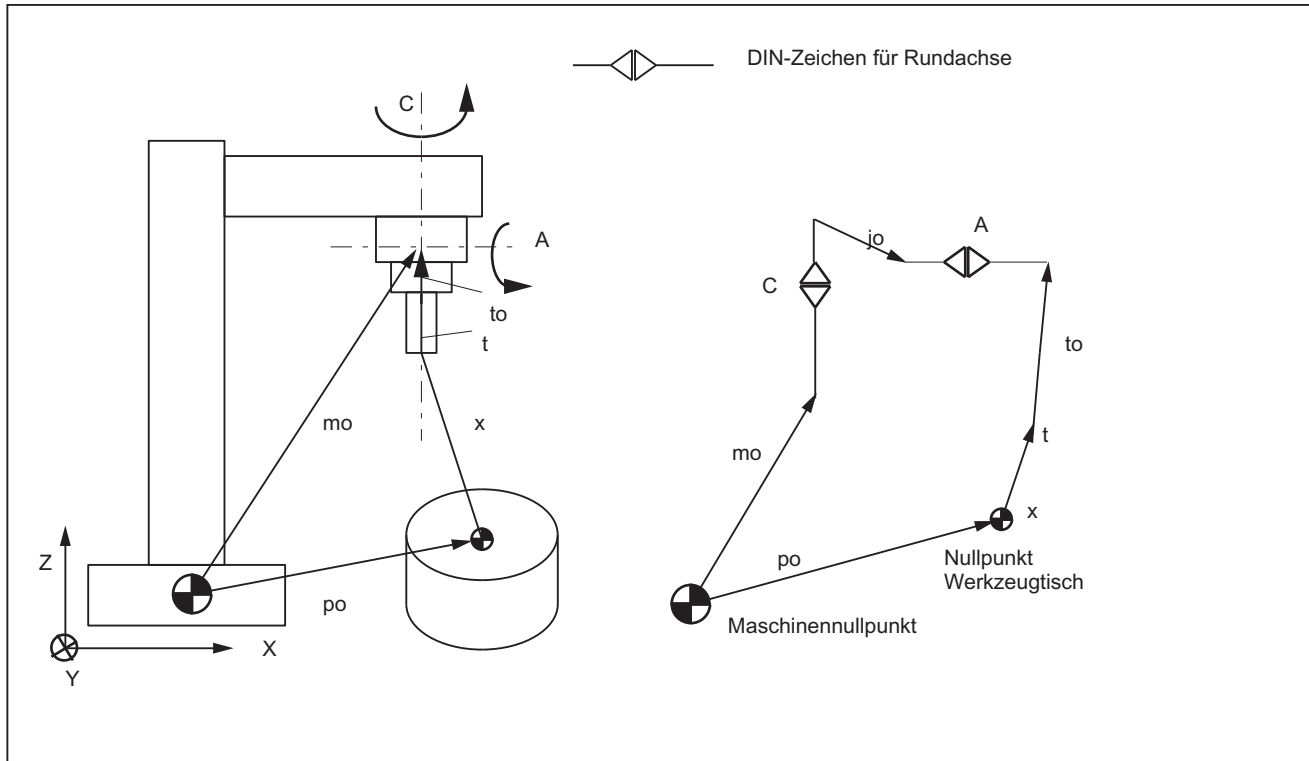


Bild 2-6 Maschinendatum MD24500 $\$MC_TRAF05_PART_OFFSET_1$ für Maschinentyp 2

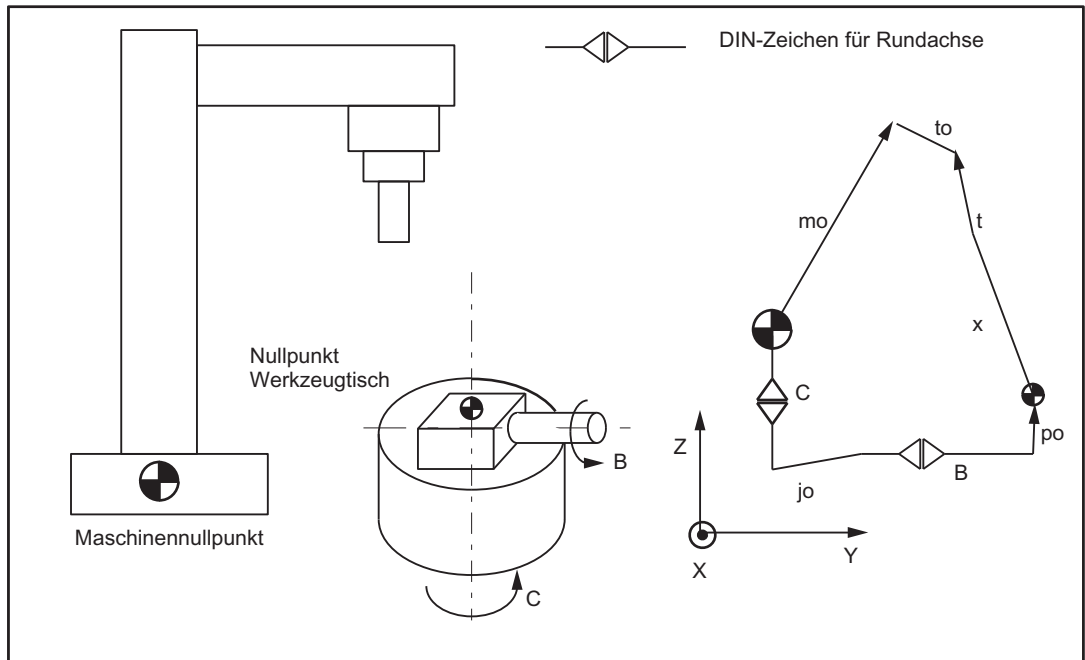
- Maschinentyp 3 (Einachsen-Schwenkkopf und Einachsen-Drehtisch):
Vektor vom Drehgelenk des Drehtisches zum Nullpunkt des Tisches
- $\$MC_TRAF05_JOINT_OFFSET_<x>$ (Vektor des kinematischen Versatzes) mit $x = 1, 2, \dots$ maximale Anzahl Transformationen im Kanal
 - Maschinentyp 1 und 2:
Vektor vom ersten zum zweiten Drehgelenk
 - Maschinentyp 3:
Vektor vom Maschinennullpunkt zum Drehgelenk des Tisches
- $\$MC_TRAF05_ROT_AX_OFFSET_<x>$ (Positionsoffset der Rundachsen 1 / 2 / 3) mit $x = 1, 2, \dots$ maximale Anzahl Transformationen im Kanal

Beispiele



- mo** Positionsvektor im MKS
- po** \$MC_TRAFO5_PART_OFFSET_<x>[0 ..2]
- x** Vektor der programmierten Position im BKS
- t** Werkzeugkorrekturvektor
- to** \$MC_TRAFO5_BASE_TOOL_<x>[0 .. 2]
- jo** MD24560 \$MC_TRAFO5_JOINT_OFFSET_<x>[0 .. 2]

Bild 2-7 Schematische Darstellung einer CA-Kinematik, bewegtes Werkzeug



- mo** Positionsvektor im MKS
- po** \$MC_TRAFO5_PART_OFFSET_<x>[0 ..2]
- x** Vektor der programmierten Position im BKS
- t** Werkzeugkorrekturvektor
- to** \$MC_TRAFO5_BASE_TOOL_<x>[0 .. 2]
- jo** \$MC_TRAFO5_JOINT_OFFSET_<x>[0 .. 2]

Bild 2-8 Schematische Darstellung einer CB-Kinematik, bewegtes Werkstück

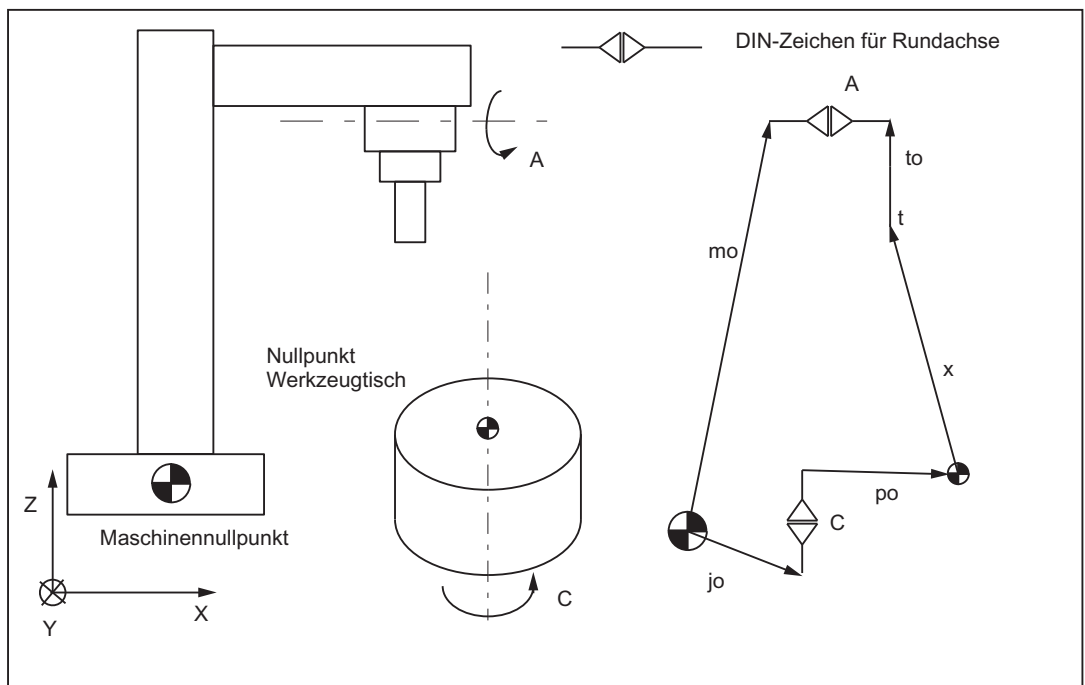


Bild 2-9 Schematische Darstellung einer AC-Kinematik, bewegtes Werkzeug, bewegtes Werkstück

Vorzeichenbehandlung der Rundachsen

Mit dem Maschinendatum wird die Vorzeichenbehandlung der Rundachsen der 5-Achs-Transformation im Kanal eingestellt:

\$MC_TRAFO5_ROT_SIGN_IS_PLUS_<x>[<n>] = <Wert>

mit x = 1, 2, ... maximale Anzahl Transformationen im Kanal

<n>	Bedeutung
0	1. Rundachse
1	2. Rundachse
2	3. Rundachse

<Wert>	Bedeutung
TRUE	Das Vorzeichen wird nicht gedreht. Die Verfahrrichtung ist so, wie in MD32100 \$MA_AX_MOTION_DIR festgelegt.
FALSE	Das Vorzeichen wird gedreht.

Mit dem Maschinendatum wird nicht festgelegt, dass die Drehrichtung der betreffenden Rundachse gedreht werden soll. Sondern es wird angegeben, ob die Rundachse sich bei einer Verfahrbewegung in positiver Verfahrrichtung in mathematisch positiver (gegen den Uhrzeigersinn) oder negativer (im Uhrzeigersinn) Richtung dreht. Die Folge einer Änderung dieses Maschinendatums ist deshalb nicht eine Änderung der Drehrichtung, sondern eine Änderung der Ausgleichsbewegung in den Linearachsen.

Wird allerdings ein Richtungsvektor und damit implizit eine Ausgleichsbewegung vorgegeben, resultiert daraus eine Drehrichtungsänderung der beteiligten Rundachse.

Das Maschinendatum muss deshalb an einer realen Maschine nur dann auf FALSE gesetzt werden, wenn sich die Rundachse bei einer Verfahrbewegung in positiver Verfahrrichtung in mathematisch positiver Richtung (Gegenuhrzeigersinn) dreht.

2.2.4 Werkzeugorientierung

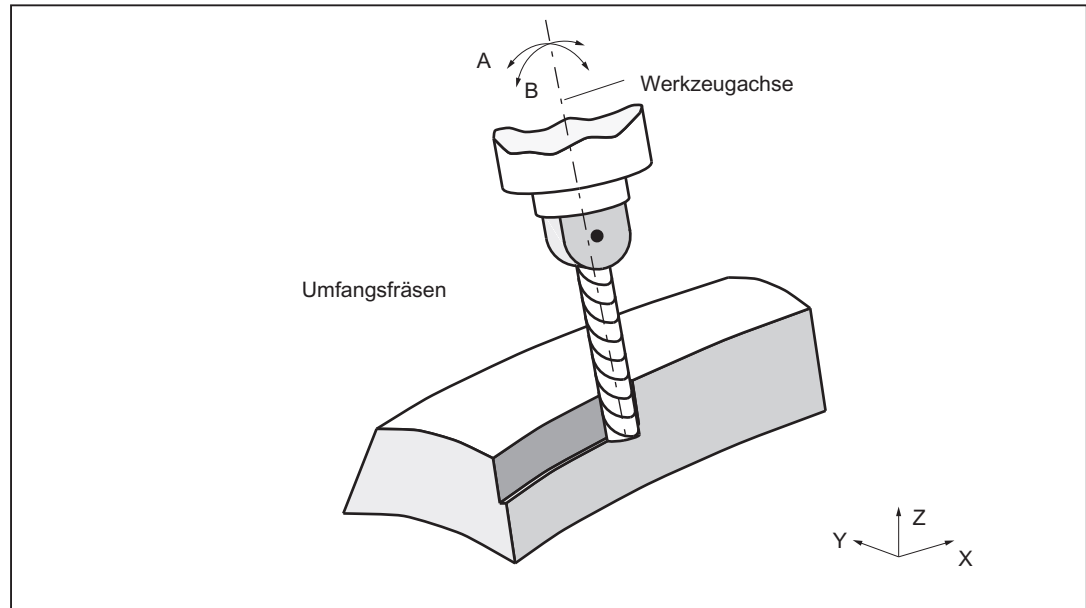


Bild 2-10 Bearbeiten von Werkstücken mit 5-Achs-Transformation

Programmierung

Die Orientierung des Werkzeugs kann in einem Satz direkt durch Angabe der Rundachsen oder indirekt durch Euler-, RPY-Winkel sowie Richtungsvektor programmiert werden. Folgende Möglichkeiten sind vorgesehen:

- direkt als Rundachsen A, B, C
- indirekt bei 5-Achs-Transformation über Euler- oder RPY-Winkel in Grad über A2, B2, C2
- indirekt bei 5-Achs-Transformation über Richtungsvektor A3, B3, C3

Die Bezeichner von Eulerwinkel und Richtungsvektoren sind über Maschinendaten einstellbar:

Eulerwinkel über:

MD10620 \$MN_EULER_ANGLE_NAME_TAB (Name der Eulerwinkel)

Richtungsvektor über:

MD10640 \$MN_DIR_VECTOR_NAME_TAB (Name der Richtungsvektoren)

Die Werkzeugorientierung kann in einem beliebigen Satz stehen. Insbesondere kann sie auch alleine stehen, was zu einer Orientierungsänderung um die in Bezug auf das Werkstück feststehende Werkzeugspitze führt.

Euler oder RPY

Über das folgende Maschinendatum kann zwischen Euler- und RPY-Eingabe umgeschaltet werden:

MD21100 \$MC_ORIENTATION_IS_EULER (Winkeldefinition bei Orientierungsprogrammierung)

Bezug der Orientierung

Die Überführung einer Werkzeugorientierung am Satzanfang in eine Orientierung am Satzende kann auf zwei verschiedene Arten erfolgen:

- im Werkstück-Koordinatensystem mit dem Befehl ORIWKS
- im Maschinen-Koordinatensystem mit dem Befehl ORIMKS

Befehl ORIWKS

Die Werkzeugorientierung wird programmiert im Werkstück-Koordinatensystem und ist damit nicht von der Maschinenkinematik abhängig.

Bei Orientierungsänderung mit raumfester Werkzeugspitze bewegt sich das Werkzeug auf einem Großkreis in der vom Anfangs- zum Endvektor aufgespannten Ebene.

Befehl ORIMKS

Die Werkzeugorientierung wird programmiert im Maschinen-Koordinatensystem und ist damit abhängig von der Maschinenkinematik.

Bei Orientierungsänderung mit raumfester Werkzeugspitze wird zwischen den Rundachpositionen linear interpoliert.

Eine Anwahl der Orientierung erfolgt über die Befehle ORIWKS und ORIMKS der NC-Sprache.

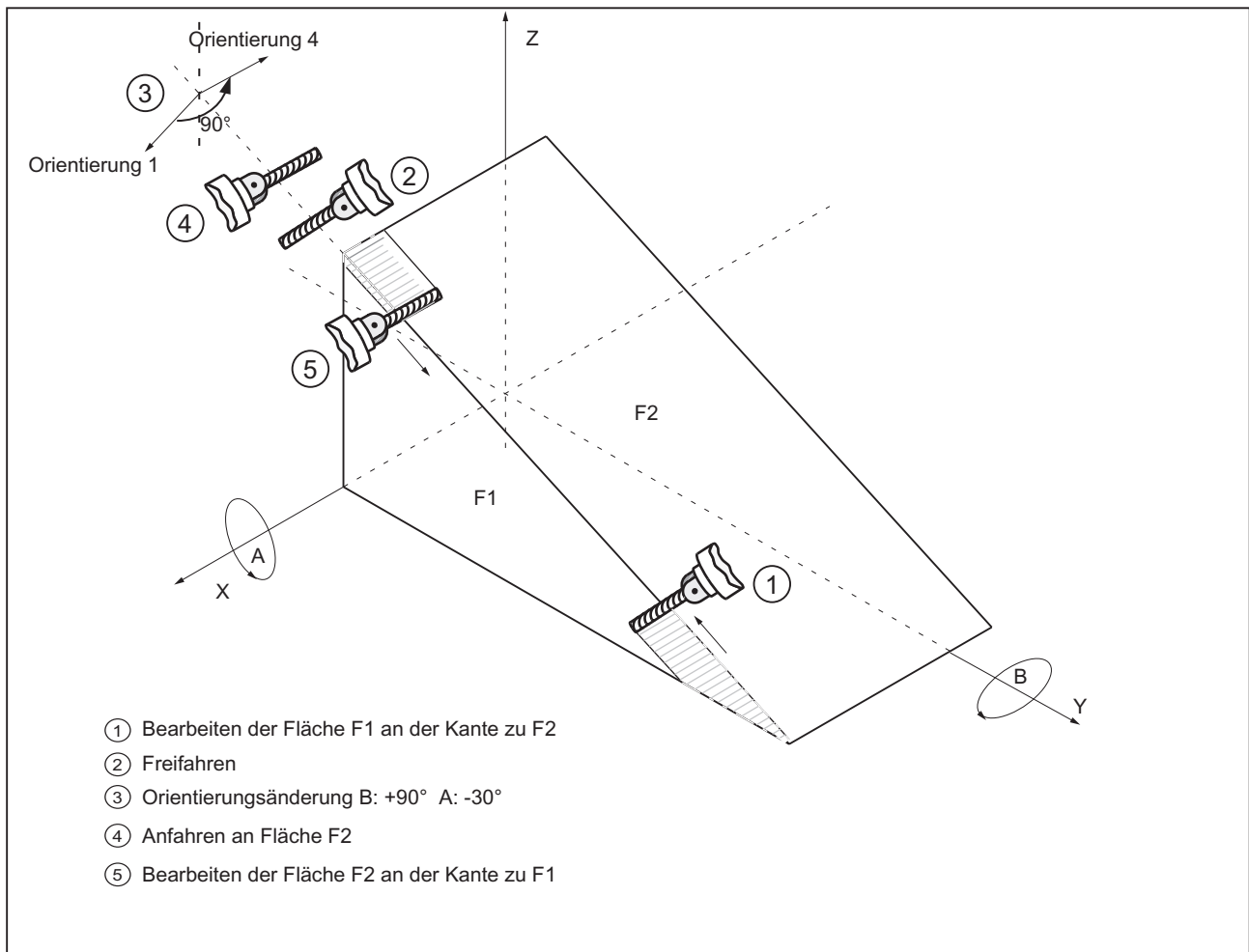


Bild 2-11 Orientierungsänderung des Fräsers bei der Bearbeitung von schrägen Kanten

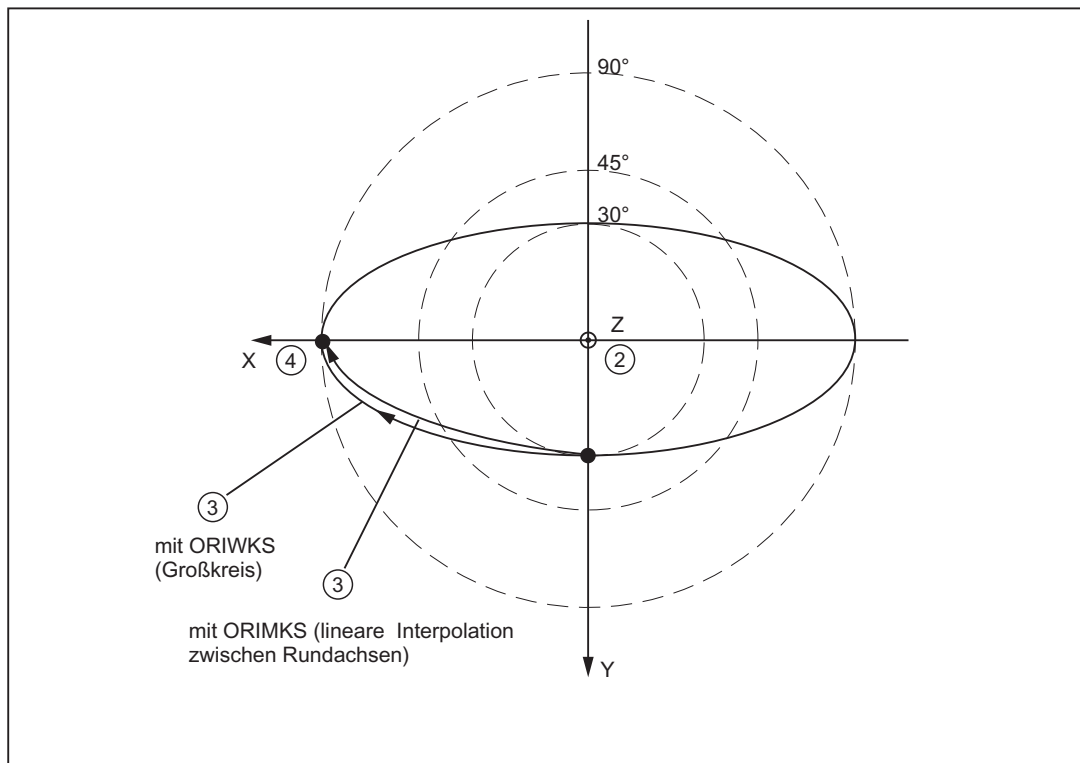


Bild 2-12 Orientierungsänderung bei der Bearbeitung von schrägen Kanten

Grundstellung ist ORIMKS

Über das folgende Maschinendatum kann die Grundstellung geändert werden:

MD20150 MC_GCODE_RESET_VALUES (Löschstellung der G-Gruppen)

MD20150 \$MC_GCODE_RESET_VALUES [24] = 1 ⇒ ORIWKS ist Grundstellung

MD20150 \$MC_GCODE_RESET_VALUES [24] = 2 ⇒ ORIMKS ist Grundstellung

Unzulässige Werkzeugorientierung

Wird die Werkzeugorientierung in Verbindung mit den folgenden Funktionen programmiert, dann kommt es bei Eulerwinkel und Richtungsvektoren zum Alarm 12130 "Unzulässige Werkzeugorientierung" und das NC-Programm hält an (dieser Alarm kann auch in Verbindung mit G331, G332 und G63 auftreten).

- G04: Verweilzeit
- G33: Gewindeschneiden mit konstanter Steigung
- G74: Referenzpunkt anfahren
- G75: Festpunkt anfahren
- REPOSL: Wiederanfahren an die Kontur
- REPOSQ: Wiederanfahren an die Kontur
- REPOSH: Wiederanfahren an die Kontur

Zur Abhilfe lässt sich die Werkzeugorientierung mit Achsendwerten programmieren.

Bei G74 und G75 werden die Alarmer 17630 bzw. 17620 ausgegeben, wenn eine Transformation aktiv ist und die zu verfahrenen Achsen an der Transformation beteiligt sind. Dies gilt unabhängig von einer Orientierungsprogrammierung.

Sind bei aktivem ORIWKS der Start- und Endvektor antiparallel, dann ist für die Orientierungsprogrammierung keine eindeutige Ebene definiert und Alarm 14120 wird ausgelöst.

Wird bei aktiver Werkzeugradiuskorrektur eine Transformationsumschaltung (Einschalten, Ausschalten oder Wechsel der Transformation) vorgenommen, dann wird der Alarm 14400 ausgelöst.

Der umgekehrte Fall, dass die Werkzeugradiuskorrektur bei aktiver Transformation an- oder abgewählt wird, führt zu keiner Alarmmeldung.

Werkzeug-Orientierung mehrfach eingeben

In einem Satz nach DIN 66025 darf jeweils nur eine Werkzeugorientierung eingegeben werden, z. B. mit Richtungsvektoren:

```
| N50           A3=1 B3=1 C3=1
```

Wird die Werkzeugorientierung mehrfach eingegeben, z. B. mit Richtungsvektoren und mit Eulerwinkeln, dann erfolgt die Fehlermeldung 12240 "Kanal X Satz Y Werkzeugorientierung xx mehrfach definiert" und das NC-Teileprogramm hält an:

```
| N60           A3=1 B3=1 C3=1           A2=0 B2=1 C2=3
```

Werkzeugorientierung mittels Orientierungsvektoren

Zusätzlich können für die Änderung des Orientierungsvektors Polynome programmiert werden.

Diese Methode führt an den Rundachsen zu sehr stetigen Geschwindigkeits- und Beschleunigungsänderungen an den Satzübergängen, wenn die Werkzeugorientierung über mehrere Sätze programmiert werden muss.

Die Interpolation von Orientierungsvektoren kann mit Polynomen bis maximal 5. Grades programmiert werden. Die Polynominterpolation von Orientierungsvektoren ist im Kapitel "Polynominterpolation von Orientierungsvektoren" beschrieben.

Hinweis

Weitere Erläuterungen zur Werkzeugorientierung mittels Orientierungsvektoren und deren Nutzung an Werkzeugmaschinen entnehmen Sie bitte:

Literatur:

Funktionshandbuch Grundmaschine, Werkzeugkorrektur, Orientierbare Werkzeugträger (W1)

2.2.5 Singuläre Stellen und ihre Behandlung

Extreme Geschwindigkeitsüberhöhung

Wenn die Bahn in der Nähe eines Pols (Singularität) verläuft, dann kann es vorkommen, dass eine oder mehrere Achsen mit sehr hoher Geschwindigkeit fahren.

Dann wird der Alarm 10910 "Irregulärer Geschwindigkeitsverlauf in einer Bahnachse" ausgelöst. Die programmierte Geschwindigkeit wird soweit reduziert, dass die maximale Achsgeschwindigkeit nicht überschritten wird.

Verhalten am Pol

Das unerwünschte Verhalten schneller Ausgleichsbewegungen lässt sich durch geeignete Wahl der folgenden Maschinendaten beeinflussen (siehe nachfolgendes Bild):

- MD24530 \$MC_TRAFO5_NON_POLE_LIMIT_1 (Definition des Polbereichs für 5-Achstransformation 1)
- MD24630 \$MC_TRAFO5_NON_POLE_LIMIT_2 (Definition des Polbereichs für 5-Achstransformation 2)
- MD24540 \$MC_TRAFO5_POLE_LIMIT_1 (Endwinkeltoleranz b. Interpolation durch Pol f. 5-Achstransform)
- MD24640 \$MC_TRAFO5_POLE_LIMIT_2 (Endwinkeltol. b. Interpolation durch Pol f. 5-Achstransform)

Hinweis

Ab SW-Stand 5.2 ändert sich die Behandlung der Singularitäten: Es gibt nur noch ein relevantes Maschinendatum \$MC_TRAFO5_POLE_LIMIT (siehe Kapitel "Singularitäten der Orientierung (Seite 89)" bzw. Programmierhandbuch Arbeitsvorbereitung).

Definition des Polbereichs für 5-Achstransformation

Dieses Maschinendatum kennzeichnet einen Grenzwinkel für die fünfte Achse der ersten MD24530 \$MC_TRAFO5_NON_POLE_LIMIT_1 oder der zweiten MD24630 \$MC_TRAFO5_NON_POLE_LIMIT_2 5-Achs-Transformation mit folgenden Eigenschaften:

Läuft die Bahn unterhalb dieses Winkels am Pol vorbei, wird durch den Pol gefahren.

Bei der 5-Achs-Transformation spannen die beiden Orientierungsachsen des Werkzeugs ein Koordinatensystem aus Längen- und Breitenkreisen auf einer Kugeloberfläche auf.

Führt bei einer Orientierungsprogrammierung (d. h. der Orientierungsvektor liegt in einer Ebene) die Bahn so dicht am Pol vorbei, dass der mit diesem Maschinendatum definierte Winkel unterschritten wird, dann wird von der vorgegebenen Interpolation in der Weise abgewichen, dass die Interpolation durch den Pol verläuft.

Endwinkeltoleranz bei Interpolation durch Pol für 5-Achstrafo

Dieses Maschinendatum kennzeichnet einen Grenzwinkel für die fünfte Achse der ersten MD24540 \$MC_TRAFO5_POLE_LIMIT_1 oder der zweiten MD24640 \$MC_TRAFO5_POLE_LIMIT_2 5-Achs-Transformation mit folgenden Eigenschaften:

Bei der Interpolation durch den Polpunkt bewegt sich nur die fünfte Achse, die vierte Achse behält ihre Startposition bei. Wird eine Bewegung programmiert, die nicht exakt durch den Polpunkt, aber innerhalb des durch folgendes Maschinendatum gegebenen Bereichs in der Nähe des Pols verlaufen soll, wird von der vorgegebenen Bahn abgewichen, da die Interpolation exakt durch den Polpunkt verläuft:

- MD24530 \$MC_TRAFO5_NON_POLE_LIMIT_1
- MD24630 \$MC_TRAFO5_NON_POLE_LIMIT_2

Dadurch ergibt sich im Endpunkt der vierten Achse (der Polachse) eine Positionsabweichung gegenüber dem programmierten Wert.

Dieses Maschinendatum gibt den Winkel an, um den die Polachse bei der 5-Achs-Transformation vom programmierten Wert abweichen kann, wenn von der programmierten Interpolation auf die Interpolation durch den Polpunkt umgeschaltet wird. Ergibt sich eine größere Abweichung, wird eine Fehlermeldung ausgegeben und die Interpolation nicht durchgeführt.

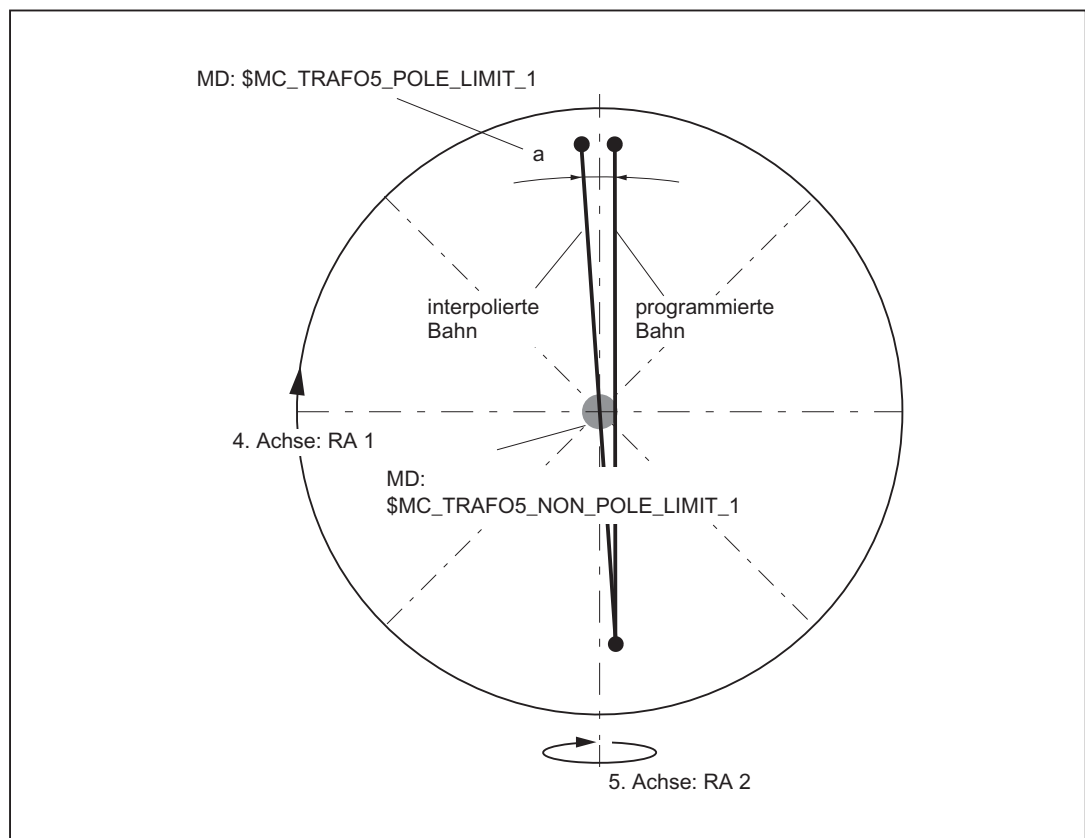


Bild 2-13 5-Achs-Transformation; Orientierungsweg in Polnähe, Beispiel für Maschinentyp 1: 2-Achsen-Schwenkkopf mit der Rundachse RA 1 (4. Achse der Transformation) und der Rundachse RA 2 (5. Achse der Transformation)

Verhalten bei Großkreisinterpolation in Polposition

Mit dem folgenden Maschinendatum kann das Verhalten bei Großkreisinterpolation in Polposition wie folgt eingestellt werden:

MD21108 \$MC_POLE_ORI_MODE

Die Behandlung der Orientierungsänderung bei Großkreisinterpolation wird erst dann definiert, wenn die Startorientierung gleich der Polorientierung ist oder dieser nahe kommt und die Endorientierung des Satzes außerhalb des durch das in folgendem Maschinendatum definierten Toleranzkreises liegt:

- MD24530 \$MC_TRAFO5_NON_POLE_LIMIT_1
- MD24630 \$MC_TRAFO5_NON_POLE_LIMIT_2

In der Polposition ist die Position der Polachse beliebig. Für die Großkreisinterpolation wird aber eine bestimmte Orientierung dieser Achse benötigt.

Das folgende Maschinendatum ist dezimal kodiert:

MD21108 \$MC_POLE_ORI_MODE

Die **Einerstellen** definieren das Verhalten, wenn die Startorientierung exakt mit der Polorientierung übereinstimmt, und die **Zehnerstellen** das Verhalten, wenn die Startorientierung innerhalb des durch das folgende Maschinendatum definierten Toleranzkreises liegt:

- MD24530 \$MC_TRAFO5_NON_POLE_LIMIT_1
- MD24630 \$MC_TRAFO5_NON_POLE_LIMIT_2

Alle Einstellwerte sind in "Kanalspezifische Maschinendaten" beschrieben.

2.3 3- und 4-Achs-Transformationen

3- und 4-Achs-Transformationen sind Sonderformen der 5-Achs-Transformationen bei denen die Werkzeugorientierung nur in der Ebene senkrecht zur Rundachse möglich ist. Sie unterstützen die Maschinentypen 1 mit beweglichem Werkzeug und 2 mit beweglichem Werkstück.

Varianten der 3-Achs- und 4-Achs-Transformation

Maschinentyp	schwenkbar/drehbar	Rundachse parallel zu	Orientierungsebene	Transformationsstyp	Werkzeugorientierung in Nullstellung
1	Werkzeug	X	Y - Z	16	Z
		Y	X - Z	18	
		Z	X - Y	20	Y
		Z	X - Y	21	X
		beliebig	beliebig	24 ¹⁾	beliebig

Maschinen- typ	schwenkbar/ drehbar	Rundachse parallel zu	Orientierungs- ebene	Transformati- onstyp	Werkzeugorien- tierung in Nullstel- lung
2	Werkstück	X	Y - Z	32, 33	Z
		Y	X - Z	34, 35	
		beliebig	beliebig	40 ¹⁾	beliebig
1) Die Rundachse und die Orientierungsebene können so eingestellt werden, dass die Orientierungs- änderung nicht in einer Ebene, sondern auf einem Kegelmantel erfolgt.					

Werkzeugorientierung in Nullstellung

Die Werkzeugorientierung in Nullstellung ist die Stellung des Werkzeugs bei: aktive Arbeitsebene G17 **UND** Stellung der Rundachse 0 Grad

Achszuordnungen

Die drei Linearachsen, die in die Transformation eingehen, werden über die Maschinendaten \$MC_TRAFO_GEOAX_ASSIGN_TAB_n[0..2] und \$MC_TRAFO_AXES_IN_n[0..2] beliebigen Kanalachsen zugeordnet. Für die Zuordnung der Kanalachsen zu den Geometrieachsen für die Transformation muss gelten:

$$\text{\$MC_TRAFO_GEOAX_ASSIGN_TAB_n[0]} = \text{\$MC_TRAFO_AXES_IN_n[0]}$$

$$\text{\$MC_TRAFO_GEOAX_ASSIGN_TAB_n[1]} = \text{\$MC_TRAFO_AXES_IN_n[1]}$$

$$\text{\$MC_TRAFO_GEOAX_ASSIGN_TAB_n[2]} = \text{\$MC_TRAFO_AXES_IN_n[2]}$$

Es sind die jeweiligen Achsen mit dem gleichen Index einander zuzuordnen.

Vorgehen bei der Parametrierung

- Geben Sie den Transformationstyp gemäß vorheriger Tabelle als Maschinendatum ein:
\$MC_TRAFO_TYPE_n
- Ordnen Sie den Geometrieachsen der Transformation Kanalachsen zu.
- Setzen Sie bei 3-Achs-Transformation die Werte für die nicht benötigte Achse:
 - \$MC_TRAFO_GEOAX_ASSIGN_TAB_n[<Geometrieachse>] = 0
 - \$MC_TRAFO_AXES_IN_n[<Geometrieachse>] = 0
\$MC_TRAFO_AXES_IN_n[4] = 0; → es gibt keine 2. rotatorische Achse
- Setzen Sie bei 4-Achs-Transformation für die 3 Linearachsen
 - \$MC_TRAFO_GEOAX_ASSIGN_TAB_n[<Geometrieachse>] = ...
 - \$MC_TRAFO_AXES_IN_n[<Geometrieachse>] = ...
\$MC_TRAFO_AXES_IN_n[4] = 0; → es gibt keine 2. rotatorische Achse

Vollständige Beispiele finden Sie im Kapitel "Beispiel für eine 3- und 4-Achs-Transformation".

2.4 Transformation mit geschwenkter Linearachse

Allgemeine Informationen

Die "Transformation mit schwenkbarer Linearachse" bildet eine eigene Transformationsgruppe. Sie kann eingesetzt werden, wenn eine Kinematik wie im Kapitel "Orientierungstransformation mit schwenkbarer Linearachse (Seite 39)" beschrieben vorliegt:

- Drei kartesischen Linearachsen (X, Y, Z) und zwei orthogonalen Rundachsen (A, B).
- Die Rundachsen sind parallel zu zwei der drei Linearachsen.
- Die erste Rundachse (A) wird von zwei kartesischen Linearachsen bewegt. Sie dreht die dritte Linearachse (Z), welche das Werkzeug bewegt.
- Das Werkzeug ist parallel zur dritten Linearachse (Z) ausgerichtet.
- Die zweite Rundachse (B) dreht das Werkstück.

Zusätzliche Voraussetzung:

- Von der ersten Rundachse (A) darf nur ein sehr kleiner Schwenkbereich (Schwenkbereich $\ll \pm 90^\circ$) überstrichen werden.

Hinweis

Alle im Text verwendeten Achsangaben beziehen sich auf die Bezeichnungen der beispielhaften Maschine im nachfolgenden Bild "Maschine mit schwenkbarer Linearachse Z"

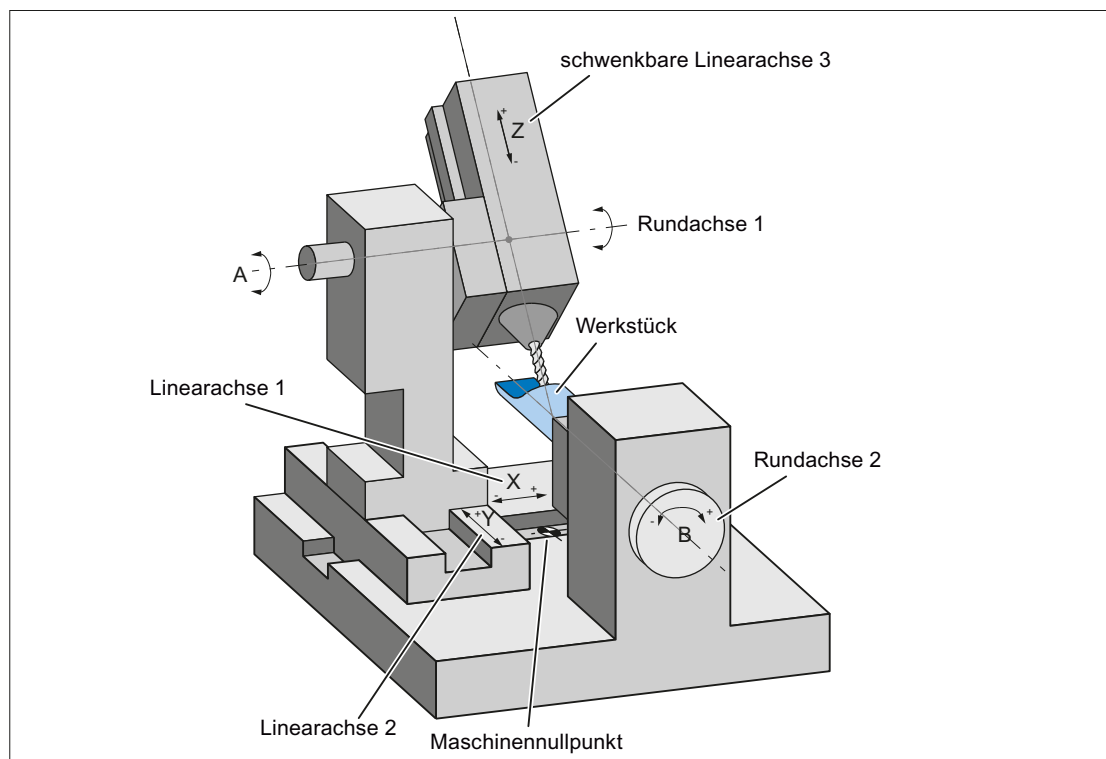


Bild 2-14 Beispiel: Maschine mit schwenkbarer Linearachse Z

Pol

Die Transformation mit schwenkbarer Linearachse besitzt einen Pol bei einer Werkzeugorientierung parallel zur zweiten Rundachse (B). In der Polstellung tritt eine Singularität auf, da die dritte Linearachse (Z) parallel zur Ebene der ersten beiden Linearachsen (X, Y) steht und daher Ausgleichsbewegungen senkrecht zu dieser Ebene nicht mehr möglich sind.

Parametrierungen**Kinematikvarianten**

Die Kinematikvariante der Maschine wird eingestellt im Maschinendatum:

MD24100, ... MD25190 \$MC_TRAFO_TYP_n = <Typ>, mit n = 1, 2, 3, ...

Kinematik			<Typ> Bit 6 - 0
1. Rundachse	2. Rundachse	geschwenkte Linearachse	
A	B	Z	10 00 000
A	C	Y	10 00 001
B	A	Z	10 00 010
B	C	X	10 00 011
C	A	Y	10 00 100
C	B	X	10 00 101

Maschinenkinematik

Die Maschinenkinematik wird für die 1. (\$MC_TRAFO5..._1) und/oder 2. (\$MC_TRAFO5..._2) 5-Achstransformation im Kanal über folgende Maschinendaten eingestellt:

- Vektor (**po**, siehe nachfolgendes Bild) von der zweiten Rundachse zum Nullpunkt des Werkstücktisches:
 - MD24500 \$MC_TRAFO5_PART_OFFSET_1
 - MD24600 \$MC_TRAFO5_PART_OFFSET_2
- Achspositionen der beiden Rundachsen bei Maschinengrundstellung:
 - MD24510 \$MC_TRAFO5_ROT_AX_OFFSET_1
 - MD24610 \$MC_TRAFO5_ROT_AX_OFFSET_2
- Vorzeichen, mit welchem die Rundachspositionen in die Transformation eingehen:
 - MD24520 \$MC_TRAFO5_ROT_SIGN_IS_PLUS_1
 - MD24620 \$MC_TRAFO5_ROT_SIGN_IS_PLUS_2
- Vektor (**jo**) vom Maschinennullpunkt zur zweiten Rundachse:
 - MD24560 \$MC_TRAFO5_JOINT_OFFSET_1
 - MD24660 \$MC_TRAFO5_JOINT_OFFSET_2

2.4 Transformation mit geschwenkter Linearachse

- Vektor (**to**) von der Werkzeugaufnahme (Flansch) zur ersten Rundachse (gemessen bei Maschinengrundstellung):
 - MD24550 \$MC_TRAFO5_BASE_TOOL_1
 - MD24650 \$MC_TRAFO5_BASE_TOOL_2
- Vektor (**ro**) vom Maschinennullpunkt zur ersten Drehachse (gemessen bei Maschinengrundstellung):
 - MD24562 \$MC_TRAFO5_TOOL_ROT_AX_OFFSET_1
 - MD24662 \$MC_TRAFO5_TOOL_ROT_AX_OFFSET_2

Bestimmung der Maschinendatenwerte

Als Hilfe für die Bestimmung der Werte für oben genannte Maschinendaten dienen die zwei folgenden Skizzen zur Verdeutlichung der Zusammenhänge zwischen den Vektoren.

Hinweis

Voraussetzung

Die Maschine wurde so verfahren, dass der Werkzeugaufnahmeflansch mit dem Nullpunkt des Tisches (*) übereinstimmt. Ist dies maschinentechnisch nicht möglich, muss der Vektor **to** um die Abweichungen korrigiert werden.

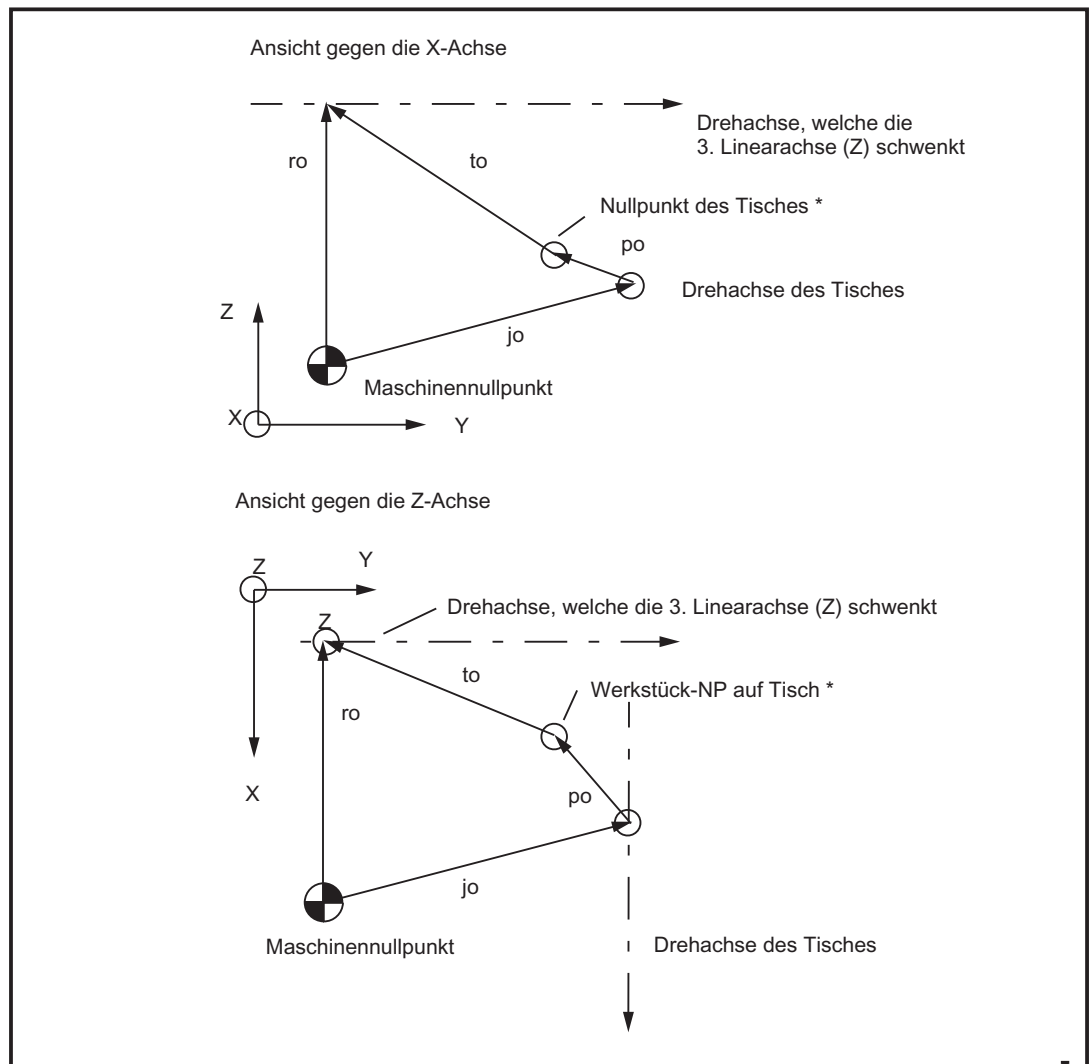


Bild 2-15 Projektionen der in Maschinendaten einzustellenden Vektoren

Hinweis

Für beide Ansichten muss der physisch gleiche Punkt auf der 1. Rundachse (z. B. Schnittpunkt der Werkzeugachse mit der 1. Rundachse) angenommen werden.

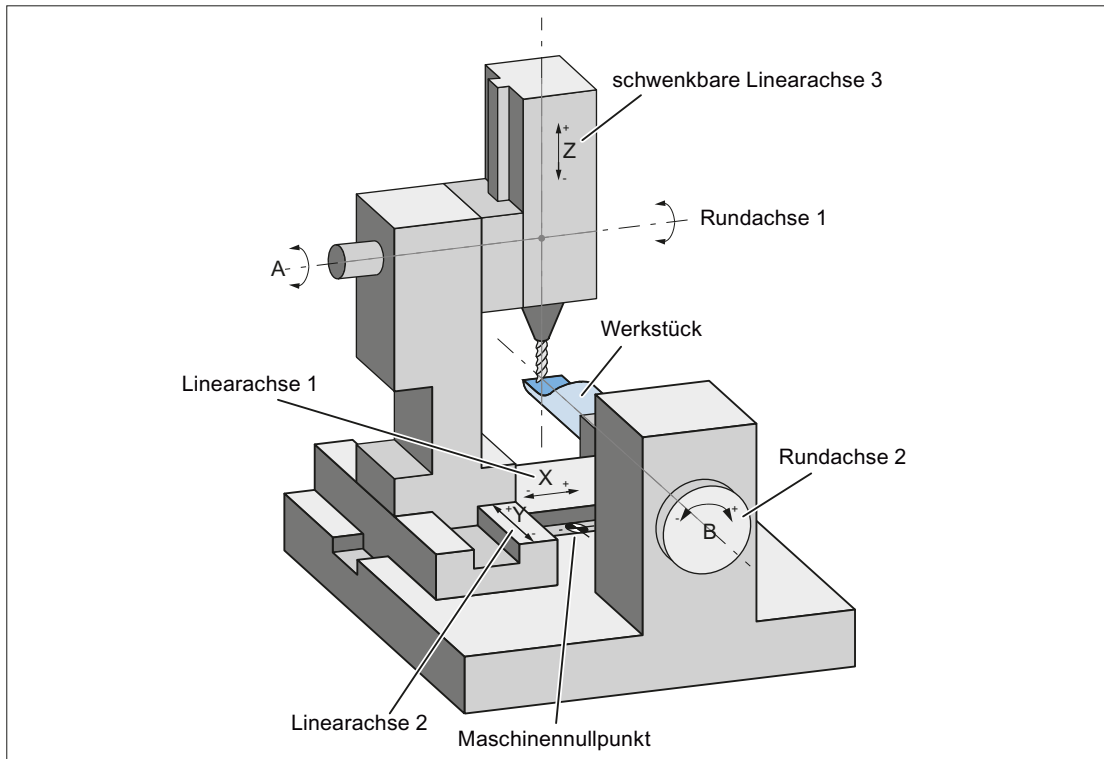


Bild 2-16 Maschine in Nullstellung

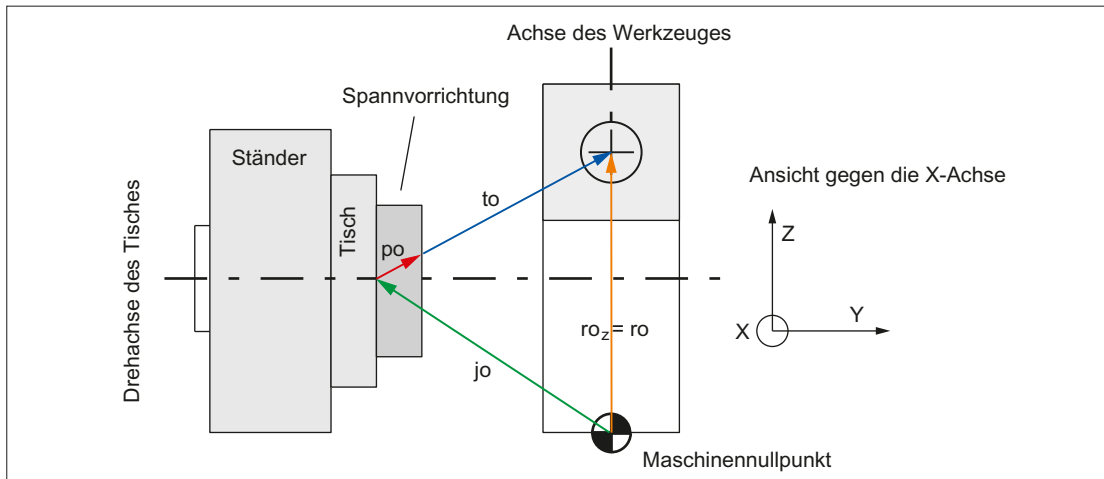


Bild 2-17 Vorderansicht: Vektoren bei Maschine in Nullstellung

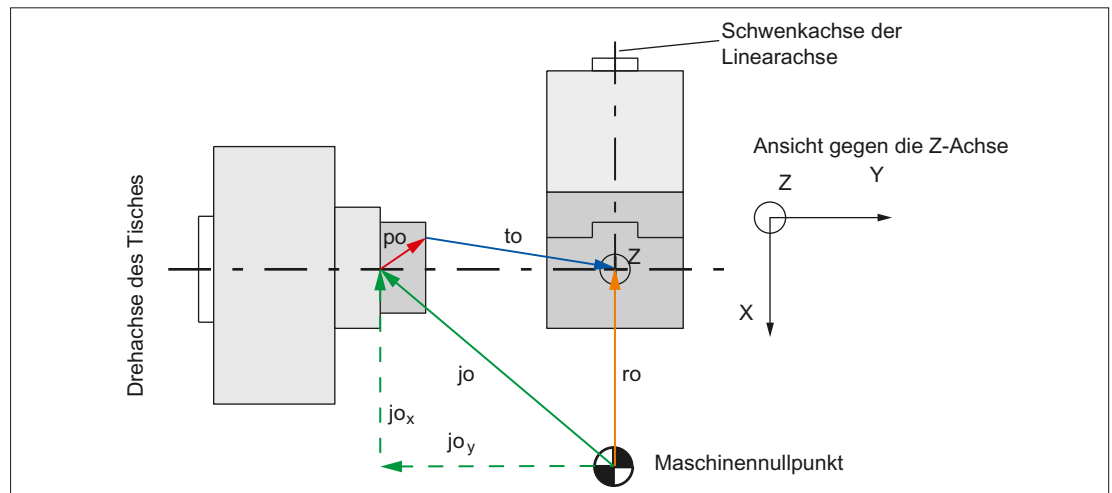


Bild 2-18 Draufsicht: Vektoren bei Maschine in Nullstellung

Ermittlung der Maschinendatenwerte

Führen Sie folgende Schritte durch:

1. Ermitteln Sie, wie im Bild "Vektoren bei Maschine in Nullstellung" im unteren Teil für Vektor jo gezeigt, die x- und y-Anteile für alle Vektoren.
2. Ermitteln Sie, wie im Bild "Vektoren bei Maschine in Nullstellung" im oberen Teil für Vektor ro gezeigt, den z-Anteil für alle Vektoren.
3. Tragen sie die x-, y- und z-Anteile der Vektoren (po , jo , to , ro) in die entsprechenden Maschinendaten ein.

Das Verfahren ist für alle einstellbaren Kinematikvarianten anwendbar.

Hinweis

Bei entsprechender Maschinengeometrie oder Lage des Maschinennullpunktes können sowohl einzelne Komponenten als auch ganze Vektoren zu Null werden.

Programmierung

Das Ein/Ausschalten der Transformation im Teileprogramm oder Synchronaktion ist beschrieben in Kapitel "Programmierung der 3- bis 5-Achs-Transformation (Seite 70)".

Werkzeugorientierung

Für eine Transformation mit schwenkbarer Linearachse gelten bezüglich der Werkzeugorientierung sinngemäß die gleichen Aussagen wie für die 5-Achs-Transformation (siehe Kapitel "Werkzeugorientierung (Seite 51)").

2.5 Kardanischer Fräskopf

2.5.1 Grundlagen kardanischer Fräskopf

Hinweis

Für das Verständnis der Transformation für den kardanischen Fräskopf wird die Kenntnis der allgemeinen 5-Achs-Transformation gemäß Kapitel "5-Achs-Transformation (Seite 44)" vorausgesetzt. Wo im folgenden Kapitel keine spezifischen Aussagen für den kardanischen Fräskopf erfolgen, gelten die Aussagen der allgemeinen 5-Achs-Transformation.

Anwendungen

Die Anwendung des kardanischen Fräskopfes liegt in der Bearbeitung von Konturen von räumlich geformten Teilen bei hohen Vorschubgeschwindigkeiten. Durch die hohe Steifigkeit des Kopfes wird eine hohe Arbeitsgenauigkeit erreicht.

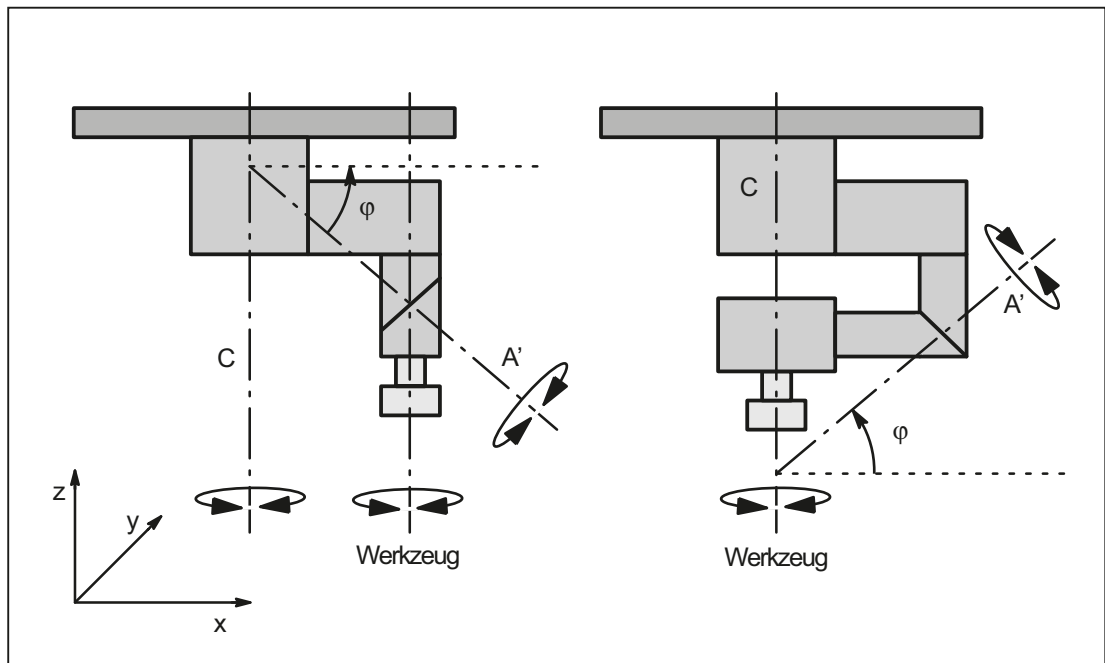


Bild 2-19 Schematische Darstellung kardanischer Fräskopf-Varianten

Projektierung Nutatorwinkel φ

Der Winkel der schrägen Achse kann über ein Maschinendatum projiziert werden:

\$MC_TRAFO5_NUTATOR_AX_ANGLE_1: für die erste Orientierungstransformation

\$MC_TRAFO5_NUTATOR_AX_ANGLE_2: für die zweite Orientierungstransformation

Der Winkel muss im Bereich von 0 Grad bis +89 Grad liegen.

Werkzeugorientierung

Die Werkzeugorientierung in Nullstellung kann wie folgt angegeben werden:

- parallel der 1. rotatorischen Achse oder
- senkrecht dazu und in der Ebene der angegebenen Achsfolge

Kinematiktypen

Die Achsfolgen der rotatorischen Achsen und die Orientierungsrichtung des Werkzeugs in Nullstellung für die vorhandenen Kinematiktypen werden mit Hilfe des folgenden Maschinendatums eingestellt:

\$MC_TRAFO_TYPE_1 ... \$MC_TRAFO_TYPE_10

Benennungsschema der Achsen

Es gilt, wie bei den anderen 5-Achs-Transformationen folgendes:

die rotatorische Achse ...

...A ist parallel zu X: A' liegt unter dem Winkel φ zur X-Achse

...B ist parallel zu Y: B' liegt unter dem Winkel φ zur Y-Achse

...C ist parallel zu Z: C' liegt unter dem Winkel φ zur Z-Achse

Winkeldefinition

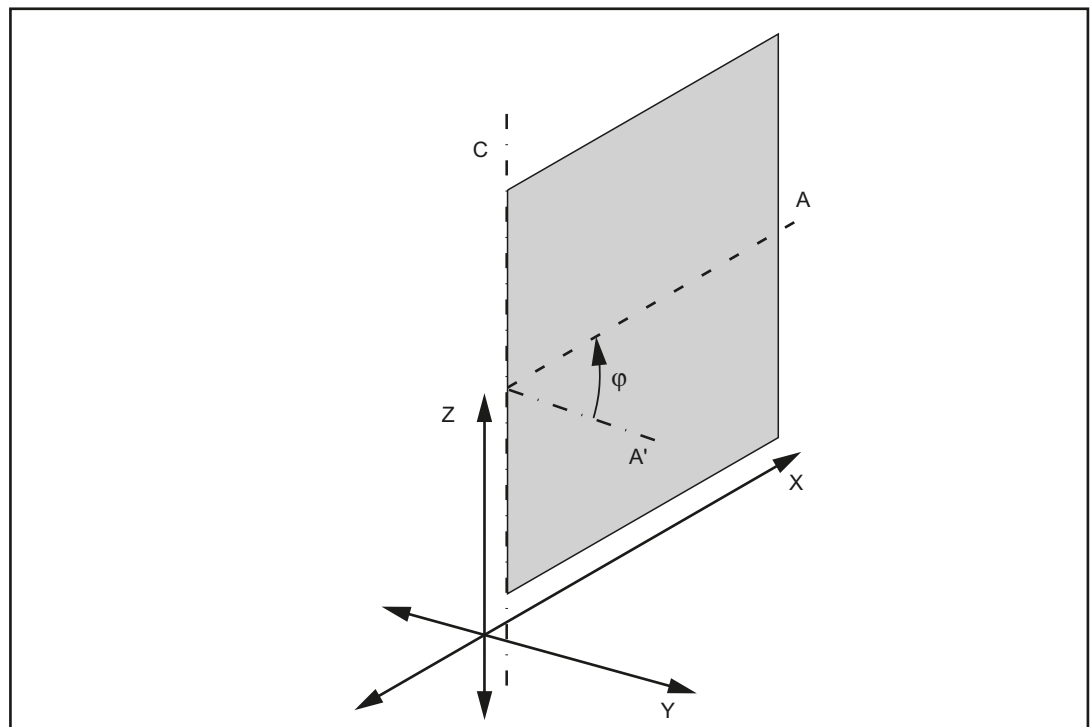


Bild 2-20 Lage der Achse A'

Die Achse A' liegt in der Ebene, die durch die rechtwinkligen Achsen der bezeichneten Achsfolge aufgespannt wird. Ist die Achsfolge CA', so liegt z. B. die Achse A' in der Ebene Z-X. Der Winkel φ ist dann der Winkel zwischen der Achse A' und der X-Achse.

2.5.2 Parametrierung

Einstellung des Transformationstyps

Die Einstellung des Transformationstyps erfolgt über das Maschinendatum des entsprechenden Transformationsdatensatzes:

MD24100, ... MD25190 \$MC_TRAFO_TYP_n, mit n = 1, 2, 3, ...

Bit	<Wert>	Beschreibung
7	1	Aktivierung der Transformation für kardanischen Fräskopf
6 - 5	Bewegliche Komponente	
	00	bewegliches Werkzeug
	01	bewegliches Werkstück
	10	bewegliches Werkzeug und Werkstück
4 - 3	Orientierung des Werkzeugs in Nullstellung	
	00	X-Richtung
	01	Y-Richtung
	10	Z-Richtung
2 - 0	Achsfolge	
	000	AB' bzw. A'B
	001	AC' bzw. A'C
	010	BA' bzw. B'A
	011	BC' bzw. B'C
	100	CA' bzw. C'A
	101	CB' bzw. C'B

Folgende Transformationstypen können eingestellt werden:

Achsfolge: Bit 0 - 2	Bewegliche Komponente: Bit 6 - 5								
	Werkzeug 00			Werkstück 01			Werkzeug/Werkstück 10		
	Nullstellung ¹⁾			Nullstellung ¹⁾			Nullstellung ¹⁾		
	X 00	Y 01	Z 10	X 00	Y 01	Z 10	X 00	Y 01	Z 10
AB' / A'B 000	x	x	-	-	-	-	-	-	-
AC' / A'C 001	x	-	x	-	-	-	-	-	-
BA' / B'A 010	x	x		-	-	-	-	x	x

Achsfolge: Bit 0 - 2	Bewegliche Komponente: Bit 6 - 5								
	Werkzeug 00			Werkstück 01			Werkzeug/Werkstück 10		
	Nullstellung ¹⁾			Nullstellung ¹⁾			Nullstellung ¹⁾		
	X 00	Y 01	Z 10	X 00	Y 01	Z 10	X 00	Y 01	Z 10
BC' / B'C 011	-	x	x	-	x	x	-	-	-
CA' / C'A 100	x	-	x	-	-	-	-	-	-
CB' / C'B 101	-	x	x	-	-	-	-	-	-
1) Orientierung des Werkzeugs in Nullstellung: Bit 3 - 4 x: Transformationstyp kann eingestellt werden -: Transformationstyp kann nicht eingestellt werden									

Aktive Arbeitsebene

Die Werkzeugorientierung in Nullstellung kann nicht nur in Z-Richtung eingestellt werden. Daher ist darauf zu achten, dass die aktive Arbeitsebene so eingestellt wird, dass die Werkzeuglängenkorrektur in Richtung der Werkzeugorientierung wirksam wird.

Die aktive Arbeitsebene sollte immer die Ebene sein, nach der die Werkzeugorientierung in Nullstellung eingestellt wird.

Sonstige Einstellungen

Die Geometrieinformationen für die Berechnung der Achswerte durch die Transformation für den kardanischen Fräskopf werden analog wie bei den anderen 5-Achs-Transformationen eingestellt.

2.5.3 Verfahren des kardanischen Fräskopfes im JOG-Betrieb

JOG

Im JOG-Betrieb können die linearen Achsen normal verfahren werden. Es ist jedoch schwierig, durch Verfahren der linearen Achsen die Orientierung korrekt einzustellen.

2.6 Programmierung der 3- bis 5-Achs-Transformation

Einschalten

Die 3- bis 5-Achs-Transformationen einschließlich der Transformationen mit geschwenkter Linearachse und kardanischem Fräskopf, werden mit dem Befehl `TRAORI (<Trafo-Nr>)` eingeschaltet. Mit dem Einschalten der Transformation wird das NC/PLC-Nahtstellensignal gesetzt:

DB21, ... DBX33.6 = 1 (Transformation aktiv)

Ausschalten

Mit dem Befehl `TRAFOOF` wird die gerade aktive 3- bis 5-Achs-Transformation ausgeschaltet. Mit dem Ausschalten der Transformation wird das NC/PLC-Nahtstellensignal rückgesetzt:

DB21, ... DBX33.6 = 0 (Transformation inaktiv)

Umschalten

Ist im Kanal bereits eine Transformation aktiv, kann mit dem Befehl `TRAORI (<Trafo-Nr>)` mit neuer Transformationsnummer auf eine andere Transformation umgeschaltet werden.

Reset / Programmende

Das Steuerungsverhalten nach Hochlauf, Programmende- oder NC-Reset wird eingestellt im Maschinendatum:

MD20110 \$MC_RESET_MODE_MASK, Bit 7 = <Wert>

<Wert>	Bedeutung
0	Grundstellung für aktive Transformation nach Reset / Programmende laut \$MC_TRAFO_RESET_VALUE
1	Die aktive Transformation bleibt über Reset / Programmende erhalten

Option

Die Funktion "5-Achs-Transformation" ist, einschließlich ihrer Sonderformen, eine Option.

Literatur

Eine ausführliche Beschreibung der Maschinendaten findet sich in:

Listenhandbuch Ausführliche Maschinendatenbeschreibung

2.7 Generische 5-Achs-Transformation und Varianten

2.7.1 Funktionalität

Funktionsumfang

Die generische 5-Achstransformation deckt mit ihrem Funktionsumfang die realisierten 5-Achstransformationen (siehe Kapitel "5-Achs-Transformation (Seite 44)") für rechtwinklig angeordnete Rundachsen sowie die Transformationen für den Kardanischen Fräskopf (eine Rundachse parallel zu einer Linearachse, die zweite Rundachse in einem beliebigen Winkel dazu, siehe Kapitel "Kardanischer Fräskopf (Seite 66)") ab.

Anwendungsbereich

In bestimmten Fällen kann die herkömmliche Transformation Maschinenungenauigkeiten nicht kompensieren, z. B. wenn:

- die Rundachsen nicht exakt senkrecht aufeinander stehen oder
- eine oder beide Rundachsen nicht exakt parallel zu den Linearachsen stehen.

In solchen Fällen kann die generische 5-Achs-Transformation zu besseren Ergebnissen führen.

Programmierbeispiel

für die generische 5-Achs-Transformation ist im Kapitel "Beispiel für generische 5-Achstransformation" dargestellt.

Aktivierung

Eine generische 5-Achstransformation kann wie jede andere Orientierungstransformation auch mit dem Befehl `TRAORI ()` bzw. `TRAORI (n)` aktiviert werden (wobei `n` die Nummer der Transformation ist). Zusätzlich kann beim Aufruf in drei weiteren Parametern die Grundorientierung übergeben werden, z. B. `TRAORI(1, 1.1, 1.5, 8.9)`.

Die Abwahl erfolgt implizit bei Anwahl einer anderen Transformation oder explizit durch `TRAFOOF`.

2.7.2 Beschreibung der Maschinenkinematik

Maschinentypen

Bei der generischen 5-Achstransformation wird ebenso wie bei den bisherigen 5-Achstransformationen zwischen den folgenden drei Varianten unterschieden:

1. Maschinentyp: Drehbares Werkzeug
Beide Rundachsen verändern die Orientierung des Werkzeugs. Die Orientierung des Werkstücks ist fest.
2. Maschinentyp: Drehbares Werkstück
Beide Rundachsen verändern die Orientierung des Werkstücks. Die Orientierung des Werkzeugs ist fest.
3. Maschinentyp: Drehbares Werkzeug und drehbares Werkstück jeweils eine Rundachse verändert die Orientierung von Werkzeug und Werkstück.

Konfigurationen

Die Maschinenkonfigurationen sind wie bisher (siehe Kapitel "Konfiguration einer Maschine für 5-Achs-Transformation (Seite 46)") in den folgenden Maschinendaten festgelegt:

\$MC_TRAFO_TYPE_1, ..., _8

Für die generische 5-Achstransformation wurden zusätzliche Typen eingeführt:

Tabelle 2-1 Übersicht über die Maschinentypen für die generische 5-Achs-Transformation

Maschinentyp	1	2	3
schwenk-/drehbar:	Werkzeug	Werkstück	Werkzeug/Werkstück
Trafotypen	24	40	56

Rundachsen-Richtung

Die Richtung der Rundachsen wird durch die folgenden Maschinendaten festgelegt:

\$MC_TRAFO5_AXIS1_n (1. Rundachse) und

\$MC_TRAFO5_AXIS2_n (2. Rundachse)

n ist dabei 1 oder 2 für die erste bzw. zweite 5-Achstransformation im System. Die genannten Maschinendaten sind Felder mit drei Werten, mit denen die Achsrichtung vektoriell beschrieben wird (analog zur Beschreibung der Rundachsen beim orientierbaren Werkzeugträger). Der Betrag der Vektoren ist unerheblich; Bedeutung hat nur die definierte Richtung.

Beispiel:

1. Rundachse ist A-Achse (parallel zur x-Richtung):

MD24570 \$MC_TRAFO5_AXIS1_1[0] = 1.0 (Richtung 1. Rundachse)

MD24570 \$MC_TRAFO5_AXIS1_1[1] = 0.0

MD24570 \$MC_TRAFO5_AXIS1_1[2] = 0.0

2. Rundachse ist B-Achse (parallel zur y-Richtung):

MD24572 \$MC_TRAFO5_AXIS2_1[0] = 0.0 (Richtung 2. Rundachse)

MD24572 \$MC_TRAFO5_AXIS2_1[1] = 1.0

MD24572 \$MC_TRAFO5_AXIS2_1[2] = 0.0

2.7.3 Generische Orientierungstransformations-Varianten

Erweiterung

Die generische Orientierungstransformation für 5-Achs-Transformationen wird für die 3- und 4-Achstransformation mit den folgenden Varianten erweitert:

Variante 1

4-Achstransformationen

Eine 4-Achstransformation wird dadurch gekennzeichnet, dass nur die erste Rundachse als Eingangssachse der Transformation vorhanden ist. Es gilt:

MD24110 \$MC_TRAFO_AXES_IN_1[4] = 0 (Achszuordnung für Transformation 1) bzw.

MD24210 \$MC_TRAFO_AXES_IN_2[4] = 0 (Achszuordnung für Transformation 2)

Variante 2

3-Achstransformationen

Bei einer 3-Achstransformation fehlt zusätzlich eine der Geometrieachsen, indem eine Null im Feld eingetragen wird:

MD24120 \$MC_TRAFO_GEOAX_ASSIGN_TAB_1[n] (Zuordnung GEOachse zu Kanalachse für Transformation 1) bzw.

MD24220 \$MC_TRAFO_GEOAX_ASSIGN_TAB_2[n] (Zuordnung GEOachse zu Kanalachse für Transformation 2)

Transformation-Typen

Die beiden Varianten der generischen 3- bzw. 4.-Achstransformation, werden durch folgenden Transformationstypen beschrieben:

- 3-bzw. 4-Achstransformation mit drehbarem Werkzeug
\$MC_TRAFO_TYPE_n = 24
- 3-bzw. 4-Achstransformation mit drehbarem Werkstück
\$MC_TRAFO_TYPE_n = 40

Bei den herkömmlichen 3- bzw. 4-Achs-Transformationen legte der Transformationstyp außer der Lage der Drehachse auch die Werkzeuggrundorientierung fest, die dann nicht mehr beeinflusst werden konnte.

Auswirkungen auf die Orientierungen

Die generische 3- bzw. 4-Achstransformation wirkt sich auf die verschiedenen Orientierungen wie folgt aus:

Die resultierende Werkzeugorientierung wird nach der für die generische 5-Achs-Transformation vorgeschriebenen Hierarchie festgelegt.

Priorität:

- hoch: Programmierte Orientierung,
- mittel: Werkzeugorientierung und
- niedrig: Basisorientierung

Insbesondere werden hierbei folgende Orientierungen berücksichtigt:

- Eine gegebenenfalls programmierte Werkzeugorientierung
- Eine durch orientierbare Werkzeugträger modifizierte Werkzeuggrundorientierung

Hinweis

Weitere Informationen zur programmierbaren Werkzeugorientierung und zur Werkzeuggrundorientierung entnehmen Sie Bitte:

Literatur:

Funktionshandbuch Grundmaschine, Werkzeugkorrektur, Orientierbare Werkzeugträger (W1)

Programmierhandbuch Grundlagen

Gegenüberstellung

Gegenüber der bereits vorhandenen 3- und 4-Achs-Transformationen aus Kapitel "3- und 4-Achs-Transformation" sind die folgenden Unterschiede zu beachten:

- Lage der Drehachse:
 - Kann beliebig sein
 - Muss nicht parallel zu einer Linearachse sein
- Richtung der Drehachse
 - Muss mit den folgenden Maschinendaten definiert werden:
MD24570 \$MC_TRAFO5_AXIS1_1[n] (Richtung 1. Rundachse) bzw.
MD24670 \$MC_TRAFO5_AXIS1_2[n] (Richtung 1. Rundachse)
- Grundorientierung des Werkzeugs
 - Muss mit den folgenden Maschinendaten definiert werden:
MD24574 \$MC_TRAFO5_BASE_ORIENT_1[n] (Werkzeuggrundorientierung) bzw.
MD24674 \$MC_TRAFO5_BASE_ORIENT_2[n] (Werkzeuggrundorientierung)
- Anwahl einer generischen 3-/4-Achstransformation
 - Kann wie bei der generischen 5-Achs-Transformation optional eine Werkzeugorientierung übergeben werden.

2.7.4 Daten orientierbarer Werkzeugträger parametrieren

Anwendung

Maschinentypen, bei denen der Tisch oder das Werkzeug drehbar sind, können alternativ als echte 5-Achs-Maschinen oder als konventionelle Maschinen mit orientierbarem Werkzeugträger betrieben werden. In beiden Fällen wird die Kinematik der Maschine durch gleiche Daten bestimmt, die bisher aufgrund der unterschiedlichen Parameter, bei orientierbaren Werkzeugträger über Systemvariable und bei Transformationen mittels Maschinendaten doppelt eingegeben werden mussten. Mit den neuen Trafotyp 72 kann festgelegt werden, dass diese beiden Maschinentypen auf identische Daten zugreifen.

Trafotyp 72

Mit den folgenden Maschinendaten wird für den Trafotyp 72 eine generische 5-Achs-Transformation festgelegt, deren Kinematikdaten aus denen eines orientierbaren Werkzeugträgers gelesen werden:

MD24100 \$MC_TRAFO_TYPE_1 (Definition der Transformation 1 im Kanal) bzw.

MD24200 \$MC_TRAFO_TYPE_2 (Definition der Transformation 2 im Kanal)

Aus dessen Nummer werden die Daten über die Maschinendaten MD24582 \$MC_TRAFO5_TCARR_NO_1 (TCARR-Nummer für die 1. 5-Achs-Trafo) für die erste bzw. MD24682 \$MC_TRAFO5_TCARR_NO_2 (TCARR-Nummer für die 2. 5-Achs-Trafo) für die zweite Orientierungstransformation bereitgestellt. Der entsprechende Trafotyp ergibt sich dann aus den Inhalt vom Kinematiktyp mit dem Parameter \$TC_CARR23, siehe nachfolgende Tabelle.

Tabelle 2-2 Maschinentypen für die generische 5-Achs-Transformation

Maschinentyp	1	2	3	4
schwenk-/drehbar:	Werkzeug	Werkstück	Werkzeug/Werkstück	Typ 3 oder orientierbarer Werkzeugträger
Kinematiktyp:	T	P	M	T, P, M
Trafotyp:	24	40	56	72 aus Inhalt \$TC_CARR23

Hinweis

Eine Übernahme erfolgt nur dann, wenn auch der betreffende orientierbare Werkzeugträger vorhanden ist, und der Wert von \$TC_CARR23 einen gültigen Eintrag für den Kinematiktyp nur M, P, T in Groß- oder Kleinschreibung enthält.

Die in den nachfolgenden Tabellen dargestellten Transformationsmaschinendaten der ersten Orientierungstransformation gelten genauso für die zweite Orientierungstransformation. Alle weiteren Maschinendaten, die das Verhalten der Transformation beeinflussen können und in den folgenden Tabellen **nicht** aufgeführt sind, bleiben weiterhin gültig und wirksam:

MD24110/MD24210 \$MC_TRAFO_AXES_IN_1/2 (Achszuordnung für Transformation) oder MD24574/MD24674 \$MC_TRAFO5_BASE_ORIENT_1/2 (Werkzeuggrundorientierung)

Wenn in den nachfolgenden Tabellen bei den Parametern des orientierbaren Werkzeugträgers in Klammern ein zweiter additiver Parameter aufgeführt ist (z. B. \$TC_CARR24 (+ \$TC_TCARR64)), so ist die Summe beider Werte nur dann wirksam, wenn bei der Übernahme der Daten aus dem orientierbaren Werkzeugträger die Feinverschiebung gemäß folgendem Settingdatum aktiv ist:

SD42974 \$SC_TOCARR_FINE_CORRECTION = TRUE (Feinverschiebung TCARR ein/aus)

Aktivierung

Die für eine Transformation maßgebenden Parameterwerte eines orientierbaren Werkzeugträgers werden jeweils im Teileprogramm mit NEWCONF wirksam gesetzt. Alternativ dazu, können die betreffenden Maschinendaten für den Trafotyp 72 über die HMI-Bedienoberfläche wirksam gesetzt werden.

Zuordnungen für alle Trafotypen

Die Zuordnungen zwischen den Werkzeugträgerdaten zur Beschreibung der linearen Offsets und den entsprechenden Maschinendaten der kinematischen Transformationen sind vom Transformationstyp abhängig. Folgende Zuordnung aller übrigen Parameter ist für alle drei möglichen Transformationstypen gleich:

Zuordnung für alle Transformationstypen gemeinsam gleich			
MD24100 \$MC_TRAFO_TYPE_1 (Definition der Transformation 1 im Kanal)	24	\$TC_CARR23 =	T
	40		P
	56		M
MD24570 \$MC_TRAFO5_AXIS1_1[0] (Richtung 1. Rundachse)		\$TC_CARR7	
MD24570 \$MC_TRAFO5_AXIS1_1[1]		\$TC_CARR8	
MD24570 \$MC_TRAFO5_AXIS1_1[2]		\$TC_CARR9	
MD24572 \$MC_TRAFO5_AXIS2_1[0] (Richtung 2. Rundachse)		\$TC_CARR10	
MD24572 \$MC_TRAFO5_AXIS2_1[1]		\$TC_CARR11	
MD24572 \$MC_TRAFO5_AXIS2_1[2]		\$TC_CARR12	
MD24510 \$MC_TRAFO5_ROT_AX_OFFSET_1[0] (Positionsoffset der Rundachsen 1/2/3 für die 5-Achstransfor. 1)		\$TC_CARR24 (+ \$TC_TCARR64)	

Zuordnung für alle Transformationstypen gemeinsam gleich	
MD24510 \$MC_TRAFO5_ROT_AX_OFFSET_1[1]	\$TC_CARR25 (+ \$TC_TCARR65)
MD24520 \$MC_TRAFO5_ROT_SIGN_IS_PLUS_1[0] (Vorzeichen der Rundachse 1/2/3 für die 5-Achstransformation 1)	TRUE*
MD24520 \$MC_TRAFO5_ROT_SIGN_IS_PLUS_1[1]	TRUE*

*) Die Maschinendaten MD24520/MD24620 \$MC_TRAFO5_ROT_SIGN_IS_PLUS_1/2 sind redundant. Sie dienen dazu, den Drehsinn der zugeordneten Rundachse zu invertieren. Das gleiche Ziel kann aber auch erreicht werden, wenn die Richtung des Achsvektors \$MC_TRAFO5_AXIS1/2_1/2 umgekehrt wird. Beim orientierbaren Werkzeugträger existiert deshalb kein entsprechender Parameter. Damit dieser Fall eindeutig beschreiben wird, muss das folgende Maschinendatum ignoriert werden:

MD24520/MD24620 TRAF05_ROT_SIGN_IS_PLUS_1/2

Zuordnungen für Trafotyp 24

Vom Transformationstyp 24 abhängige Zuordnungen der Werkzeugträgerdaten

Transformationstyp "T" (entspricht MD24100 \$MC_TRAFO_TYPE_1 = 24)	
MD24500 \$MC_TRAFO5_PART_OFFSET_1[0] (Verschiebungsvektor der 5-Achstransformation 1)	\$TC_CARR1 (+\$TC_TCARR41)
MD24500 \$MC_TRAFO5_PART_OFFSET_1[1]	\$TC_CARR2 (+\$TC_TCARR42)
MD24500 \$MC_TRAFO5_PART_OFFSET_1[2]	\$TC_CARR3 (+\$TC_TCARR43)
MD24560 \$MC_TRAFO5_JOINT_OFFSET_1[0] (Vektor des kinematischen Versatzes der 5-Achstransformation 1)	\$TC_CARR4 (+\$TC_TCARR44)
MD24560 \$MC_TRAFO5_JOINT_OFFSET_1[1]	\$TC_CARR5 (+\$TC_TCARR45)
MD24560 \$MC_TRAFO5_JOINT_OFFSET_1[2]	\$TC_CARR6 (+\$TC_TCARR46)
MD24550 \$MC_TRAFO5_BASE_TOOL_1[0] (Vektor d. Basiswerkzeugs b. Aktivier. der 5-Achstransform. 1)	\$TC_CARR15 (+\$TC_TCARR55)
MD24550 \$MC_TRAFO5_BASE_TOOL_1[1]	\$TC_CARR16 (+\$TC_TCARR56)
MD24550 \$MC_TRAFO5_BASE_TOOL_1[2]	\$TC_CARR17 (+\$TC_TCARR57)

Zuordnungen für Trafotyp 40

Vom Transformationstyp 40 abhängige Zuordnungen der Werkzeugträgerdaten

Transformationstyp "P" (entspricht MD24100 \$MC_TRAFO_TYPE_1 = 40)	
MD24550 \$MC_TRAFO5_BASE_TOOL_1[0]	\$TC_CARR4 (+\$TC_TCARR44)
MD24550 \$MC_TRAFO5_BASE_TOOL_1[1]	\$TC_CARR5 (+\$TC_TCARR45)
MD24550 \$MC_TRAFO5_BASE_TOOL_1[2]	\$TC_CARR6 (+\$TC_TCARR46)
MD24560 \$MC_TRAFO5_JOINT_OFFSET_1[0]	\$TC_CARR15 (+\$TC_TCARR55)
MD24560 \$MC_TRAFO5_JOINT_OFFSET_1[1]	\$TC_CARR16 (+\$TC_TCARR56)

Transformationstyp "P" (entspricht MD24100 \$MC_TRAFO_TYPE_1 = 40)	
MD24560 \$MC_TRAFO5_JOINT_OFFSET_1[2]	\$TC_CARR17 (+\$TC_TCARR57)
MD24500 \$MC_TRAFO5_PART_OFFSET_1[0]	\$TC_CARR18 (+\$TC_TCARR58)
MD24500 \$MC_TRAFO5_PART_OFFSET_1[1]	\$TC_CARR19 (+\$TC_TCARR59)
MD24500 \$MC_TRAFO5_PART_OFFSET_1[2]	\$TC_CARR20 (+\$TC_TCARR60)

Zuordnungen für Trafotyp 56

Vom Transformationstyp 56 abhängige Zuordnungen der Werkzeugträgerdaten

Transformationstyp "M" (entspricht MD24100 \$MC_TRAFO_TYPE_1 = 56)	
MD24560 \$MC_TRAFO5_JOINT_OFFSET_1[0] (Vektor des kinematischen Versatzes der 5-Achstransformation 1)	\$TC_CARR1 (+\$TC_TCARR41)
MD24560 \$MC_TRAFO5_JOINT_OFFSET_1[1]	\$TC_CARR2 (+\$TC_TCARR42)
MD24560: TRAF05_JOINT_OFFSET_1[2]	\$TC_CARR3 (+\$TC_TCARR43)
MD24550 \$MC_TRAFO5_BASE_TOOL_1[0]	\$TC_CARR4 (+\$TC_TCARR44)
MD24550 \$MC_TRAFO5_BASE_TOOL_1[1]	\$TC_CARR5 (+\$TC_TCARR45)
MD24550 \$MC_TRAFO5_BASE_TOOL_1[2]	\$TC_CARR6 (+\$TC_TCARR46)
MD24558 \$MC_TRAFO5_JOINT_OFFSET_PART_1[0]	\$TC_CARR15 (+\$TC_TCARR55)
MD24558 \$MC_TRAFO5_JOINT_OFFSET_PART_1[1]	\$TC_CARR16 (+\$TC_TCARR56)
MD24558 \$MC_TRAFO5_JOINT_OFFSET_PART_1[2]	\$TC_CARR17 (+\$TC_TCARR57)
MD24500 \$MC_TRAFO5_PART_OFFSET_1[0]	\$TC_CARR18 (+\$TC_TCARR58)
MD24500 \$MC_TRAFO5_PART_OFFSET_1[1]	\$TC_CARR19 (+\$TC_TCARR59)
MD24500 \$MC_TRAFO5_PART_OFFSET_1[2]	\$TC_CARR20 (+\$TC_TCARR60)

Beispiel für eine Parametrierung

Die erste 5-Achs-Transformation soll ihre Daten aus den Maschinendaten erhalten und die zweite 5-Achs-Transformation soll dagegen mit Hilfe der Daten des 3. orientierbaren Werkzeugträgers parametrieren werden.

MD24100 \$MC_TRAFO_TYPE_1 = 24	; erste 5-Achs-Transformation
MD24200 \$MC_TRAFO_TYPE_2 = 72	; zweite 5-Achs-Transformation
MD24682 \$MC_TRAFO5_TCARR_NO_2 = 3;	; Daten des 3. orientierbaren Werkzeugträgers parametrieren

2.7.5 Erweiterung der Generischen Transformation auf 6 Achsen - nur 840D sl

Die 6-Achstransformation ist eine Erweiterung der generischen 5-Achs-Transformation um eine zusätzliche Rundachse. Diese ermöglicht die Beschreibung einer Drehung des Werkzeugs um sich selbst.

Neben den Maschinendaten der 5-Achs-Transformation müssen zur Parametrierung einer 6-Achs-Transformation weitere Einstellungen in den nachfolgend aufgeführten Maschinendaten vorgenommen werden.

Parametrierung: Transformationstyp

Der Transformationstyp wird kanalspezifisch eingestellt über:

\$MC_TRAFO_TYPE_<x> = <Transformationstyp>

Der einzustellende Transformationstyp ergibt sich aus dem Maschinentyp bzw. der Aufteilung der drei Rundachsen bezüglich der Werkzeug- und Werkstückorientierung:

<Transformationstyp>	24 ¹⁾	40 ¹⁾	56 ²⁾	57 ³⁾
Maschinentyp	1	2	3	4
Werkzeugorientierung	3 Rundachsen	---	2 Rundachsen	1 Rundachse
Werkstückorientierung	---	3 Rundachsen	1 Rundachse	2 Rundachsen

- 1) Die **erste** Rundachse der Transformation ist die Rundachse, die in der kinematischen Kette am nächsten beim **Werkstück** liegt.
Die **dritte** Rundachse der Transformation ist die Rundachse, die in der kinematischen Kette am nächsten beim **Werkzeug** liegt.
Bei einer Verfahrbewegung der **zweiten** Rundachse wird die dritte Rundachse mit bewegt.
- 2) Die **zweite** Rundachse der Transformation ist die Rundachse, die in der kinematischen Kette am nächsten beim **Werkstück** liegt.
Die **dritte** Rundachse der Transformation ist die Rundachse, die in der kinematischen Kette am nächsten beim **Werkzeug** liegt.
Bei einer Verfahrbewegung der **ersten** Rundachse wird die dritte Rundachse mit bewegt.
- 3) Die **erste** Rundachse der Transformation ist die Rundachse, die in der kinematischen Kette am nächsten beim **Werkzeug** liegt.
Die **dritte** Rundachse der Transformation ist die Rundachse, die in der kinematischen Kette am nächsten beim **Werkstück** liegt.
Bei einer Verfahrbewegung der **zweiten** Rundachse wird die dritte Rundachse mit bewegt.

Hinweis

Durch die Transformationstypen 24, 40, 56 und 57 werden nur solche Kinematiken abgebildet, bei denen die drei Linearachsen der Transformation ein rechtshändiges kartesisches Koordinatensystem bilden und in der kinematischen Kette keine Rundachse zwischen den Linearachsen liegt.

Parametrierung: Offsetvektoren für Transformationstyp 57

Die Offsetvektoren sind beim Transformationstyp 57 wie die Rundachsen ausgehend vom Werkzeug zum Werkstücknullpunkt aufzuführen:

Offset	Maschinendatum
Werkzeugspitze → Rundachse 1	\$MC_TRAFO5_BASE_TOOL_<x>[0 ... 2]
Rundachse 1 → Rundachse 2	\$MC_TRAFO5_JOINT_OFFSET_<x>[0 ... 2]
Rundachse 2 → Rundachse 3	\$MC_TRAFO6_JOINT_OFFSET_2_3_<x>[0 ... 2]
Rundachse 3 → Werkstücknullpunkt	\$MC_TRAFO5_PART_OFFSET_<x>[0 ... 2]

Parametrierung: Kanalachsnummer der 3. Rundachse

Die Kanalachsnummer der dritten Rundachse wird eingestellt in:

$$\$MC_TRAFO_AXES_IN_<x>[5] = <Kanalachsnummer der dritten Rundachse>$$

Parametrierung: Richtungsvektor der 3. Rundachse

Der Richtungsvektor der dritten Rundachse wird eingestellt in:

$$\$MC_TRAFO5_AXIS3_<x>[0 \dots 2] = <Komponente des Richtungsvektors>$$

Parametrierung: Werkzeugnormalenvektor

Der Werkzeugnormalenvektor darf nicht parallel oder antiparallel zum Vektor der Werkzeuggrundorientierung (MD24574 \$MC_TRAFO5_BASE_ORIENT_<x>) sein. Er wird eingestellt in:

$$\$MC_TRAFO6_BASE_ORIENT_NORMAL_<x>[0 \dots 2] = <Komponente des Werkzeugnormalenvektors>$$

Parametrierung: Offsetvektor zwischen zweiter und dritter Rundachse

Der Offsetvektor zwischen der zweiten und dritten Rundachse wird eingestellt in:

$$\$MC_TRAFO6_JOINT_OFFSET_2_3_<x>[0 \dots 2] = <Komponente des Offsetvektors>$$

Orientierung

Mit der Erweiterung der generischen Orientierungstransformation auf sechs Achsen können alle drei Freiheitsgrade der Orientierung frei gewählt werden. Sie lassen sich eindeutig durch die Lage eines rechtwinkligen, kartesischen Koordinatensystems bestimmen. Dabei bestimmt die Achsrichtung der **dritten Achse** die **Orientierung**.

Für die Festlegung dieser Achsrichtung werden zwei Freiheitsgrade benötigt. Der dritte Freiheitsgrad wird bestimmt durch eine Drehung um die Achsrichtung, z. B. durch die Angabe eines Winkels THETA oder eines Richtungsvektors für eine der beiden anderen Achsen des Koordinatensystems (siehe Kapitel "Drehungen des Orientierungsvektors (Seite 125)").

Mit den Adressen AN3, BN3, CN3 wird die Richtung der **zweiten Achse** des Koordinatensystems bestimmt, d. h. der **Orientierungsnormalenvektor**. Der mit den Adressen

AN3, BN3, CN3 programmierte Orientierungsnormalenvektor soll senkrecht auf der Orientierung stehen. Ist das nicht der Fall, wird der programmierte Orientierungsnormalenvektor intern so modifiziert. Der modifizierte Orientierungsnormalenvektor liegt dann in der von programmiertem Orientierungs- und Orientierungsnormalenvektor aufgespannten Ebene und steht senkrecht auf dem Orientierungsvektor. Diese Orthogonalisierung ist nur möglich, wenn die beiden programmierten Vektoren nicht parallel bzw. antiparallel sind. Ist diese Bedingung nicht erfüllt, wird Alarm 4342 angezeigt.

Richtungsvektoren in den Werkzeugdaten

Orientierungsvektors

Die Vektorkomponenten (x,y,z) des Orientierungsvektors eines Werkzeugs können in den Werkzeugdaten über \$TC_DP3 bzw. \$TC_DP3 - \$TC_DP5 definiert werden.

Literatur: Funktionshandbuch Grundmaschine; Werkzeugkorrekturen (W1), Kapitel: Summen- und Einrichtkorrekturen.

Orientierungsnormalenvektors

Die Vektorkomponenten (x,y,z) des Orientierungsnormalenvektors eines Werkzeugs können in den Werkzeugdaten über \$TC_DP3 - \$TC_DP5 definiert werden. Die Bedeutung der Vektorkomponenten ist dabei analog zu den Vektorkomponenten des Orientierungsvektors:

- \$TC_DP3: Komponente in Richtung der Werkzeuglänge L1,
- \$TC_DP4: Komponente in Richtung der Werkzeuglänge L2
- \$TC_DP5: Komponente in Richtung der Werkzeuglänge L3.

Damit die Werkzeugdaten für den Orientierungsnormalenvektor verwendet werden können, muss folgende Einstellung vorgenommen werden:

MD18114 \$MN_MM_ENABLE_TOOL_ORIENT = 3 (Werkzeugschneiden Orientierung zuordnen)

Durch eine Drehung des Werkzeugs mit AN3, BN3, CN3 bzw. THETA wird das Koordinatensystem nicht gedreht.

Grundstellung des Orientierungsnormalenvektors

Die Grundstellung des Orientierungsnormalenvektors wird bei der **Aktivierung der Transformation** auf eine der folgenden Arten bestimmt:

1. Die Vektorkomponenten (x,y,z) werden beim Funktionsaufruf TRAORI als Parameter 8 - 10 übergeben:
 TRAORI(p1, p2, p3, p4, p5, p6, p7, **p8, p9, p10**)
2. Wird beim Funktionsaufruf TRAORI **kein** Orientierungsnormalenvektor angegeben und es ist **ein** Werkzeug aktiv, werden die Vektorkomponenten (x,y,z) aus den Werkzeugdaten genommen:
 \$TC_DPVN3, \$TC_DPVN4, \$TC_DPVN5
3. Wird beim Funktionsaufruf TRAORI **kein** Orientierungsnormalenvektor angegeben und es ist **kein** Werkzeug aktiv, werden die Vektorkomponenten (x,y,z) aus folgendem Maschinendatum genommen:
 MD24567 \$MC_TRAFO6_BASE_ORIENT_NORMAL_<x>[0 ... 2]
 (Werkzeugnormalenvektor)

Die Lage des Orientierungskordinatensystems eines Standardwerkzeugs ist abhängig von der **aktiven Ebene** (G17, G18, G19) entsprechend der nachfolgenden Tabelle:

	G17	G18	G19
Richtung des Orientierungsvektors	Z	Y	X
Richtung des Orientierungsnormalenvektors	Y	X	Z

Siehe auch

Beispiel für eine Generische 6-Achs-Transformation (Seite 147)

2.7.6 Erweiterung der Generischen Transformation auf 7 Achsen - nur 840D sl

Anwendung

Die Generische 5-/6-Achs Transformation mit Transformationstyp 24 wird um eine 7. bzw. 6. Achse erweitert, die das Werkstück dreht. Dadurch kann der Arbeitsbereich der Transformation erweitert werden.

Voraussetzung

Für die Generische 7-Achs-Transformation müssen mindestens 7 bzw. 6 Achsen vorhanden sein.

Funktion

In Verbindung mit der Generischen 6-Achs-Transformation wird eine weitere 7. Achse benötigt, die das Werkstück dreht. Diese 7. Achse wird nur zusammen mit Trafo-Typ 24 (Generische 6-Achs-Transformation mit 3 Rundachsen, die das Werkzeug bewegen) berücksichtigt.

Die Position der 7. Achse wird gemäß einer Strategie des CAD-Systems vorgegeben und von der Generischen Transformation so mit der kartesischen Position (X, Y, Z) verrechnet, dass unabhängig von der Position der 7. Achse immer die bezüglich des Werkstücks programmierte TCP-Position angefahren wird. Wenn ORIWKS aktiv ist, wird die bezüglich des Werkstücks programmierte Endorientierung durch die 7. Achse ebenfalls gedreht. Hiermit ist es möglich, auch die Orientierung werkstückbezogen zu programmieren.

Die Transformation verwendet die 7. Achse als beobachtete Eingangsgröße.

Zur Projektierung der 7. Achse sind die Kanal-Maschinendaten für die 5-/6-Achs-Transformation um ein Feld mit den 3 Komponenten des Richtungsvektors der 7. Achse und einen Achs-Offset erweitert.

Daraus resultieren folgende Vorteile:

- Die Programmierung der Kontur und der Orientierung am Werkstück kann werkstückbezogen erfolgen.
- Der programmierte Vorschub auf der Kontur wird eingehalten, auch wenn sich die 7. Achse mitbewegt.
- Alle konturbezogenen Steuerungsfunktionen können verwendet werden.
- Die angezeigte WKS-Position entspricht der programmierten Position.
- Die Projektierung der Transformation erfolgt wie bei der Generischen 6-Achs-Transformation. Es kann problemlos zwischen einer 6- und einer 7-Achs-Transformation umgeschaltet werden.
- Bei der Großkreisinterpolation erfolgt die Auflösung von Singularitäten unter Einbeziehung der 7. Achse.

Schreibweisen

Für jede allgemeine Transformation und für jede Orientierungstransformation existieren eigene Maschinendaten, die sich durch die Endungen _1, _2 usw. voneinander unterscheiden (z. B. \$MC_TRAFO_TYPE_1, \$MC_TRAFO_TYPE_2 usw.). Im Folgenden werden jeweils nur die Namen für die erste Transformation, d.h. die mit den Endungen _1 angegeben. Soll eine andere als die erste Transformation parametrisiert werden, müssen die entsprechend modifizierten Namen verwendet werden.

Beschreibung der Kinematik

Die 7-Achs-Transformation baut auf der Generischen 5-/6-Achs-Transformation auf.

Hinweis

Mit der 7-Achs Transformation sind auch Kinematiken abgedeckt, bei denen die 6. Achse nicht vorhanden ist. Es wird im Folgenden immer nur von einer 7. Achse oder von einer 7-Achs-Transformation gesprochen, auch wenn es sich in Verbindung mit einer 5-Achs-Kinematik eigentlich um die 6. Achse handelt.

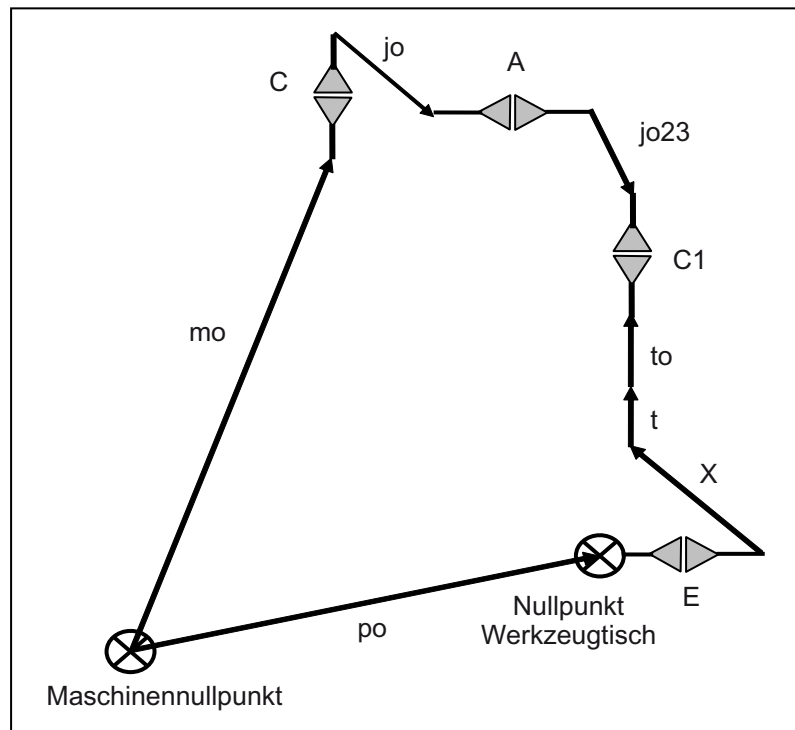
Mit der 7-Achs-Transformation sind nur solche Kinematiken abgedeckt, bei denen die drei Linearachsen ein rechtwinkliges Koordinatensystem bilden, d.h. es werden keine Kinematiken erfasst, bei denen in der kinematischen Kette zwischen zwei Linearachsen mindestens eine Rundachse liegt.

Es gibt nur eine Maschinen-Kinematik, für die eine 7. Achse projiziert werden kann. Diese wird durch den Transformationstyp 24 gekennzeichnet:

\$MC_TRAFO_TYPE_1 = 24 Drehbares Werkzeug: Drei (oder zwei) Achsen drehen das Werkzeug; die 7. Achse dreht das Werkstück.

Zur Konfiguration einer Generischen 7-Achs-Transformation sind Erweiterungen für folgende Maschinendaten notwendig:

Maschinendatum	Erweiterung
\$MC_TRAFO_AXES_IN_1[9]	Hier wird der Kanalachsindex der 4. Rundachse eingetragen.
\$MC_TRAFO_AXES_IN_1[10] und \$MC_TRAFO_AXES_IN_1[11] \$MC_TRAFO_AXES_IN_1[6..8]	Diese Maschinendaten sind mit 0 vorzubesetzen. (Voreinstellung) Diese Maschinendaten werden von der Generischen 7-Achs-Transformation nicht ausgewertet.
\$MC_TRAFO7_EXT_AXIS1_1[0..2]	Hier wird die Richtung der 4. Rundachse spezifiziert.
\$MC_TRAFO7_EXT_AX_OFFSET_1[0..2]	Hier wird ein Positions-Offset der 4. Rundachse eingetragen.



- mo: Positionsvektor im MKS
 po: \$MC_TRAFO5_PART_OFFSET_n[0..2]
 x. Vektor der programmierten Position im WKS
 t: Werkzeugkorrekturvektor
 to: \$MC_TRAFO5_BASE_TOOL_n[0..2]
 jo: \$MC_TRAFO5_JOINT_OFFSET_n[0..2]
 jo23: \$MC_TRAFO6_JOINT_OFFSET_2_3_n[0..2]
 Bild 2-21 Schematische Darstellung einer 7-Achs-Kinematik

Programmierung

1. Programmierung der kartesischen Position

Die Stellung der 7. Achse muss immer zusätzlich zur kartesischen Position im Werkstück-Koordinatensystem programmiert werden. Die kartesische Position wird hierbei bezogen auf das feststehende Werkstück programmiert. Die 7-Achs-Transformation rechnet die WKS-Position über die Drehung der 7. Achse ins Basis-Koordinatensystem um. Eventuell programmierte oder eingestellte Frames werden standardmäßig schon vor der 7-Achs-Transformation verrechnet.

2. Programmierung der Orientierung

Bei der Programmierung der Orientierung stehen alle Programmiermöglichkeit der Generischen 5-/6-Achs-Transformation zur Verfügung. Die 7. Achse muss immer zusätzlich programmiert werden.

Hierbei sind über den G-Befehl zwei unterschiedliche Verhaltensweisen einstellbar:

- Die Stellung der 7. Achse beeinflusst die programmierte Orientierung nicht.
- Die programmierte End-Orientierung wird mit der 7. Achse verdreht.

Orientierung

1. Orientierung mit Achsinterpolation

Wenn die 7. Achse keinen Einfluss auf die programmierte Orientierung haben soll, müssen die G-Befehle der Gruppen 25 und 51 entsprechend eingestellt werden:

G-Gruppe 25: ORIMKS

G-Gruppe 51: ORIAxes (wenn MD21104 \$MC_ORI_IPO_WITH_G_CODE = 1 gesetzt ist).

Die programmierten Positionen der Rundachsen werden in diesem Fall nicht von der Stellung der 7. Achse verändert, sondern direkt angefahren. Die Programmierung der Orientierung erfolgt bezogen auf die Maschine.

Beispiel

Programmcode

```
TRAORI(1)
ORIAxes
ORIMKS
G1 X500 Y300 Z800 C15 A5 C1=10 E1=120
```

1. Orientierung und Großkreis-Interpolation

Wenn mit Großkreis-Interpolation verfahren werden soll, wird die End-Orientierung mit der 7. Achse verdreht.

G-Befehl Gruppe 25: ORIWKS

G-Befehl Gruppe 51: ORIVect (wenn MD21104 \$MC_ORI_IPO_WITH_G_CODE = 1 gesetzt ist).

In diesem Fall muss die Programmierung der Orientierung werkstückbezogen erfolgen. Die programmierte Orientierung bezieht sich hierbei auf das feststehende Werkstück. Die Stellung der 7. Achse ist hierbei nicht in der programmierten Orientierung enthalten.

Beispiel

Programmcode

```
TRAORI(1)
ORIVect
ORIWKS
G1 X500 Y800 Z100 A3=0 B3=1 C3=0 AN3=0 BN3=0 CN3=-1 E1=-90
```

Frames

Das Basis-Koordinatensystem sitzt auf der 7. Achse. Es wird bei einer Drehung der 7. Achse mitgedreht. Dadurch bleibt auch das Werkstück-Koordinatensystem (WKS) nicht raumfest, wenn das Werkstück über die 7. Achse gedreht wird. Eine zur Nullstellung der 7. Achse verdrehte Werkstücklage kann durch eine axiale Frame-Verschiebung der 7. Achse kompensiert werden.

Verfahren mit der 7. Achse in der Betriebsart JOG

Wenn die 7. Achse in der Betriebsart JOG mit aktiver 7-Achs-Transformation verfahren wird, werden nur Ausgleichsbewegungen für die Linearachsen erzeugt. Dadurch wird die Position am Werkstück konstant gehalten. Da die Rundachsen nur als Eingangsachsen in die Transformation eingehen, werden sie beim JOG-Verfahren der 7. Achse nicht beeinflusst. Die Orientierung am Werkstück wird damit nicht konstant gehalten.

2.7.7 Kartesisches Handverfahren bei Generischer Transformation

Hinweis

Option

Für die Funktion ist die Option "Transformationspaket Handling" erforderlich.

Funktionalität

Mit der Funktion "Kartesisches manuelles Verfahren" können für das Verfahren der Geometrie- und Orientierungsachse in der Betriebsart JOG folgende kartesische Koordinatensystemen eingestellt werden:

- Basiskoordinatensystem (BKS)
- Werkstückkoordinatensystem (WKS)
- Werkzeugkoordinatensystem (TCS)

Die Verfahrensbewegungen der Linear- und Rundachsen erfolgen über die NC/PLC-Nahtstellensignale der Geometrie- bzw. Orientierungsachsen.

Hinweis

Weitere Informationen zur Darstellung der Translationen beim Kartesischen manuellen Verfahren in den entsprechenden Koordinatensystemen siehe:

Literatur:

Funktionshandbuch Erweiterungsfunktionen; Kinematische Transformation (M1)

Parametrierung: Maschinendaten

Koordinatensysteme

Mit dem folgenden Maschinendatum wird sowohl die Funktion aktiviert, als auch die erlaubten Koordinatensysteme zwischen denen umgeschaltet werden kann, eingestellt:

MD21106 \$MC_CART_JOG_SYSTEM, Bit n = <Wert>

Bit n	Wert	Bedeutung
0	1	Funktion aktiv: Basiskoordinatensystem (BKS)
1	1	Funktion aktiv: Werkstückkoordinatensystem (WKS)
2	1	Funktion aktiv: Werkzeugkoordinatensystem (TCS)

Parametrierung: Settingdaten

Definition der virtuellen Kinematik für die Orientierungsachsen

Die wirksame virtuelle Kinematik für das manuelle Verfahren der Orientierungsachsen wird über folgendes das Settingdatum festgelegt. Es können hier nur Kinematiken eingestellt werden, bei denen die Rundachsen senkrecht aufeinander stehen..

SD42660 \$SC_ORI_JOG_MODE = <Wert>

Wert	Bedeutung
0	Die virtuelle Kinematik wird durch die Transformation festgelegt.
1	Drehung mit Euler-Winkeln: Drehreihenfolge ZX'Z"-Konvention
2	Drehung mit Euler-Winkeln: XY'Z"-Konvention (RPY-Winkel)
3	Drehung mit Euler-Winkeln: Drehreihenfolge ZY'X"-Konvention (RPY-Winkel)
4	Festlegung der Drehreihenfolge mit: MD21120 \$MC_ORIAX_TURN_TAB_1
5	Festlegung der Drehreihenfolge mit: MD21130 \$MC_ORIAX_TURN_TAB_2

Weitere Erläuterungen zu Orientierungsbewegungen (siehe die Kapitel "Orientierung (Seite 91)" und "Orientierungsachsen (Seite 113)").

Hinweis

Weitere Informationen zur Programmierung der Drehungen siehe:

Literatur:

Programmierhandbuch Arbeitsvorbereitung; Kapitel: "Transformationen"

2.8 Einschränkungen für Kinematiken und Interpolationen

Bei Systemen bei denen für die Transformation weniger als 6 Achsen zur Verfügung stehen, sind folgende Einschränkungen zu beachten.

5-Achs-Kinematik

Bei einer 5-Achs-Kinematik sind zwei Freiheitsgrade für die Orientierung vorhanden. Die Zuordnung der Orientierungsachsen und die Richtung des Werkzeugvektors sind so zu wählen, dass die Drehung um den Werkzeugvektor entfällt. Deshalb sind nur zwei Orientierungswinkel zur Beschreibung der Orientierung nötig. Wird mit `ORIVECT` verfahren, wird nur eine reine Schwenkbewegung des Werkzeugvektors ausgeführt.

3- und 4-Achs-Kinematik

Bei 3- und 4-Achs-Kinematiken ist nur ein Freiheitsgrad für die Orientierung vorhanden. Um welchen Orientierungswinkel gedreht werden soll, ist von der jeweiligen Transformation abhängig. Dabei ist es nur sinnvoll, die Orientierungsachse mit `ORIXES` zu verfahren. Die Orientierungsachse wird dabei direkt linear interpoliert.

Interpolation der Werkzeugorientierung über mehrere Sätze mittels Orientierungsvektoren

Wird die Orientierungen eines Werkzeugs über mehrere aufeinanderfolgenden Teileprogrammätze durch direkte Angabe der entsprechenden Rundachspalten programmiert, ergeben sich an den Satzübergängen unerwünschte nichtstetige Änderungen des Orientierungsvektors. Daraus ergeben sich nichtstetige Geschwindigkeits- und Beschleunigungsänderungen der Rundachsen. Daraus folgt, dass mittels Großkreisinterpolation keine geschwindigkeits- und beschleunigungsstetige Bewegung der Orientierungsachsen über mehrere Sätze hinweg erreicht werden kann.

Stetige Satzübergänge

Solange nur Linearätze ($G1$) programmiert werden, verhalten sich auch die Orientierungsachsen wie Linearachsen. Dabei wird dann eine beschleunigungsstetige Bewegung durch Polynominterpolation erreicht. Wesentlich bessere Ergebnisse lassen sich aber durch die Programmierung der Orientierung im Raum mittels Orientierungsvektoren erzielen (siehe Kapitel "Polynominterpolation von Orientierungsvektoren (Seite 121)").

2.8.1 Singularitäten der Orientierung

Problemstellung

Wie im Kapitel "Singuläre Stellen und ihre Behandlung" bereits beschrieben, sind Singularitäten (Pole) solche Konstellationen, bei denen die Werkzeugorientierung parallel zur ersten Rundachse wird. Wird bei oder nahe bei einer solchen Orientierung die Orientierungsänderung vorgegeben (wie das bei der Großkreisinterpolation ORIWKS der Fall ist), so werden für kleine Orientierungsänderungen Große Änderungen der Rundachspalten nötig. Im Extremfall müsste die Rundachspalten springen.

Eine derartige Situation wird wie folgt behandelt:

Es gibt nur noch ein relevantes Maschinendatum, das wie bisher einen Kreis um den Pol beschreibt:

MD24540 \$MC_TRAFO5_POLE_LIMIT_1 (Endwinkeltoleranz b. Interpolation durch Pol f. 5-Achstransformation)

bzw.

MD24640 \$MC_TRAFO5_POLE_LIMIT_2 (Endwinkeltol. b. Interpolation durch Pol f. 5-Achstransformation)

Weitere Informationen zur Behandlung singulärer Stellen siehe:

Literatur:

Programmierhandbuch Arbeitsvorbereitung; Transformationen, Kapitel: Kartesisches PTP-Fahren

Beispiel für Maschinentyp 1

Drehbares Werkzeug

Beide Rundachsen verändern die Orientierung des Werkzeugs. Die Orientierung des Werkstücks ist fest.

2-Achsen-Schwenkkopf mit der Rundachse RA 1 (4. Achse der Transformation) und der Rundachse RA 2 (5. Achse der Transformation).

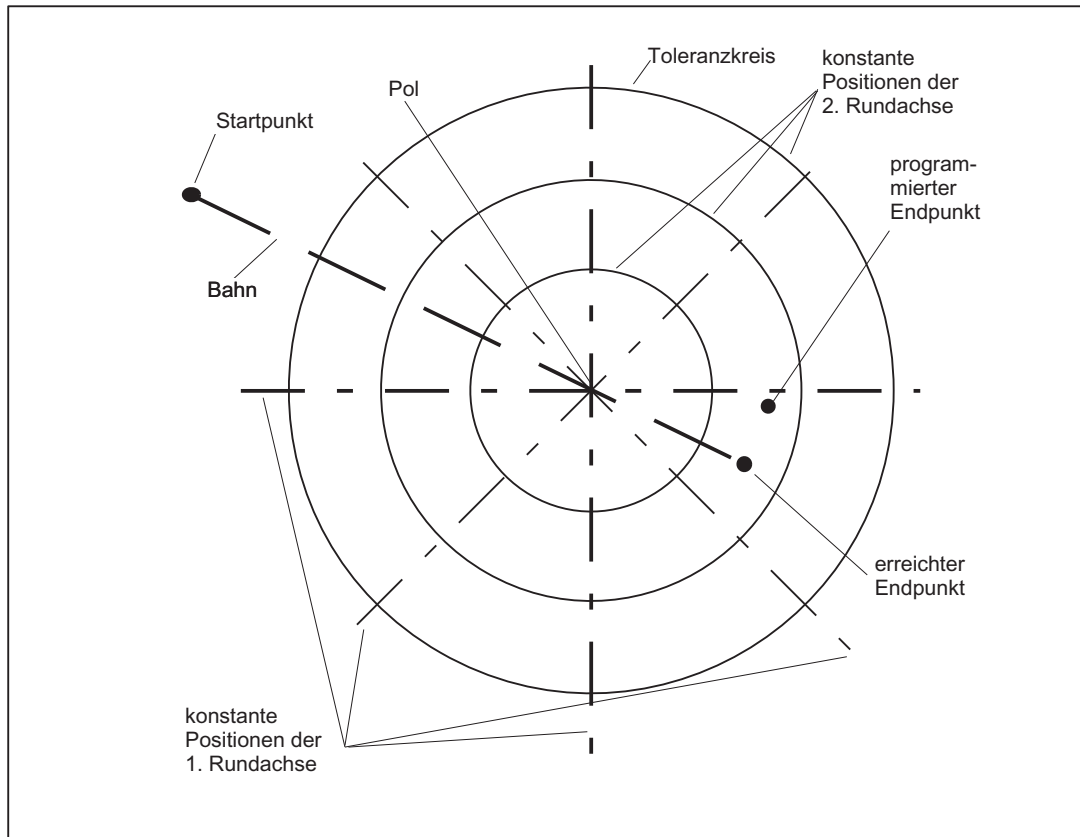


Bild 2-22 Generische 5-Achs-Transformation; Endpunkt der Orientierung innerhalb des Toleranzkreises.

Endpunkt innerhalb des Kreises

Liegt der Endpunkt innerhalb des Kreises, so bleibt die erste Achse stehen, und es fährt nur die zweite Achse, und zwar so weit, dass die Differenz zwischen Soll- und Istorientierung minimal wird. Da die erste Rundachse nicht bewegt wird, wird die Orientierung aber im Allgemeinen vom programmierten Wert abweichen (siehe vorheriges Bild). Es wird aber damit erreicht, dass die programmierte Orientierung zumindest dann exakt erreicht wird, wenn die erste Rundachse zufälligerweise richtig steht.

Hinweis

Im vorherigen Bild ist die resultierende Bahn eine Gerade, da auf ihr die Position der ersten Rundachse konstant ist. Diese Darstellung ist immer richtig, unabhängig davon, in welchem Winkel die beiden Rundachsen zueinander stehen. Der Orientierungsvektor bewegt sich jedoch nur dann in einer Ebene, wenn die beiden Rundachsen und die Grundorientierung jeweils aufeinander senkrecht stehen. In allen anderen Fällen beschreibt der Orientierungsvektor einen Kegelmantel.

Endpunkt außerhalb des Kreises

Würde die Orientierungsinterpolation durch den Kreis hindurchführen, während der Endpunkt außerhalb des Kreises liegt, wird der Endpunkt mit Achsinterpolation angefahren. Dies gilt insbesondere auch dann, wenn der Startpunkt der Interpolation innerhalb des Kreises liegt. Dadurch ergeben sich auf der Bahn zwangsläufig Abweichungen von der programmierten Sollorientierung.

2.9 Orientierung

2.9.1 Grundorientierung

Unterschiede zu bisherigen 5-Achs-Transformationen

Bei den bisher realisierten 5-Achstransformationen war die Grundorientierung des Werkzeugs durch den Transformationstyp festgelegt.

Mit der generischen 5-Achstransformation ist es möglich, beliebige Grundorientierungen des Werkzeugs zuzulassen, d. h. die Orientierung des Werkzeugs im Raum bei Grundstellung der Achsen ist beliebig.

Wird eine Orientierung mit Hilfe von Euler-Winkeln, RPY-Winkeln (A2, B2, C2) oder mittels Vektoren (A3, B3, C3) programmiert, wird die Grundorientierung berücksichtigt, d. h. die Rundachsen stellen sich so ein, dass ein in Grundorientierung ausgerichtetes Werkzeug in die programmierte Orientierung überführt wird.

Werden die Rundachsen direkt programmiert, so hat die Grundorientierung darauf keinen Einfluss.

Definition

Die Definition der Grundorientierung kann auf drei verschiedenen Arten erfolgen:

- Definition beim Aufruf der Transformation
- Definition über die Orientierung des aktiven Werkzeugs
- Definition über ein Maschinendatum

Definition beim Aufruf der Transformation

Beim Aufruf der Transformation kann der Richtungsvektor der Grundorientierung im Aufruf angegeben werden, z. B. `TRAORI(0, 0., 1., 5.)`. Der Richtungsvektor wird dabei durch die Parameter 2 bis 4 definiert. Im Beispiel hat er deshalb den Wert (0., 1., 5.).

Der erste Parameter gibt die Transformationsnummer an. Soll die erste Transformation aktiviert werden, kann die Nummer auch weggelassen werden. Bei Angabe einer Orientierung

müssen aber die Parameter hierbei eindeutig bleiben, so dass für die Transformationsnummer ein leerer Platz vorgesehen wird, z. B. TRAORI (, 0., 1., 5.).

Hinweis

Die Orientierungsangabe gilt absolut. Sie wird durch einen eventuell aktiven Frame nicht modifiziert.

Der Betrag des Vektors ist unerheblich, nur die Richtung ist von Bedeutung. Nicht programmierte Vektorkomponenten werden auf Null gesetzt.

Sind insbesondere alle drei Vektorkomponenten Null (entweder weil sie explizit so programmiert oder weil sie nicht angegeben wurden), wird die Grundorientierung nicht durch Angaben im Aufruf von TRAORI (. . .), sondern durch eine der beiden anderen Möglichkeiten definiert.

Wird die Grundorientierung beim Aufruf der Transformation definiert, kann sie während einer aktiven Transformation nicht mehr verändert werden. Eine Änderung ist nur durch eine Neuanwahl der Transformation möglich.

Definition über die Orientierung des aktiven Werkzeugs

Die Grundorientierung wird durch das Werkzeug festgelegt, wenn:

- sie nicht durch Angabe eines Richtungsvektors bei der Transformationsanwahl definiert wurde und
- wenn ein Werkzeug aktiv ist.

Die Orientierung eines Werkzeugs hängt von der angewählten Ebene ab. Sie ist bei G17 parallel zu Z, bei G18 parallel zu Y und bei G19 parallel zu X.

Sie kann durch orientierbare Werkzeugträger beliebig modifiziert werden, siehe:

Literatur:

Funktionshandbuch Grundfunktionen; Werkzeugkorrektur (W1), Kapitel: "Orientierbare Werkzeugträger"

Wird bei aktiver Transformation das Werkzeug gewechselt, so wird auch die Grundorientierung aktualisiert. Gleiches gilt, wenn sich die Orientierung eines Werkzeugs aufgrund eines Ebenenwechsels ändert (ein Ebenenwechsel ist gleichbedeutend mit einem Werkzeugwechsel, da sich dadurch auch die Zuordnung der Werkzeuglängenenkomponenten zu den einzelnen Achsen ändert).

Wird das Werkzeug abgewählt, so dass auch keine Werkzeugorientierung mehr definiert ist, wird die Grundorientierung aus den Maschinendaten wirksam.

Definition über ein Maschinendatum

Wird die Grundorientierung durch keine der beiden zuvor beschriebenen Varianten definiert, wird sie aus dem folgenden Maschinendatum entnommen:

\$MC_TRAFO5_BASE_ORIENT_n (Werkzeuggrundorientierung)

Dieses Maschinendatum muss vom Nullvektor verschieden sein, da sonst beim Steuerungshochlauf bei aktiver Transformation ein Alarm erzeugt wird.

Ist bei einer aktiven Transformation die Grundorientierung durch das Maschinendatum \$MC_TRAFO5_BASE_ORIENT_n festgelegt und wird ein Werkzeug aktiviert, so wird die Grundorientierung durch das Werkzeug neu definiert.

Hinweis

Der Bereich der einstellbaren Orientierungen hängt von den Richtungen der beteiligten Rundachsen und der Grundorientierung ab. Nur wenn diese jeweils aufeinander senkrecht stehen, sind alle denkbaren Orientierungen erreichbar. Bei Abweichungen von dieser Bedingung gibt es "tote" Bereiche.

Beispiele:

1. Extrembeispiel: Eine Maschine mit drehbarem Werkzeug hat als erste Rundachse eine C-Achse und als zweite Rundachse eine A-Achse. Wird die Grundorientierung parallel zur A-Achse definiert, kann die Orientierung nur noch in der X-Y-Ebene verändert werden (bei Drehung der C-Achse), d. h. eine Orientierung mit einem Z-Anteil ungleich Null ist dann nicht möglich. Eine Drehung der A-Achse verändert die Orientierung nicht.
2. Realistisches Beispiel: Eine Maschine mit Nutatorkinematik (kardanischer Kopf) mit einer schrägen Achse unter 45° unter einer Grundorientierung parallel zur Z-Achse kann nur Orientierungen innerhalb einer Halbkugel abdecken: die obere Halbkugel bei Grundorientierung in Richtung +Z bzw. die untere bei Grundorientierung in Richtung -Z.

2.9.2 Orientierungsbewegungen mit Achsgrenzen

Rundachspannung berechnen

Wird bei einer 5-Achs-Transformation die Endorientierung in einem Satz indirekt, durch Euler-, RPY-Winkel oder Richtungsvektor programmiert, so müssen die Rundachspannungen berechnet werden, bei denen sich die gewünschte Orientierung ergibt. Diese Berechnung ist nicht eindeutig.

Es gibt immer mindestens zwei wesentlich verschiedene Lösungen. Dazu kommen beliebig viele weitere Lösungen, die sich daraus ergeben, dass die Rundachspannungen um beliebige Vielfache von 360 Grad modifiziert werden.

Die Steuerung wählt unter den möglichen Lösungen diejenige aus, die unter Berücksichtigung des programmierten Interpolationstyps der Orientierung vom aktuellen Startpunkt aus auf dem kürzesten Weg zu erreichen ist.

Zulässige Achsgrenzen ermitteln

Immer wenn das Anfahren der gewünschten Achsstellung auf dem kürzesten Wege zu einer Verletzung der Achsgrenzen führen würde, versucht die Steuerung eine andere zulässige Lösung zu bestimmen. Zunächst wird die zweite Lösung untersucht, und falls auch diese Lösung die Achsgrenzen verletzt, werden für beide Lösungen die Achspannungen um Vielfache von 360 modifiziert, mit dem Ziel eine gültige Position zu finden.

Damit die Achsbegrenzungen einer Rundachse überwacht und die zu ermittelnde Endpositionen gegebenenfalls modifiziert werden, müssen die folgenden Voraussetzungen erfüllt sein:

- Eine generische 5-Achs-Transformation vom Trafotyp 24, 40 oder 56 muss aktiv sein.
- Die Achse muss referenziert sein.
- Die Achse darf keine Modulo-Rundachse sein
- Das folgende Maschindatum muss ungleich Null sein:
MD21180 \$MC_ROT_AX_SWL_CHECK_MODE (Check Softwarelimits für Ori.-Achsen)

Mit MD21180 \$MC_ROT_AX_SWL_CHECK_MODE wird festgelegt, wann eine Veränderung der Rundachspositionen zulässig ist:

Wert	Bedeutung
0	Keine Veränderung zulässig (Default, entspricht dem bisherigen Verhalten).
1	Veränderung ist nur zulässig, wenn Achsinterpolation aktiv ist (ORIAxes bzw. ORIMKS).
2	Veränderung ist immer zulässig, auch wenn ursprünglich Vektorinterpolation (Großkreis-, Kegelmantelinterpolation usw.) aktiv war.

Umschalten auf Achsinterpolation

Müssen die Achspositionen gegenüber dem ursprünglich ermittelten Wert verändert werden, wird grundsätzlich auf Achsinterpolation der Rundachsen umgeschaltet, da der ursprünglich vorgesehene Interpolationsweg, z. B. durch Großkreisinterpolation oder Kegelmantelinterpolation, nicht mehr eingehalten werden kann.

Beispiel

Ein Beispiel für die Modifikation der Rundachsbewegung einer 5-Achs-Maschine mit drehbarem Werkzeug ist im Kapitel "Beispiele für generische Achstransformationen (Seite 146)" dargestellt.

2.9.3 Komprimierung der Orientierung

Funktion

Mit den Kompressor-Funktionen COMPON, COMPCURV, COMPCAD und COMPSURF können NC-Programme, in denen die Orientierung mittels Richtungsvektoren programmiert ist, unter Einhaltung einer vorgebbaren Toleranz komprimiert werden.

Voraussetzung

Eine Orientierungsbewegung wird nur unter folgenden Voraussetzungen komprimiert:

- Orientierungstransformation (TRAORI) ist aktiv.
- Großkreisinterpolation ist aktiv.
D. h., die Änderung der Werkzeugorientierung erfolgt in der Ebene, die von Start- und Endorientierung aufgespannt wird.
Eine Großkreisinterpolation wird unter den folgenden Bedingungen durchgeführt:
 - MD21104 \$MC_ORI_IPO_WITH_G_CODE = 0
+ ORIWKS ist aktiv.
+ Orientierung ist mittels Vektoren programmiert (mit A3, B3, C3 bzw. A2, B2, C2).
 - MD21104 \$MC_ORI_IPO_WITH_G_CODE = 1
+ ORIVECT bzw. ORIPLANE ist aktiv.
Die Werkzeugorientierung kann entweder als Richtungsvektor oder mit Rundachspalten programmiert sein. Ist einer der G-Befehle ORICONxx oder ORICURVE aktiv oder sind Polynome für die Orientierungswinkel (PO[PHI] und PO[PSI]) programmiert, wird keine Großkreisinterpolation durchgeführt.

Parametrierung

NC-Sätze sind nur dann komprimierbar, falls man Abweichungen der interpolierten Kontur von der programmierten Kontur zulässt bzw. der interpolierten Orientierung vom programmierten Orientierungslauf.

Die maximale erlaubte Abweichung ist durch Kompressor-Toleranzen einstellbar. Je größer die erlaubten Toleranzen sind, umso mehr Sätze können komprimiert werden. Jedoch weicht die interpolierte Kontur bzw. Orientierung umso mehr von den programmierten Werten ab.

Achsgenauigkeit

Der Kompressor erzeugt für jede programmierte Bahn eine Spline-Kurve so, dass für die daran beteiligten Achsen die mit dem Maschinendatum eingestellte maximale Toleranz in den Endpunkten eingehalten wird:

MD33100 \$MA_COMPRESS_POS_TOL = <Maximale Abweichung bei Kompression>

Konturgenauigkeit

Die bei Kompression erlaubte maximale Abweichung von der programmierten Kontur und Werkzeugorientierung wird über folgende Settingdaten eingestellt:

- SD42475 \$SC_COMPRESS_CONTUR_TOL = <Maximale Konturabweichung>
- SD42476 \$SC_COMPRESS_ORI_TOL = <Maximale Winkel-Abweichung der Werkzeugorientierung>
Nur bei Orientierungstransformation wirksam.
- SD42477 \$SC_COMPRESS_ORI_ROT_TOL = <Maximale Winkel-Abweichung der Werkzeugdrehung>
Nur bei Orientierungstransformation in Zusammenhang mit 6-Achs-Transformation wirksam.

Hinweis

Die Vorgabe einer maximalen Abweichung der Werkzeugorientierung ist nur dann möglich, wenn eine Orientierungstransformation (TRAORI) aktiv ist.

Kompressionsmodus

Mit dem Maschinendatum werden dezimal codiert folgende Funktionen parametrisiert:

- Einerstelle: Berücksichtigung von Toleranzen
- Zehnerstelle: Komprimierung von Sätzen mit programmierter Werkzeugorientierung und / oder Wertzuweisungen.
- Hunderterstelle: Komprimierung von Sätzen außer Linearsätzen (G1).
- Tausenderstelle: Komprimierung für spezielle Anwendungen.

MD20482 \$MC_COMPRESSOR_MODE = <Wert>

Wert	Bedeutung
Einerstelle (WirksameToleranzvorgaben)	
xxx0	Geometrie- und Orientierungsachsen MD33100 \$MA_COMPRESS_POS_TOL
xxx1	Geometrieachsen SD42475 \$SC_COMPRESS_CONTUR_TOL
	Orientierungsachsen MD33100 \$MA_COMPRESS_POS_TOL
xxx2	Geometrieachsen MD33100 \$MA_COMPRESS_POS_TOL
	Orientierungsachsen SD42476 \$SC_COMPRESS_ORI_TOL SD42477 \$SC_COMPRESS_ORI_ROT_TOL
xxx3	Geometrieachsen SD42475 \$SC_COMPRESS_CONTUR_TOL
	Orientierungsachsen SD42476 \$SC_COMPRESS_ORI_TOL SD42477 \$SC_COMPRESS_ORI_ROT_TOL
Zehnerstelle (Komprimierung von Sätzen mit programmierter Werkzeugorientierung und / oder Wertzuweisungen)	
xx0x	Sätze mit programmierter Werkzeugorientierung und/oder Wertzuweisungen werden komprimiert.
xx1x	Sätze mit programmierter Werkzeugorientierung werden komprimiert. Sätze mit Wertzuweisungen werden nicht komprimiert.
xx2x	Sätze mit Wertzuweisungen werden komprimiert. Sätze mit programmierter Werkzeugorientierung werden nicht komprimiert.
xx3x	Sätze mit programmierter Werkzeugorientierung und / oder Wertzuweisungen werden nicht komprimiert.
Hunderterstelle (Komprimierung von Sätzen außer Linearsätzen (G1))	
x0xx	Kreissätze und G0-Sätze werden nicht komprimiert.
x1xx	Kreissätze werden durch COMPCAD linearisiert und komprimiert.
x2xx	G0-Sätze ¹⁾ werden komprimiert.
x3xx	Kreissätze und G0-Sätze ¹⁾ werden komprimiert.

Tausenderstelle (Komprimierung für spezielle Anwendungen)	
0xxx	Optimierung hinsichtlich Oberflächengüte im Werkzeug- und Formenbau.
1xxx	Optimierung hinsichtlich weichem und schnellem Verfahren. Relevant bei Anwendungen wie z. B. Tape-Laying.
1) Durch MD20560 \$MC_G0_TOLERANCE_FACTOR bzw. NC-Befehl <code>STOLF</code> kann für die Komprimierung eine andere Toleranz vorgegeben werden.	

Programmierung

Werkzeugorientierung

Falls eine Orientierungstransformation (TRAORI) aktiv ist, kann bei 5-Achs Maschinen die Werkzeugorientierung folgendermaßen (kinematikunabhängig) programmiert werden:

- Programmierung des Richtungsvektors über:
A3=<...> B3=<...> C3=<...>
- Programmierung der Eulerwinkel bzw. RPY-Winkel über:
A2=<...> B2=<...> C2=<...>

Drehung des Werkzeugs

Bei **6-Achs** Maschinen kann zusätzlich zur Werkzeugorientierung noch die Drehung des Werkzeugs programmiert werden.

Die Programmierung des Drehwinkels erfolgt mit:

THETA=<...>

Hinweis

NC-Sätze, in denen zusätzlich eine Drehung programmiert ist, sind nur dann komprimierbar, falls sich der Drehwinkel **linear** ändert. D. h., für den Drehwinkel darf kein Polynom mit `PO[THT]=(...)` programmiert sein.

Allgemeine Form eines komprimierbaren NC-Satzes

Die allgemeine Form eines komprimierbaren NC-Satzes kann daher wie folgt aussehen:

N... X=<...> Y=<...> Z=<...> A3=<...> B3=<...> C3=<...> THETA=<...> F=<...>

bzw.

N... X=<...> Y=<...> Z=<...> A2=<...> B2=<...> C2=<...> THETA=<...> F=<...>

Programmierung der Werkzeugorientierung durch Rundachspositionen

Die Werkzeugorientierung kann auch durch Rundachspositionen angegeben sein, z. B. in der Form:

N... X=<...> Y=<...> Z=<...> A=<...> B=<...> C=<...> THETA=<...> F=<...>

In diesem Fall wird die Komprimierung auf zwei unterschiedliche Arten durchgeführt, abhängig davon, ob eine Großkreisinterpolation durchgeführt wird oder nicht. Wenn keine Großkreisinterpolation stattfindet, dann wird die komprimierte Orientierungsänderung durch axiale Polynome für die Rundachsen in üblicher Weise dargestellt.

Kompressorfunktion einschalten

Die Kompressor-Funktionen werden durch die modalen G-Befehle `COMPON`, `COMPCURV`, `COMPCAD` bzw. `COMPSURF` eingeschaltet.

Kompressorfunktion ausschalten

Beendet wird die Kompressor-Funktion mit `COMPOF`.

Beispiel

Im nachfolgenden Programmbeispiel wird ein Kreis, der durch einen Polygonzug angenähert ist, komprimiert. Die Werkzeugorientierung bewegt sich dabei synchron dazu auf einem Kegelmantel. Obwohl die aufeinander folgenden programmierten Orientierungsänderungen unstetig verlaufen, generiert die Kompressor-Funktion einen glatten Verlauf der Orientierung.

Programmcode	Kommentar
DEF INT ANZAHL=60	
DEF REAL RADIUS=20	
DEF INT COUNTER	
DEF REAL WINKEL	
N10 G1 X0 Y0 F5000 G64	
\$SC_COMPRESS_CONTUR_TOL=0.05	; Maximale Abweichung der Kontur = 0.05 mm
\$SC_COMPRESS_ORI_TOL=5	; Maximale Abweichung der Orientierung = 5 Grad
TRAORI	
COMPCURV	
	; Es wird ein Kreis gefahren, der aus Polygonen gebildet wird. Die Orientierung bewegt sich dabei auf einem Kegel um die Z-Achse mit einem Öffnungswinkel von 45 Grad.
N100 X0 Y0 A3=0 B3=-1 C3=1	
N110 FOR COUNTER=0 TO ANZAHL	
N120 WINKEL=360*COUNTER/ANZAHL	
N130 X=RADIUS*cos(WINKEL) Y=RADIUS*sin(WINKEL)	
A3=sin(WINKEL) B3=-cos(WINKEL) C3=1	
N140 ENDFOR	

Literatur

Funktionshandbuch Grundfunktionen; Kapitel "B1: Bahnsteuerbetrieb, Genauhalt, LookAhead" > "Kompressor-Funktionen"

2.9.4 Glättung des Orientierungsverlaufs

2.9.4.1 Funktion

Bei vielen durch CAD/CAM-Systemen erzeugten NC-Programmen zur 5-Achs-Bearbeitung kommt es vor, dass zwar der Verlauf der Kontur genügend glatt gemäß der zugrunde liegenden Geometrie verläuft, der Orientierungsverlauf jedoch mehr oder weniger große Schwankungen enthält. Diese Schwankungen der Orientierung führen zu einem sehr unruhigen Verlauf der Orientierungsachsen mit ständigen Beschleunigungs- bzw. Bremsvorgängen. Durch die dann notwendigen Ausgleichsbewegungen der Linearachsen müssen auch diese ständig beschleunigt bzw. abgebremst werden. Diese unnötigen Beschleunigungen führen zu einer starken Begrenzung der möglichen Bahngeschwindigkeit und damit zu einer unnötigen Verlängerung der Bearbeitungszeit.

Mit der Funktion "Glättung des Orientierungsverlaufs" können Schwankungen der Orientierung über mehrere Sätze hinweg geglättet werden. Dadurch wird ein glatter Verlauf sowohl der Orientierung als auch der Kontur erzielt.

Voraussetzungen

Für die Anwendung der Funktion gelten folgende Voraussetzungen:

- Nur in Systemen mit 5/6-Achs-Transformation verfügbar.
- Nur in Verbindung mit der Kompressor-Funktion COMPCAD einsetzbar.

2.9.4.2 Inbetriebnahme

Parametrierung

Anzahl der Sätze

Die Glättung des Orientierungsverlaufs erfolgt über eine einstellbare Anzahl von Sätzen:

```
MD28590 $MC_MM_ORISON_BLOCKS = <Wert>
```

Für die meisten Anwendungen sollten 10 Sätze ausreichend sein. Minimal sollte mindestens der Wert 4 eingegeben werden.

Hinweis

Wird die Glättung des Orientierungsverlaufs aktiviert, ohne dass ausreichend Satzspeicher dafür konfiguriert wurde (MD28590 < 4), dann erfolgt eine Alarmmeldung und die Funktion kann nicht ausgeführt werden.

Maximale Satzweglänge

Der Orientierungsverlauf wird nur in solchen Sätzen geglättet, deren Verfahrensweg kleiner ist als die einstellbare maximale Satzweglänge:

```
MD20178 $MC_ORISON_BLOCK_PATH_LIMIT
```

Sätze mit längeren Verfahrenswegen unterbrechen die Glättung und werden wie programmiert abgefahren.

Maximale Toleranz

Die Glättung des Orientierungsverlaufs erfolgt unter Einhaltung der vorgegebenen maximalen Toleranz (maximale Winkelabweichung der Werkzeugorientierung in Grad):

SD42678 \$SC_ORISON_TOL

Maximaler Bahnweg

Die Glättung erfolgt maximal über den vorgegebenen Bahnweg:

SD42680 \$SC_ORISON_DIST

2.9.4.3 Glättung des Orientierungsverlaufs ein-/ausschalten (ORISON, ORISOF)

Die "Glättung des Orientierungsverlaufs" wird im Teileprogramm durch die Befehle der G-Gruppe 61 ein-/ausgeschaltet. Die Befehle sind modal wirksam.

Voraussetzungen

- System mit 5/6-Achs-Transformation.
- Kompressor-Funktion COMPCAD ist aktiv.

Syntax

```
ORISON
...
ORISOF
```

Bedeutung

ORISON:	Glättung des Orientierungsverlaufs einschalten
ORISOF:	Glättung des Orientierungsverlaufs ausschalten

Beispiel

Programmcode	Kommentar
...	
TRAORI ()	; Einschalten der Orientierungstransformation.
COMPCAD	; Einschalten der Kompressor-Funktion COMPCAD.
ORISON	; Einschalten der Orientierungsglättung.
\$SC_ORISON_TOL=1.0	; Maximale Winkelabweichung der Werkzeugorientierung = 1,0 Grad.
G91	
X10 A3=1 B3=0 C3=1	
X10 A3=-1 B3=0 C3=1	

Programmcode	Kommentar
X10 A3=1 B3=0 C3=1	
X10 A3=-1 B3=0 C3=1	
X10 A3=1 B3=0 C3=1	
X10 A3=-1 B3=0 C3=1	
X10 A3=1 B3=0 C3=1	
X10 A3=-1 B3=0 C3=1	
X10 A3=1 B3=0 C3=1	
X10 A3=-1 B3=0 C3=1	
...	
ORISOF	; Ausschalten der Orientierungsglättung.
...	

Die Orientierung wird um 90 Grad in der XZ–Ebene von -45 bis +45 Grad geschwenkt. Durch die Glättung des Orientierungsverlaufs erreicht die Orientierung nicht mehr die maximalen Winkelwerte von -45 bzw. +45 Grad.

2.9.5 Bahnrelative Orientierung (ORIPATH, ORIPATHS, ORIROT)

Funktionalität

Die bisherige Programmierung der Werkzeugorientierung wird unabhängig von bestimmten technologischen Anwendungen dahingehend verbessert, dass die programmierte relative Orientierung bezogen auf die gesamte Bahn eingehalten wird. Die dabei notwendigen Abweichungen vom idealen Orientierungsverlauf können vorgegeben werden, falls z. B. eine Ecke in der Kontur auftritt.

Die Werkzeugorientierung kann sowohl über projektierbare Maschinendaten als auch über neue Sprachbefehle im Teileprogramm verändert werden. Dadurch ist es möglich, die relative

Orientierung zur Bahn nicht nur am Satzende, sondern über den gesamten Bahnverlauf einzuhalten. Die gewünschte Orientierung wird erreicht:

- Durch einstellbare Orientierungsarten bei `ORIPATH`, wie relativ zur Bahn interpoliert werden soll.
- Ob die Werkzeugorientierung entweder immer stetig mit vorgebbaren Abweichungen vom idealen bahnrelativen Orientierungsverlauf an einem Satzübergang verlaufen soll, **oder** ob der Sprung der Orientierung in einem eigenen, eingefügten Zwischensatz geglättet werden soll. In diesem Fall wird dann die Bahnbewegung in der Konturrecke gestoppt.
- Bei 6-Achs-Transformationen gibt es zwei Möglichkeiten:
 - Die Werkzeugorientierung sowie die Drehung der Orientierung wird relativ zur Bahn interpoliert (`ORIPATH`, `ORIPATHS`).
 - Der Orientierungsvektor wird wie üblich programmiert und interpoliert. Die Drehung des Orientierungsvektors wird bei `ORIROTC` relativ zur Bahntangente angestellt.

Hinweis

Die bahnrelative Orientierungsinterpolation mit `ORIPATH` bzw. `ORIPATHS` und `ORIROTC` kann **nicht zusammen** mit der Funktion "Orientierungsglättung" verwendet werden. Hierfür muss im Teileprogramm `OSOF` aktiv sein. Anderenfalls wird der Alarm 10980 "Orientierungsglättung nicht möglich" ausgegeben.

Abweichung von der gewünschten Orientierung

Während der Interpolation des Satzes kann die Orientierung mehr oder weniger stark von der gewünschten relativen Orientierung abweichen. Es wird die im Vorgängersatz erreichte Orientierung mittels Großkreisinterpolation in die programmierte Endorientierung überführt. Die dadurch bedingte Abweichung von der gewünschten relativen Orientierung hat zwei wesentliche Ursachen:

1. Die **Endorientierung des Vorgängersatzes** bezieht sich auf die Tangente und dem Flächennormalvektor am Ende des Vorgängersatzes. Beide können am Anfang des aktuellen Satzes davon abweichen. Damit hat im aktuellen Satz die Startorientierung nicht die gleiche Ausrichtung im Bezug zur Tangente und zum Flächennormalvektor wie am Ende des Satzes.
2. Sowohl die Tangente als auch der Flächennormalvektor können sich **über den ganzen Satz ändern**. Dies ist dann der Fall, wenn für die Geoachsen Kreise, Splines oder Polynome programmiert werden, oder wenn für den Flächennormalvektor sowohl ein Start- als auch ein Endwert programmiert wird. In diesem Fall muss sich die Werkzeugorientierung während der Interpolation des Satzes geeignet ändern, um in jedem Bahnpunkt denselben Bezug zur Bahntangente und zum Flächennormalvektor zu haben.

Parametrierung: Maschinendaten

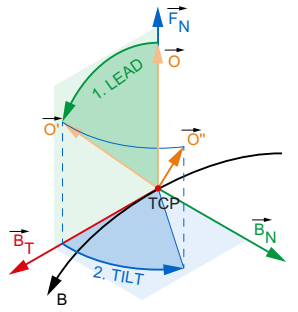
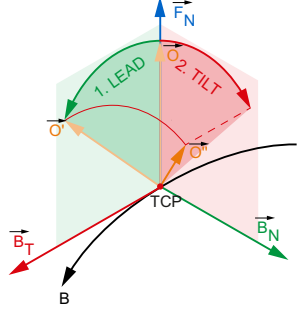
Einstellung für bahnrelative Orientierung

Mit dem Maschinendatum werden dezimal codiert folgende Funktionen parametrieren:

- Einerstelle: Bahnrelative Orientierung
- Zehnerstelle: Interpretation der Drehwinkel `LEAD` und `TILT`

- Hunderterstelle: Aktivierung und Definition der Richtung der Abhebebewegung bei Umorientierungen während aktivem ORIPATH
- Tausenderstelle: Verhalten der bahnrelativen Orientierung in Aktivierungs- / Deaktivierungssätzen der Werkzeugkorrektur

MD21094 \$MC_ORIPATH_MODE = <Wert>

Wert	Bedeutung
Einerstelle (Bahnrelative Orientierung)	
xxx0	Die Werkzeugorientierung hat nur am Satzende den mit LEAD und TILT programmierten Bezug zu Bahntangente und Flächennormalvektor. Während des Satzes folgt die Orientierung nicht der Bahntangente.
xxx1	Die Werkzeugorientierung hat im ganzen Satz den mit LEAD und TILT programmierten Bezug zu Bahntangente und Flächennormalvektor.
Zehnerstelle (Interpretation der Drehwinkel LEAD und TILT)	
Bei bahnrelativer Orientierung wird das Koordinatensystem durch die Vektoren Bahntangente T und programmiertem Flächennormalvektor N aufgespannt.	
Hinweis 1: Programmierung des Flächennormalvektors N	
<ul style="list-style-type: none"> • am Satzanfang: A4=... B4=... C4=... • am Satzende: A5=... B5=... C5=... 	
xx0x	 <div style="margin-left: 20px;"> <ol style="list-style-type: none"> 1. Drehung LEAD: Drehung des Orientierungsvektors O um den Bahnnormalenvektor $B_N \Rightarrow O'$ 2. Drehung TILT: Drehung des neuen Orientierungsvektors O' um den Flächennormalenvektor $F_N \Rightarrow O''$ </div>
xx1x	 <div style="margin-left: 20px;"> <ol style="list-style-type: none"> 1. Drehung LEAD: Drehung des Orientierungsvektors O um den Bahnnormalenvektor $B_N \Rightarrow O'$ 2. Drehung TILT: Drehung des neuen Orientierungsvektors O' um den Bahntangentenvektor $B_T \Rightarrow O''$ </div>

xx2x		<p>1. Drehung LEAD: Drehung des Orientierungsvektors O und des Bahntangentenvektor $B_T \Rightarrow O'$ und B'_T</p> <p>2. Drehung TILT: Drehung des neuen Orientierungsvektors O' um den neuen Bahntangentenvektor $B'_T \Rightarrow O''$</p>
xx3x		<p>1. Drehung TILT: Drehung des Orientierungsvektors O um den Bahntangentenvektor $B_T \Rightarrow O'$</p> <p>2. Drehung LEAD: Drehung des neuen Orientierungsvektors O' um den Bahnnormalenvektor $B_N \Rightarrow O''$</p>
xx4x		<p>1. Drehung TILT: Drehung des Orientierungsvektors O und des Bahnnormalenvektor B_N um den Bahntangentenvektor $B_T \Rightarrow O'$ und B'_N</p> <p>2. Drehung LEAD: Drehung des neuen Orientierungsvektors O' um den neuen Bahnnormalenvektor $B'_N \Rightarrow O''$</p>
<p>Hunderterstelle</p> <p>(Aktivierung und Definition der Richtung der Abhebebewegung bei Umorientierungen während aktivem ORIPATH)</p> <p>Hinweis</p> <p>Eine programmierte Abhebebewegung wird nur durchgeführt, bei:</p> <ul style="list-style-type: none"> • Einerstelle == 1 UND • Länge des Abhebevektors $\neq 0.0$ 		
x0xx	Es wird keine Abhebebewegung ausgeführt.	
x1xx	Es erfolgt eine Abhebebewegung in Richtung des programmierten Abhebevektors im Werkzeug koordinatensystem. Der Abhebevektor bezieht sich dabei auf das Koordinatensystem, das durch die aktuelle Werkzeugrichtung (z-Koordinate) und der Orientierungsänderung (x-Koordinate) definierte ist.	
x2xx	Es erfolgt eine Abhebebewegung in Richtung des programmierten Abhebevektors im Werkstück koordinatensystem. Der Abhebevektor bezieht sich dabei auf das Koordinatensystem, das durch die aktive Ebene (z-Koordinate ist der Flächennormalvektor der aktiven Ebene) und die Orientierungsänderung (x-Koordinate) definierte ist.	
<p>Tausenderstelle</p> <p>(Verhalten der bahnrelativen Orientierung in Aktivierungs- / Deaktivierungssätzen der Werkzeugkorrektur)</p>		

0xxx	Die bahnrelative Orientierung wird auch in Aktivierungs- bzw. Deaktivierungssätzen der Werkzeugkorrektur eingehalten .
1xxx	Die bahnrelative Orientierung wird in Aktivierungs- bzw. Deaktivierungssätzen der Werkzeugkorrektur nicht eingehalten . Hinweis In Aktivierungs- bzw. Deaktivierungssätzen der Werkzeugkorrektur bleibt die Werkzeugorientierung normalerweise konstant. Es ist jedoch erlaubt in diesen Sätzen eine Werkzeugorientierung zu programmieren, die dann in diesen Sätzen verfahren wird. Die Programmierung der Orientierung kann in diesen Sätzen jedoch nur mit Vektoren erfolgen, die Programmierung von Rundachspositionen ist nicht erlaubt.

Adressnamen für die Komponenten des Abhebevektors

Mit dem Maschinendatum werden die Adressnamen für die Komponenten des Abhebevektors definiert:

MD10624 \$MN_ORIPATH_LIFT_VECTOR_TAB[<Vektorkomponenten>] = <Name>

Beispiel mit Standarddaten

Maschinendatum

\$MN_ORIPATH_LIFT_VECTOR_TAB[0] = "A8" ; x-Koordinate

\$MN_ORIPATH_LIFT_VECTOR_TAB[1] = "B8" ; y-Koordinate

\$MN_ORIPATH_LIFT_VECTOR_TAB[2] = "C8" ; z-Koordinate

Programmierung

ORIPATHS A8=X_KOORD B8=Y_KOORD C8=Z_KOORD

Hinweis

Namenskonvention

Es sind die bei MD20080 \$MC_AXCONF_CHANAX_NAME_TAB beschriebenen Regeln für Achsbezeichner zu beachten.

Adressnamen für den Sicherheitsfaktor bei Orientierungsänderung

Normalerweise erfolgt die Abhebebewegung simultan zur Orientierungsänderung. Mit dem Maschinendatum kann der Adressnamen für den Sicherheitsfaktor R definiert werden:

MD10626 \$MN_ORIPATH_LIFT_FACTOR_NAME = "<Name>"

Ist der Sicherheitsfaktor R größer 0.0, wird die Orientierung erst dann geändert, wenn das Werkzeug um den Sicherheitsabstand S in Richtung des Abhebevektors verfahren ist. Der Sicherheitsabstand S berechnet sich zu:

$S = R \cdot \text{Betrag des Abhebevektors}$; Definitionsbereich des Faktors R: $0 \leq R < 1$

Beispiel mit **Standarddaten**

Maschinendatum

`$MN_ORIPATH_LIFT_FACTOR_NAME = "ORIPLF"`

Programmierung

`ORIPATHS A8=X_KOORD B8=Y_KOORD C8=Z_KOORD ; Abhebevektor`

`ORIPLF=0.1 ; Sicherheitsabstand = 0.1 * Betrag des Abhebevektors
(A8, B8, C8)`

Hinweis

Namenskonvention

Es sind die bei MD20080 \$MC_AXCONF_CHANAX_NAME_TAB beschriebenen Regeln für Achsbezeichner zu beachten.

Parametrierung: Settingdatum SD42670 (Bahnstrecke zur Orientierungsglättung)

Mit dem Settingdatum wird eine Bahnstrecke vorgegeben, innerhalb der bei einem Sprung der Werkzeugorientierung an einem Satzübergang vom programmierten Orientierungsverlauf abgewichen werden darf, um den Orientierungsverlauf zu glätten. Wird diese Bahnstrecke zu klein eingestellt, muss gegebenenfalls die Bahngeschwindigkeit beträchtlich reduziert werden.

`SD42670 $SC_ORIPATH_SMOOTH_DIST = <Bahnlänge>`

Verhalten bei Bahnlänge 0.0

Wird als Bahnlänge der Wert 0.0 eingestellt, wird zur Glättung des Orientierungsverlaufs, falls notwendig, ein eigener Zwischensatz eingefügt. Dabei wird dann auf der Bahn so lange angehalten, bis die Orientierungsänderung im Zwischensatz ausgeführt wurde. Die Orientierungsänderung erfolgt dabei aber nur dann **beschleunigungsstetig**, wenn `ORIPATHS` aktiv ist. Andernfalls erfolgt die Orientierungsänderung von Start- zur Endorientierung mittels linearer Großkreisinterpolation.

Während des Verfahrens der Orientierungsachsen zur Orientierungsänderung kann eine Abhebebewegung des Werkzeugs durchgeführt werden. Die Abhebebewegung wird über die Hunderterstelle von MD21094 \$MC_ORIPATH_MODE (siehe Absatz oben: "Parametrierung: Maschinendaten" > "Einstellung für bahnrelative Orientierung") aktiviert.

Die Richtung und Weglänge der Abhebebewegung wird durch den programmierten Abhebevektor (siehe Absatz oben: "Parametrierung: Maschinendaten" > "Adressnamen für die Komponenten des Abhebevektors"). Ist die Länge dieses Vektors gleich Null erfolgt keine Abhebebewegung.

Normalerweise erfolgt die Abhebebewegung simultan zur Orientierungsänderung. Es kann aber auch ein "Sicherheitsabstand" programmiert werden (siehe Absatz oben: "Parametrierung: Maschinendaten" > "Adressnamen für den Sicherheitsfaktor bei Orientierungsänderung").

Parametrierung: Settingdatum SD42672 (Toleranz zur Orientierungsglättung)

Ist an einem Satzübergang nur der Verlauf der Bahn- und der Bahngeschwindigkeit, nicht aber der Bahnbeschleunigung stetig (z. B. tangentialer Übergang Gerade / Kreis), ist bezüglich der Orientierung nur der Orientierungsverlauf, nicht aber die Orientierungsänderung, stetig. Dies führt zu einem im Allgemeinen nicht erwünschten Geschwindigkeitssprung in den Orientierungsachsen. Im Settingdatum kann ein Toleranzfenster parametrierbar werden, innerhalb dessen der verfahrenere Orientierungsverlauf vom programmierten Orientierungsverlauf abweichen darf, um den Orientierungsverlauf zu glätten.

SD42672 \$SC_ORIPATH_SMOOTH_TOL = <Toleranz>

Diese Art der Orientierungsglättung wird nur ausgeführt, wenn gilt:

- ORIPATHS ist aktiv **UND**
- SD42672 \$SC_ORIPATH_SMOOTH_TOL > 0.0

Bahnrelative Interpolation der Drehung (ORIROTC)

Bei **6-Achs-Transformationen** gibt es neben der bahnrelativen Interpolation der gesamten Komponenten der Werkzeugorientierung (Drehungen um jede der drei Rotationsachsen), mittels `ORIROTC` die Möglichkeit, dass nur die Drehung des Werkzeugs um den Orientierungsvektor relativ zur Bahntangente interpoliert wird.

Der Orientierungsvektor kann dabei weiterhin mit Richtungskomponenten bzw. über Euler- oder RPY-Winkel programmiert und die Interpolationsart mittels `ORIVECT`, `ORIXES`, `ORICONxx`, `ORICURVE` festgelegt werden (siehe Kapitel "Drehungen des Orientierungsvektors (Seite 125)").

2.9.6 Programmierung von Orientierungspolynomen

Funktion

Die Orientierungspolynome und auch Achspolynome können unabhängig davon, welche Polynominterpolation gerade aktiv ist, mit unterschiedlichen Polynomarten programmiert sein. Dies lässt sich anwenden auf:

- Linearinterpolation mit G-Befehl G01
- Polynominterpolation mit G-Befehl POLY
- Kreisinterpolation mit G-Befehl G02, G03 oder CIP
- Evolventeninterpolation mit G-Befehl INVCW oder INVCCW

Damit ist es möglich, für eine Kontur **gleichzeitig** mehrere Polynome zu programmieren.

Hinweis

Weitere Informationen zur Programmierung von Achspolynomen mit PO[X], PO[Y], PO[Z] und Orientierungspolynomen wie PO[PHI], PO[PSI], PO[THT] sowie PO[XH], PO[YH], PO[ZH] siehe:

Literatur:

Programmierhandbuch Arbeitsvorbereitung

Es werden zwei verschiedene Typen von Orientierungspolynomen definiert:

- Polynome für Winkel in Bezug auf die Ebene, die von Start- und Endorientierung aufgespannt wird (Orientierungspolynome vom Typ 1)
- Polynome für Koordinaten der Raumkurve eines Bezugspunkts auf dem Werkzeug (Orientierungspolynome vom Typ 2)

Polynome vom Typ 1

Orientierungspolynome vom Typ 1 sind Polynome **für Winkel**

PO[PHI]: Winkel in der Ebene zwischen Start- und Endorientierung

PO[PSI]: Winkel, der die Auskippung der Orientierung aus der Ebene zwischen Start- und Endorientierung beschreibt

Polynome vom Typ 2

Orientierungspolynome vom Typ 2 sind Polynome **für Koordinaten**

PO[XH]: x-Koordinate des Bezugpunktes auf dem Werkzeug

PO[YH]: y-Koordinate des Bezugpunktes auf dem Werkzeug

PO[ZH]: z-Koordinate des Bezugpunktes auf dem Werkzeug

Polynome für Drehwinkel und Drehvektoren

Bei **6-Achstransformationen** kann zur Werkzeugorientierung die Drehung des Werkzeugs um sich selbst programmiert werden. Diese Drehung einer dritten Rundachse wird entweder durch einen Drehwinkel oder durch einen Drehvektor, der in der Ebene senkrecht zur Werkzeugrichtung liegt, beschrieben.

Zusätzlich kann in den drei Fällen ein Polynom **für die Drehung** mit PO[THT] des Orientierungsvektors programmiert werden. Dies ist immer dann möglich, wenn die jeweilige kinematische Transformation auch Drehwinkel unterstützt.

Drehwinkel mit ORIPATH und ORIPATHS

Bei **bahnrelativer Orientierungsinterpolation** mit ORIPATH bzw. ORIPATHS kann die zusätzliche Drehung durch Programmierung des Winkels $\text{THETA}=\langle \dots \rangle$ erfolgen. Zusätzlich können für diesen Drehwinkel mit $\text{PO}[\text{THT}] = (\dots)$ Polynome maximal 5. Grades programmiert werden.

Die 3 möglichen Winkel, wie Voreilwinkel, Seitwärtswinkel und Drehwinkel, haben bezogen auf die Drehwirkung folgende Bedeutung:

LEAD: Winkel relativ zum Flächennormalenvektor, in der von Bahntangente und Flächennormalenvektor aufgespannten Ebene.

TILT: Drehung der Orientierung um z-Richtung bzw. Drehung um die Bahntangente.

THETA: Drehung um die Werkzeugrichtung. Ist nur möglich, falls die Werkzeugorientierung insgesamt 3 Freiheitsgrade hat (siehe Kapitel "Erweiterung der Generischen Transformation auf 6 Achsen - nur 840D sl (Seite 79)").

Wie die Winkel LEAD und TILT interpretiert werden sollen, kann mit dem folgenden Maschinendatum eingestellt werden:

MD21094 \$MC_ORIPATH_MODE (Einstellung für bahnrelative Orientierung ORIPATH)

Zusätzlich zu den mit LEAD und TILT programmierten konstanten Winkeln können für den Voreilwinkel und für den Seitwärtswinkel Polynome programmiert werden. Die Polynome werden mit den Winkeln PHI und PSI programmiert:

$\text{PO}[\text{PHI}] = (a_2, a_3, a_4 \text{ Polynom für den Voreilwinkel LEAD}, a_5):$

$\text{PO}[\text{PSI}] = (b_2, b_3, b_4 \text{ Polynom für den Seitwärtswinkel TILT}, b_5):$

Für beide Winkel können Polynome maximal 5. Grades programmiert werden. Die Werte der Winkel am Satzende werden dabei mit den NC-Adressen $\text{LEAD}=\langle \dots \rangle$ bzw. $\text{TILT}=\langle \dots \rangle$ programmiert.

Bei der Programmierung können die höheren Polynomkoeffizienten, die Null sind, weggelassen werden. Zum Beispiel wird mit $\text{PO}[\text{PHI}] = (a_2)$ eine Parabel für den Voreilwinkel LEAD programmiert.

Drehungen der Drehvektoren mit ORIOTC

Der Drehvektor wird relativ zur Bahntangente mit einem durch den Winkel THETA programmierbaren Offset interpoliert.

Dabei kann für den Offsetwinkel auch mit $PO[THT] = (c2, c3, c4, c5)$ ein Polynom maximal 5. Grades programmiert werden.

Hinweis

Ist ORIAXES aktiv, d. h. die Werkzeugorientierung wird über Achsinterpolation interpoliert, wird die bahnrelative Anstellung des Drehvektors nur am Satzende erfüllt.

Weitere Informationen zur Programmierung siehe:

Literatur:

Programmierhandbuch Arbeitsvorbereitung; Transformationen, Interpolationsart (ORIPATH, ORIPATHS)

Randbedingungen

Orientierungspolynomen können nur bei bestimmten Interpolationsarten, die sowohl die Kontur als auch der Orientierung beeinflussen sinnvoll programmiert werden. Zur Vermeidung unerlaubter Programmierungen sind dann einige Randbedingungen zu berücksichtigen:

Orientierungspolynome können nicht programmiert werden,

- wenn Splineinterpolationen ASPLINE, BSPLINE, CSPLINE aktiv sind.
Polynome für Orientierungswinkel vom Typ 1 sind für jede Interpolationsart außer Spline d. h. bei Linearinterpolation mit Eilgang G00 bzw. mit Vorschub G01 und Polynominterpolation POLY und Kreis- bzw. Evolventeninterpolation G02, G03, CIP, CT, INVCW und INVCCW möglich.
Orientierungspolynome vom Typ 2 sind dagegen nur möglich, wenn Linearinterpolation mit Eilgang G00 bzw. mit Vorschub G01 oder Polynominterpolation POLY aktiv ist.
- wenn die Orientierung mittels Achsinterpolation ORIAXES interpoliert wird.
In diesem Fall können direkt Polynome mit $PO[A]$ und $PO[B]$ für die Orientierungsachsen A und B programmiert werden.

Ist ORICURVE aktiv, werden die kartesischen Komponenten des Orientierungsvektors interpoliert, und nur Orientierungspolynome vom Typ 2 sind möglich. Orientierungspolynome vom Typ 1 sind dagegen nicht erlaubt.

Bei Großkreisinterpolation und Kegelinterpolation mit ORIVECT, ORIPLANE, ORICONxxx sind nur Orientierungspolynome vom Typ 1 möglich. Dagegen sind Orientierungspolynome vom Typ 2 nicht erlaubt.

Alarmer

Ein unzulässig programmierter Orientierungspolynom wird mit folgenden Alarmen gemeldet:

- Alarm 14136: Orientierungspolynom ist generell nicht erlaubt.
- Alarm 14137: Polynome $PO[PHI]$ und $PO[PSI]$ sind nicht erlaubt.
- Alarm 14138: Polynome $PO[XH]$, $PO[YH]$, $PO[ZH]$ sind nicht erlaubt.
- Alarm 14139: Polynom für Drehwinkel $PO[THT]$ ist nicht erlaubt.

2.9.7 Systemvariable für Werkzeugorientierung

Die Orientierung des Werkzeugs kann in verschiedenen Koordinatensystemen (BKS, WKS, ENS) sowohl über Systemvariablen als auch über BTSS-Variablen gelesen werden.

Werkzeugorientierung im BKS

Systemvariable		Bedeutung
\$AC_TOOLO_ACT[<i>]	; <i> = 1, 2, 3	i-te Komponente des Vektors der aktuellen Sollorientierung
\$AC_TOOLO_END[<i>]	; <i> = 1, 2, 3	i-te Komponente des Vektors der Endorientierung des aktuellen Satzes
\$AC_TOOLO_DIFF		Restwinkel in Grad, d. h. dies ist der Winkel zwischen den Vektoren \$AC_TOOLO_END[<i>] und \$AC_TOOLO_ACT[<i>]
\$VC_TOOLO[<i>]	; <i> = 1, 2, 3	i-te Komponente des Vektors der Istorientierung
\$VC_TOOLO_DIFF		Winkel in Grad zwischen Soll- und Istorientierung
\$VC_TOOLO_STAT		Statusvariable für Istorientierung Zeigt an, ob die Berechnung der Istorientierung durchgeführt werden kann. Folgende Werte sind möglich:
	0	Istorientierung kann berechnet werden.
	-1	Istorientierung kann nicht berechnet werden, da die momentan aktive Transformation diese Werte nicht in Echtzeit berechnen kann.

Diese Systemvariablen können generell sowohl vom Teileprogramm als auch in Synchronaktionen gelesen werden. Schreibzugriffe sind nicht erlaubt.

Hinweis

Die Komponenten der Vektoren \$AC_TOOLO_ACT[<i>], \$AC_TOOLO_END[<i>] und \$VC_TOOLO[<i>] der Orientierung sind so normiert, dass der Orientierungsvektor den Betrag 1 hat.

Drehvektor im BKS

Bei einer 6-Achs-Kinematik gibt es außer der Orientierung des Werkzeugs auch eine Drehung des Werkzeugs, die veränderbar ist.

Systemvariable		Bedeutung
\$P_TOOLROT[<i>]	; <i> = 1, 2, 3	i-te Komponente des aktuellen Drehvektors im NC-Programm
\$AC_TOOLR_ACT[<i>]	; <i> = 1, 2, 3	i-te Komponente des Vektors des aktuellen Sollwerts der Drehung der Orientierung
\$AC_TOOLR_END[<i>]	; <i> = 1, 2, 3	i-te Komponente des Vektors des Sollwerts der Drehung am Ende des aktuellen Satzes

Systemvariable		Bedeutung
\$AC_TOOLR_DIFF		Restwinkel in Grad, d. h. dies ist der Winkel zwischen den Vektoren \$AC_TOOLR_END[<i>] und \$AC_TOOLR_ACT[<i>].
\$VC_TOOLR[<i>]	; <i> = 1, 2, 3	i-te Komponente des Vektors des Istwerts der Drehung der Orientierung
\$VC_TOOLR_DIFF		Winkel in Grad zwischen Soll- und Istwert der Drehung der Orientierung
\$VC_TOOLR_STAT		Statusvariable für Istwert des Drehvektors Zeigt an, ob die Berechnung des Istwerts des Drehvektors durchgeführt werden kann. Folgende Werte sind möglich:
		0 Istwert des Drehvektors kann berechnet werden.
		-1 Istwert des Drehvektors kann nicht berechnet werden, da die momentan aktive Transformation diese Werte nicht in Echtzeit berechnen kann.

Orientierung und Drehung des Werkzeugs in verschiedenen Koordinatensystemen (BKS, WKS, ENS)

Werkzeugorientierung

Systemvariable		Bedeutung
\$P_TOOL_O[<i>,<j>]	; <i> = 1, 2, 3 ; <j> = 0, 1, 2	i-te Komponente des aktuellen Orientierungsvektors im NC-Programm im Koordinatensystem <j>
		<j> = 0: BKS
		<j> = 1: WKS
		<j> = 2: ENS
\$AC_TOOL_O_ACT[<i>,<j>]	; <i> = 1, 2, 3 ; <j> = 0, 1, 2	i-te Komponente des aktuellen Orientierungsvektors im Koordinatensystem <j>
\$AC_TOOL_O_END[<i>,<j>]	; <i> = 1, 2, 3 ; <j> = 0, 1, 2	i-te Komponente der Endorientierung des aktuellen Satzes im Koordinatensystem <j>
\$AC_TOOL_O_DIFF[<j>]	; <j> = 0, 1, 2	Restwinkel des Orientierungsvektors in Grad in verschiedenen Koordinatensystemen <j>
\$VC_TOOL_O[<i>,<j>]	; <i> = 1, 2, 3 ; <j> = 0, 1, 2	i-te Komponente des Vektors der Istorientierung in verschiedenen Koordinatensystemen <j>
\$VC_TOOL_O_DIFF[<j>]	; <j> = 0, 1, 2	Winkel in Grad zwischen Soll- und Istorientierung in verschiedenen Koordinatensystemen <j>

Drehvektor (nur bei 6-Achs-Kinematiken)

Systemvariable		Bedeutung
\$P_TOOL_R[<i>,<j>]	; <i> = 1, 2, 3 ; <j> = 0, 1, 2	i-te Komponente des aktuellen Drehvektors im NC-Programm im Koordinatensystem <j>
		<j> = 0: BKS
		<j> = 1: WKS
		<j> = 2: ENS

Systemvariable		
\$AC_TOOL_R_ACT[<i>,<j>]	; <i> = 1, 2, 3 ; <j> = 0, 1, 2	i-te Komponente des aktuellen Drehvektors im Koordinatensystem <j>
\$AC_TOOL_R_END[<i>,<j>]	; <i> = 1, 2, 3 ; <j> = 0, 1, 2	i-te Komponente des Drehvektors am Ende des aktuellen Satzes im Koordinatensystem <j>
\$AC_TOOL_R_DIFF[<j>]	; <j> = 0, 1, 2	Restwinkel des Drehvektors in Grad in verschiedenen Koordinatensystemen <j>
\$VC_TOOL_R[<i>,<j>]	; <i> = 1, 2, 3 ; <j> = 0, 1, 2	i-te Komponente des Istwerts des Drehvektors in verschiedenen Koordinatensystemen <j>
\$VC_TOOL_R_DIFF[<j>]	; <j> = 0, 1, 2	Winkel in Grad zwischen Soll- und Istwert des Drehvektors in verschiedenen Koordinatensystemen <j>

Randbedingungen

Der Istwert der Werkzeugorientierung wird nicht von allen Transformationen in Echtzeit bereitgestellt. In diesem Fall können die Variablen \$VC_TOOLO[<i>] bzw. VC_TOOL_O[<i>] und \$VC_TOOLO_DIFF bzw. \$VC_TOOL_O_DIFF nicht berechnet werden. Die Komponenten von \$VC_TOOLO[<i>] bzw. \$VC_TOOL_O[<i>] sind alle Null und die Statusvariable \$VC_TOOLO_STAT liefert den Wert "-1". Dasselbe gilt für die Istwerte des Drehvektors \$VC_TOOLR[<i>] bzw. \$VC_TOOL_R[<i>,<j>].

2.10 Orientierungsachsen

Drehrichtungen

Die Richtungen, um die gedreht wird, werden durch die Achsen des Bezugssystems festgelegt. Das Bezugssystem wird durch die Befehle ORIMKS und ORIWKS festgelegt:

- ORIMKS: Bezugssystem = Basiskoordinatensystem
- ORIWKS: Bezugssystem = Werkstückkoordinatensystem

Reihenfolge der Drehungen

Die Reihenfolge, in der die Orientierungsachsen drehen, ist durch das folgende Maschinendatum festgelegt:

MD21120 \$MC_ORIAX_TURN_TAB_1[0..2] (Bezugsachsensdefinition für ORI-Achsen)

1. Erste Drehung um die im folgenden Maschinendatum angegebene Achse des Bezugssystems:
MD21120 \$MC_ORIAX_TURN_TAB_1[0]
2. Zweite Drehung um die im folgenden Maschinendatum angegebene verdrehte Achse des Bezugssystems:
MD21120 \$MC_ORIAX_TURN_TAB_1[1]
3. Dritte Drehung um die im folgenden Maschinendatum angegebene verdrehte Achse des Bezugssystems:
MD21120 \$MC_ORIAX_TURN_TAB_1[2]

Richtung des Werkzeugvektors

Die Richtung des Werkzeugvektors in der Grundstellung der Maschine wird durch die folgenden Maschinendaten festgelegt:

MD24580 \$MC_TRAFO5_TOOL_VECTOR_1 (Orientierungsvektorrichtung) bzw.

MD24680 \$MC_TRAFO5_TOOL_VECTOR_2 (Orientierungsvektorrichtung)

Zuordnung zu den Kanalachsen

Über die Maschinendaten MD24585 \$MC_TRAFO5_ORIAX_ASSIGN_TAB_1[0..2] (ORI/Kanalachs-Zuordnung Transformation 1) werden die bis zu 3 virtuellen Orientierungsachsen dem Kanal zugeordnet, welche als Eingangsgrößen laut den Maschinendaten \$MC_TRAFO_AXES_IN_n[4..6] (Achszuordnung für Transformation n) eingehen.

Für die Zuordnung der Kanalachsen zu den Orientierungsachsen gilt:

- \$MC_TRAFO5_ORIAX_ASSIGN_TAB_n[0] = \$MC_TRAFO_AXES_IN_n[4]
- \$MC_TRAFO5_ORIAX_ASSIGN_TAB_n[1] = \$MC_TRAFO_AXES_IN_n[5]
- \$MC_TRAFO5_ORIAX_ASSIGN_TAB_n[2] = \$MC_TRAFO_AXES_IN_n[6]

Orientierungs-Transformation 1:

MD24585 \$MC_TRAFO5_ORIAX_ASSIGN_TAB_1[n] n = Kanalachse [0..2]

Orientierungs-Transformation 2:

MD24685 \$MC_TRAFO5_ORIAX_ASSIGN_TAB_2[n] n = Kanalachse [0..2]

Transformation [1..4]

MD24110 \$MC_TRAFO5_AXES_IN_1[n] (Achszuordnung n = Achsindex [0..7] für Transformation)

bis

MD24410 \$MC_TRAFO5_AXES_IN_4[n] (Achszuordnung für Transformation 4)

Transformation [5..8]
MD24432 \$MC_TRAFO5_AXES_IN_5[n] (Achszuordnung n = Achsindex [0..7]
für Transformation 5)
bis
MD24462 MC_TRAFO5_AXES_IN_8[n] (Achszuordnung
für Transformation 8)

Beispiel

Für Orientierungsachsen siehe Kapitel "Beispiel für Orientierungsachsen (Seite 142)".

2.10.1 Betriebsart JOG

Voraussetzungen

Orientierungsachsen können erst dann in der Betriebsart JOG verfahren werden, wenn folgende Voraussetzungen erfüllt sind:

- Die Orientierungsachse muss als solche definiert sein, d. h. im folgenden Maschinendatum muss ein Wert gesetzt sein:
MD24585 \$MC_TRAFO5_ORIAX_ASSIGN_TAB (ORI/Kanalachszuordnung
Transformation 1)
- Eine Transformation muss aktiv sein (TRAORI).

Verfahren über Verfahrtaasten

Beim kontinuierlichen und inkrementellen Verfahren über Verfahrtaasten ist zu beachten, dass simultan nur **eine** Orientierungsachse verfahren werden kann.

Wird mehr als eine Orientierungsachse verfahren, wird der Alarm 20062 "Kanal 1 Achse 2 bereits aktiv" ausgegeben.

Verfahren über Handräder

Über die Handräder können die Orientierungsachsen auch gleichzeitig verfahren werden.

Geschwindigkeit

Für das manuelle Verfahren von Orientierungsachsen wirkt der kanalspezifische Vorschub-Override-Schalter bzw. der Eilgang-Override-Schalter bei Eilgangüberlagerung.

Normalerweise werden die Geschwindigkeiten beim Verfahren in JOG immer von den Geschwindigkeiten der Maschinenachsen abgeleitet. Bei Geometrie- und Orientierungsachsen besteht jedoch nicht unbedingt eine direkte Zuordnung zu einer

Maschinenachse. Deshalb existieren für Geometrieachsen und Orientierungsachsen eigene Maschinendaten, die eine separate Geschwindigkeitsvorgabe ermöglichen:

- MD21150 \$MC_JOG_VELO_RAPID_ORI[n] (Konventioneller Eilgang für ORI-Achsen)
- MD21155 \$MC_JOG_VELO_ORI[n] (Konventionelle ORI-Achsgeschwindigkeit)
- MD21160 \$MC_JOG_VELO_RAPID_GEO[n] (Konventioneller Eilgang für GEO-Achsen)
- MD21165 \$MC_JOG_VELO_GEO[n] (Konventionelle GEO-Achsgeschwindigkeit)

Beschleunigung

Die Beschleunigung für Orientierungsachsen wird eingestellt im Maschinendatum:

MD21170 \$MC_ACCEL_ORI[n] (Beschleunigung für Orientierungsachsen)

2.10.2 Programmierung bei Orientierungstransformation

Die Programmierung ist nur in Verbindung mit einer Orientierungstransformation erlaubt.

Programmierung der Orientierung

Die Programmierung der Orientierungsachsen erfolgt über die Achsnamen **A2**, **B2** und **C2**.

Die Unterscheidung Euler-/RPY-Winkel erfolgt durch die G-Gruppe 50:

- ORIEULER:
Orientierungsprogrammierung über Euler-Winkel (Standard)
- ORIRPY:
Orientierungsprogrammierung über RPY-Winkel
- ORIVIRT1:
Orientierungsprogrammierung über virtuelle Orientierungsachsen (Definition 1)
- ORIVIRT2:
Orientierungsprogrammierung über virtuelle Orientierungsachsen (Definition 2)

Die Unterscheidung der Art der Interpolation erfolgt durch die G-Gruppe 51:

- ORIAXES:
Orientierungsprogrammierung der Linearen Interpolation von Orientierungsachsen oder Maschinenachsen
- ORIVECT:
Orientierungsprogrammierung der Großkreisinterpolation von Orientierungsachsen (Interpolation des Orientierungsvektors)

Mit Maschinendatum MD21102 \$MC_ORI_DEF_WITH_G_CODE (Def. der ORI-Achsen über G-Befehl) wird festgelegt, ob das MD21100 \$MC_ORIENTATION_IS_EULER (Winkeldefinition bei Orientierungsprogrammierung) wirksam ist (Standard) oder die G-Gruppe 50.

Folgende vier Varianten stehen für die Programmierung der Orientierung zur Verfügung:

1. A, B, C:
Angabe der Maschinenachseposition
2. A2, B2, C2:
Winkelprogrammierung virtueller Achsen
3. A3, B3, C3:
Angabe der Vektorkomponenten
4. LEAD, TILT:
Angabe von Voreil- und Seitwärtswinkel bezogen auf Bahn und Oberfläche

Literatur:

Programmierhandbuch Grundlagen

Hinweis

Die vier Varianten der Orientierungsprogrammierung schließen sich gegenseitig aus. Wenn gemischte Werte programmiert werden, wird Alarm 14130 oder Alarm 14131 ausgegeben.

Ausnahme:

Bei 6-Achs-Kinematiken mit einem 3. Freiheitsgrad für die Orientierung ist es zulässig, bei Variante 3 und 4 zusätzlich C2 zu programmieren. Mit C2 wird in diesem Fall die Verdrehung des Orientierungsvektors um seine Achse beschrieben.

Beispiel

Von Orientierungsachsen für eine Kinematik mit 6 bzw. 5 transformierten Achsen siehe Kapitel "Beispiel für Orientierungsachsen (Seite 142)".

Interpolationsart

Über das folgende Maschinendatum wird festgelegt, welche Interpolationsart verwendet wird:

MD21104 \$MC_ORI_IPO_WITH_G_CODE (G-Befehl für Orientierungsinterpolation):

- ORIMKS oder ORIWKS (Beschreibung siehe Kapitel "Werkzeugorientierung (Seite 51)")
- G-Gruppe 51 mit den Befehlen `ORIXES` bzw. `ORIVECT`
 - `ORIXES`:
Lineare Interpolation der Maschinenachsen oder Orientierungsachsen.
 - `ORIVECT`:
Die Orientierungsführung erfolgt durch Schwenken des Orientierungsvektors in der durch den Start- und Zielvektor aufgespannten Ebene (Großkreisinterpolation). Bei 6 transformierten Achsen erfolgt zusätzlich zur Schwenkbewegung eine Drehung um den Orientierungsvektor.
Bei `ORIVECT` werden die Orientierungsachsen immer auf kürzestem Weg verfahren.

Wertebereich

Wertebereich für Orientierungsachsen:

- 180 Grad < A2 < 180 Grad
- 90 Grad < B2 < 90 Grad
- 180 Grad < C2 < 180 Grad

Mit diesem Wertebereich sind alle möglichen Drehungen darstellbar. Werte außerhalb dieses Bereiches werden von der Steuerung in den oben genannten Bereich normiert.

Vorschub bei der Programmierung von ORIAXES

Der Vorschub für eine Orientierungsachse kann über die FL[]-Anweisung begrenzt werden (feed limit).

2.10.3 Programmierbarer Offset für Orientierungsachsen

Wirkungsweise des programmierbaren Offsets

Der zusätzlich programmierbare Offset für Orientierungsachsen wirkt additiv zum bereits bestehenden Offset und wird bei Aktivierung der Transformation festgelegt. Danach kann dieser additive Offset nicht mehr verändert werden und bewirkt auch keine Nullpunktverschiebung der Orientierungsachsen bei einer Orientierungstransformation.

Es bestehen zwei Möglichkeiten den programmierbaren Offsets festzulegen.

1. Der Offset wird bei der Aktivierung der Transformation mit `TRAORI ()` direkt programmiert.
2. Der Offset wird automatisch aus der bei Aktivierung der Transformation aktiven Nullpunktverschiebung für die Orientierungsachsen übernommen. Diese automatische Übernahme wird über Maschinendaten projiziert.

Offset direkt programmieren

Bei der Aktivierung der Transformation kann der Offset direkt in der Form `TRAORI (n, x, y, z, a, b)` programmiert werden. Folgende Parameter stehen optional zur Verfügung:

- n: Nummer der Transformation n = 1 oder 2
- x, y, z: Komponenten des Vektors der Grundorientierung des Werkzeugs (nur bei generischer 5-Achs Transformation).
- a, b: Offset für die Rundachsen

Diese optionalen Parameter können weggelassen werden. Im Falle einer Programmierung muss jedoch die richtige Reihenfolge eingehalten werden. Soll z. B. nur ein Offset der Rundachsen eingegeben werden, so wird z. B. `TRAORI (, , , , a, b)` programmiert.

Weitere Informationen zur Programmierung entnehmen Sie bitte:

Literatur:

Programmierhandbuch Arbeitsvorbereitung; Transformationen

Offset automatisch programmieren

Da der Offset aus der momentan aktiven Nullpunktverschiebung der Orientierungsachsen automatisch übernommen wird, sind die Auswirkungen einer Nullpunktverschiebung für die Rundachsen mit und ohne aktiver Transformation immer gleich. Die automatische Übernahme des Offsets aus der Nullpunktverschiebung wird über die Maschinendaten MD24590 $\$MC_TRAFO5_ROT_OFFSET_FROM_FR_1 = TRUE$ (Offset der Trafo-Rundachsen aus NPV) für die erste bzw. Maschinendatum MD24690 $\$MC_TRAFO5_ROT_OFFSET_FROM_FR_2 = TRUE$ (Offset der Trafo-Rundachsen aus NPV) für die zweite Transformation im Kanal, ermöglicht.

Hinweis

Eine Nullpunktverschiebung der Orientierungsachsen, die bei aktiver Transformation programmiert wird, unterscheidet sich gegenüber der bisherigen nicht.

Ist die automatische Übernahme des Offsets aktiviert, und wird gleichzeitig ein Offset der Rundachsen programmiert, so wird der programmierte Offsetwert vorrangig wirksam.

Orientierbare Werkzeugträger mit additiven Offset

Beim orientierbaren Werkzeugträger kann der Offset der beiden Rundachsen mit den Systemvariablen $\$TC_CARR24$ und $\$TC_CARR25$ programmiert werden. Dieser Offset der Rundachsen kann aus der zum Zeitpunkt der Aktivierung des orientierbaren Werkzeugträgers wirksamen Nullpunktverschiebung automatisch übernommen werden.

Die automatische Übernahme des Offsets aus der Nullpunktverschiebung wird durch das folgende Maschinendatum ermöglicht:

MD21186 $\$MC_TOCARR_ROT_OFFSET_FROM_FR = TRUE$ (Offset der TOCARR-Rundachsen aus NPV)

Hinweis

Weitere Informationen zu orientierbare Werkzeugträger entnehmen Sie bitte:

Literatur:

Funktionshandbuch Grundmaschine; Werkzeugkorrektur (W1)

2.10.4 Orientierungstransformation und orientierbare Werkzeugträger

Hinweis

Orientierungstransformation und Orientierbare Werkzeugträger können kombiniert werden.

Die resultierende Orientierung des Werkzeugs ergibt sich durch Verkettung von Orientierungstransformation und den Orientierbarem Werkzeugträger.

2.10.5 Modulo-Anzeige von Orientierungsachsen

Funktion

Die Positionen von Orientierungsachsen können für die BKS- und WKS-Anzeige in einem einstellbaren Modulo-Bereich angezeigt werden. Dabei spielt es keine Rolle, ob die jeweiligen Maschinenachsen Linearachsen oder Rundachsen sind. D. h. auch bei der "normalen" generischen 5/6-Achs-Transformation kann diese Anzeigemöglichkeit aktiviert werden.

Voraussetzungen

- Es müssen Orientierungsachsen vorhanden sein. Dies ist dann der Fall, wenn eine Orientierungstransformation aktiv ist (z. B. generische 5/6-Achs Transformation).
- Bei OEM-Transformationen müssen zusätzlich folgende Maschinendaten gesetzt sein: MD24585 \$MC_TRAFO5_ORIAX_ASSIGN_TAB_1[0..2]

Parametrierung

Die Modulo-Anzeige von Orientierungsachsen wird folgendermaßen aktiviert:
MD21132 \$MC_ORI_DISP_IS_MODULO[0..2] = TRUE

Der Modulo-Bereich wird durch folgende Maschinendaten festgelegt:

- MD21134 \$MC_ORI_MODULO_RANGE[0..2]
(Größe des Modulo-Bereichs für Anzeige der Orientierungsachsen)
- MD21136 \$MC_ORI_MODULO_RANGE_START[0..2]
(Startposition des Modulo-Bereichs für Anzeige der Orientierungsachsen)

Bitte beachten Sie dabei Folgendes:

- Die Maschinendaten werden mit NEWCONF wirksam.
- Diese Maschinendaten haben keinen Einfluss, bzw. keine Auswirkungen auf:
 - die möglichen Achspositionen, die für diese Achsen programmiert werden können.
 - die Verfahrbewegungen dieser Achsen.
 - die Anzeige von MKS-Werten dieser Achsen

2.11 Orientierungsvektoren

2.11.1 Polynominterpolation von Orientierungsvektoren

Programmierung von Polynomen für Achsbewegungen

Bei Orientierungsänderungen mittels Rundachsinterpolation werden normalerweise die Rundachsen linear interpoliert. Es ist jedoch möglich auch hier in üblicher Weise Polynome für die Rundachsen zu programmieren. Damit lassen sich im Allgemeinen homogenere Achsbewegungen erreichen.

Hinweis

Weitere Informationen zur Programmierung der Polynom-Interpolation mit POLY und zur Interpolation von Orientierungsvektoren sind beschrieben in:

Literatur:

Programmierhandbuch; Arbeitsvorbereitung

Mit einem Satz mit POLY wird die Polynom-Interpolation interpoliert. Ob die programmierten Polynome dann auch als Polynom interpoliert werden, hängt davon ab, ob der G-Befehl POLY aktiv ist oder nicht:

- Der G-Befehl ist **nicht aktiv**: Es werden die programmierten Achsendpunkte linear verfahren.
- Der G-Befehl ist **aktiv**: Die programmierten Polynome werden auch als Polynom interpoliert.

MD10674

Durch das Maschinendatum MD10674 \$MN_PO_WITHOUT_POLY = **FALSE** (Polynomprogrammierung ohne G-Befehl POLY programmierbar) kann eingestellt werden, ob die folgende Programmierung möglich ist:

- PO[...] bzw. PO(...) ist nur dann möglich, falls POLY aktiv ist, **oder**
- PO[] bzw. PO() Polynome sind auch ohne aktiven G-Befehl POLY möglich.

Standardmäßig ist MD10674: PO_WITHOUT_POLY = FALSE eingestellt und mit MD10674 \$MN_PO_WITHOUT_POLY = **TRUE** ist die folgende Programmierung immer möglich:

- PO[...] = (...), unabhängig davon ob POLY aktiv ist oder nicht.

Orientierungspolynome können mit verschiedenen Interpolationsarten zusammen programmiert werden und sind im Kapitel "Programmierung von Orientierungspolynomen" beschrieben.

POLYPATH:

Zusätzlich zu der modalen G-Befehl POLY kann mit dem vordefinierten Unterprogramm POLYPATH(Argument) die Polynominterpolation für verschiedene Achsgruppen selektiv aktiviert werden. Zur Aktivierung der Polynominterpolation sind folgende Argumente zulässig

("AXES"):	Für alle Bahnachsen und Zusatzachsen
("VECT"):	Für Orientierungsachsen
("AXES", "VECT"):	Für Bahnachsen, Zusatzachsen und Orientierungsachsen
(ohne Argument):	deaktiviert die Polynominterpolation für alle Achsgruppen

Standardmäßig ist die Polynominterpolation für alle Achsgruppen aktiviert.

Programmierung von Orientierungsvektoren

In jedem Satz kann ein Orientierungsvektor programmiert werden. Werden Polynome für die Orientierung programmiert, liegt im Allgemeinen der interpolierte Orientierungsvektor nicht mehr in der Ebene zwischen Start- und Endvektor, sondern kann beliebig aus dieser Ebene gedreht werden.

Die Orientierungsvektoren können auf folgende Weise programmiert werden:

1. Programmierung der Rundachspositionen mit A, B und C bzw. mit den tatsächlichen Rundachsamen.
2. Programmierung in Eulerwinkel oder RPY-Winkel über A2, B2, C2
3. Programmierung des Richtungsvektors über A3, B3, C3.
4. Programmierung über Voreilwinkel `LEAD` und Seitwärtswinkel `TILT`.

Auswahl der Art der Interpolation

Die Interpolationsart der Orientierungsachsen wird mittels des G-Befehls der Gruppe 51 ausgewählt und ist unabhängig von der Art der Programmierung des Endvektors:

- `ORIXES`: Lineare Interpolation der Maschinenachsen bzw. mittels Polynomen bei aktivem `POLY` oder
- `ORIVECT`: Interpolation des Orientierungsvektor mittels Großkreisinterpolation

Bei aktivem `ORIXES` kann die Interpolation der Rundachsen auch mittels Polynomen wie Polynom-Interpolation von Achsen mit `POLY` erfolgen.

Ist dagegen `ORIVECT` aktiv, so wird eine "normale" Großkreisinterpolation durch lineare Interpolation des Winkels des Orientierungsvektors in der Ebene, die von Start- und Endvektor aufgespannt wird, durchgeführt.

Polynome für 2 Winkel

Durch zusätzliche Programmierung von Polynomen für 2 Winkel, die den Startvektor in den Endvektor überführen, können auch bei `ORIVECT` komplexere Orientierungsänderungen programmiert werden.

Beide Winkel PHI und PSI werden in Grad angegeben.

POLY	Einschalten der Polynominterpolation für alle Achsgruppen.
POLYPATH ()	Einschalten der Polynominterpolation für alle Achsgruppen. Mögliche Gruppen sind "AXES" und "VECT".
Die Koeffizienten a_n und b_n werden in Grad angegeben.	
PO[PHI]=(a_2, a_3, a_4, a_5)	Der Winkel PHI wird gemäß $\text{PHI}(u) = a_0 + a_1 \cdot u + a_2 \cdot u^2 + a_3 \cdot u^3 + a_4 \cdot u^4 + a_5 \cdot u^5$ interpoliert.
PO[PSI]=(b_2, b_3, b_4, b_5)	Der Winkel PSI wird gemäß $\text{PSI}(u) = b_0 + b_1 \cdot u + b_2 \cdot u^2 + b_3 \cdot u^3 + b_4 \cdot u^4 + b_5 \cdot u^5$ interpoliert.
PL	Länge des Parameterintervalls auf dem die Polynome definiert sind. Das Intervall beginnt immer bei 0. Theoretischer Wertebereich für PL: 0,0001 ... 99999,9999. Der PL-Wert gilt für den Satz, in dem er steht. Ist kein PL programmiert, wirkt PL = 1.

Drehung des Orientierungsvektors

Orientierungsänderungen bei `ORIVECT` sind unabhängig von der Art der Programmierung des Endvektors möglich. Es gelten folgende Fälle:

Beispiel 1: Es werden direkt die Komponenten des Endvektors programmiert.

N... POLY A3=a B3=b C3=c PO[PHI] = (a2, a3, a4, a5) PO[PSI] = (b2, b3, b4, b5)

Beispiel 2: Der Endvektor wird durch die Positionen der Rundachsen bestimmt.

N... POLY Aa Bb Cc PO[PHI] = (a2, a3, a4, a5) PO[PSI] = (b2, b3, b4, b5)

Der Winkel PHI beschreibt die Drehung des Orientierungsvektors in der Ebene zwischen Start- und Endvektor (Großkreisinterpolation, siehe folgendes Bild). Die Interpolation des Orientierungsvektors erfolgt dabei genauso wie im Beispiel 1.

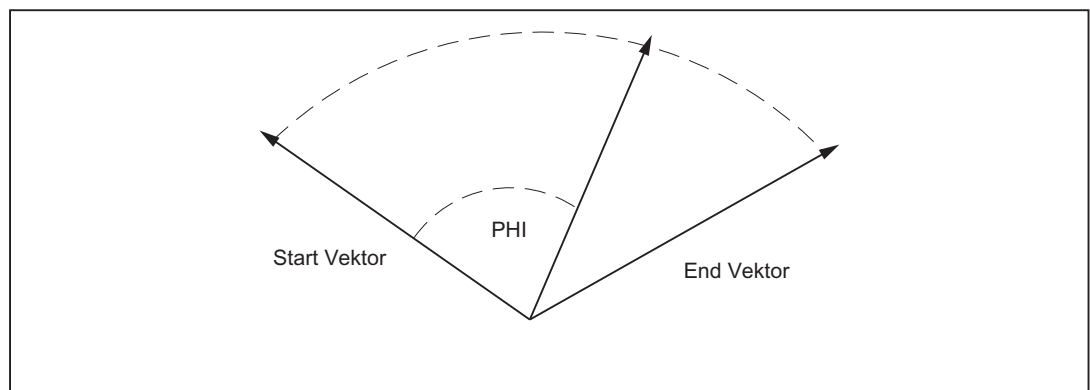


Bild 2-23 Drehung des Orientierungsvektors in der Ebene zwischen Start- u. Endvektor

Winkel PHI und PSI

Die Programmierung von Polynomen für die beiden Winkel $\text{PO}[\text{PHI}]$ und $\text{PO}[\text{PSI}]$ ist immer möglich. Ob die programmierten Polynome für PHI und PSI auch tatsächlich interpoliert werden, ist abhängig von:

- `POLYPATH("VECT")` und `ORIVECT` sind **aktiv**, so werden die Polynome interpoliert.
- `POLYPATH("VECT")` und `ORIVECT` sind **nicht aktiv**, so werden die programmierten Orientierungsvektoren am Satzende durch eine "normale" Großkreisinterpolation angefahren. Das heißt die Polynome für die beiden Winkel PHI und PSI werden in diesem Fall ignoriert.

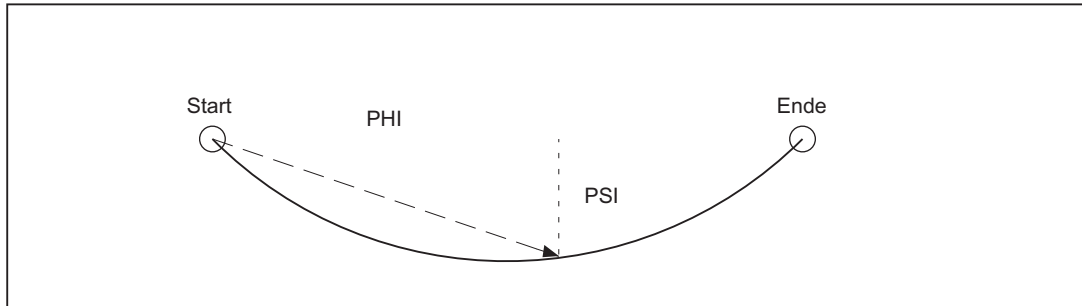


Bild 2-24 Bewegung des Orientierungsvektors in der Draufsicht

Mit Hilfe des Winkels PSI können Bewegungen des Orientierungsvektors senkrecht zur Großkreisinterpolations-Ebene erzeugt werden (siehe vorheriges Bild).

Maximal Polynome 5. Grades zulässig

Für die Winkel PHI und PSI können maximal Polynome 5. Grades programmiert werden. Dabei wird jeweils der konstante und lineare Koeffizient durch den Anfangswert oder Endwert des Orientierungsvektors festgelegt.

In der Liste der Koeffizienten (... , ...) können Koeffizienten höherer Ordnung weggelassen werden, falls diese alle gleich Null sind.

Mit PL kann zusätzlich die Länge des Parameterintervalls programmiert werden, auf dem die Polynome definiert sind.

Besonderheiten

Wird für den Winkel **PSI kein Polynom** programmiert, verläuft die Interpolation des Orientierungsvektors immer in der Ebene, die vom Start- und Endvektor aufgespannt wird.

Der Winkel PHI in dieser Ebene wird gemäß dem programmierten Polynom für PHI interpoliert. Damit wird vorrangig erreicht, dass der Orientierungsvektor durch eine "normale" Großkreisinterpolation in der Ebene zwischen Start- und Endvektor bewegt wird und dabei aber abhängig von dem programmierten Polynom eine mehr oder weniger ungleichförmige Bewegung durchführt.

Auf diese Weise ist der Geschwindigkeits- und Beschleunigungsverlauf der Orientierungsachsen innerhalb eines Satzes zum Beispiel beeinflussbar.

Hinweis

Weitere Informationen zur Polynominterpolation für Achsbewegungen und zur allgemeinen Programmierung von Polynomen sind beschrieben in:

Literatur:

Programmierhandbuch; Arbeitsvorbereitung

Randbedingungen

Die Polynominterpolation von Orientierungsvektoren ist nur bei Steuerungsvarianten möglich, zu deren Funktionsumfang folgende Funktionen gehören:

- Orientierungstransformation
- Polynominterpolation

2.11.2 Drehungen des Orientierungsvektors**Funktionalität**

Änderungen der Orientierung des Werkzeugs werden programmiert, indem in jedem Satz ein Orientierungsvektor angegeben wird, der am Satzende erreicht werden soll. Die Programmierung der Endorientierung jedes Satzes kann folgendermaßen durchgeführt werden:

1. direkte Programmierung des Vektors oder
2. Programmierung der Rundachspositionen.

Die zweite Möglichkeit ist von der Kinematik der Maschine abhängig. Zusätzlich kann noch durch Programmierung von Polynomen die Interpolation des Orientierungsvektors zwischen Start- und Endwert modifiziert werden.

Programmierung von Orientierungsrichtung

Zur Programmierung der Werkzeugeorientierung gibt es folgende Möglichkeiten:

1. Direkte Programmierung der Rundachspositionen (der Orientierungsvektor folgt aus der Kinematik der Maschine).
2. Programmierung in Eulerwinkel über A2, B2, C2 (der Winkel C2 hat keine Bedeutung).
3. Programmierung in RPY-Winkel über A2, B2, C2.
4. Programmierung des Richtungsvektors über A3, B3, C3 (die Länge des Vektors ist ohne Bedeutung).

Die Umschaltung zwischen Euler- und RPY-Winkel Programmierung erfolgt über das folgende Maschinendatum oder kann über die G-Befehle `ORIEULER` und `ORIRPY` ausgewählt werden:

MD21100 \$MC_ORIENTATION_IS_EULER (Winkeldefinition bei Orientierungsprogrammierung)

Programmierung von Orientierungsrichtung und Drehung

Während bei der Programmierung der Orientierung mittels RPY-Winkel die Drehrichtung bereits festgelegt wird, sind bei den anderen Orientierungen zusätzliche Angaben zur Festlegung der Drehrichtung notwendig:

1. Direkte Programmierung der Rundachspositionen
Eine zusätzliche Rundachse für die Drehrichtung muss festgelegt werden.
2. Programmierung in Eulerwinkel über A2, B2, C2
Es muss der Winkel C2 zusätzlich programmiert werden. Damit wird die vollständige Orientierung einschließlich Drehung des Werkzeugs festgelegt.
3. Programmierung in RPY-Winkel über A2, B2, C2
Zusätzliche Angaben sind nicht erforderlich.
4. Programmierung des Richtungsvektors über A3, B3, C3
Der Drehwinkel wird mittels `THETA=<Wert>` programmiert.

Hinweis

Die folgenden Fälle berücksichtigen eine programmierte Drehung nicht:

Mehrfache Programmierung der Drehrichtung sind nicht erlaubt und führen zu einem Alarm. So wird bei gleichzeitiger Programmierung des Eulerwinkels C2 und des Drehwinkels `THETA` die programmierte Drehung nicht ausgeführt.

Falls durch die Kinematik der Maschine bedingt, das Werkzeug nicht gedreht werden kann, wird eine eventuell programmierte Drehung ignoriert. Dies ist zum Beispiel bei einer normalen 5-Achs Werkzeugmaschine der Fall.

Drehung des Orientierungsvektors

Durch direkte Programmierung des Vektors bestehen folgende Möglichkeiten eine Drehung des Orientierungsvektors zu interpolieren:

- Linear interpoliert, d. h. der Winkel zwischen dem aktuellen Drehvektor und dem Startvektor ist eine lineare Funktion des Bahnparameters.
- Nicht linear durch die zusätzliche Programmierung eines Polynoms für den Drehwinkel q von maximal 5. Grades in der Form:
$$PO[THT] = (d_2, d_3, d_4, d_5)$$

Interpolation des Drehwinkels

In der Liste der Koeffizienten (... ,) können Koeffizienten höherer Ordnung weggelassen werden, falls diese alle gleich Null sind.

Der Endwert des Winkels sowie der konstante und lineare Koeffizient d_n des Polynoms können hierbei nicht direkt programmiert werden.

Der lineare Koeffizient d_n wird durch den Endwinkel q_e festgelegt und in Grad angegeben.

Der Endwinkel q_e bestimmt sich aus der Programmierung des Drehvektors.

Der Startwinkel q_s bestimmt sich aus dem Startwert des Drehvektors, der sich aus den Endwert des vorhergehenden Satzes ergibt. Durch den Startwinkel wird der konstante Koeffizient des Polynoms festgelegt.

Der Drehvektor ist immer senkrecht zur aktuellen Orientierung des Werkzeugs und bildet mit dem Basis-Drehvektor den Winkel THETA .

Hinweis

Bei der Maschinenkonfiguration kann festgelegt werden, in welche Raumrichtung der Drehvektor bei einem bestimmten Drehwinkel zeigt, wenn das Werkzeug die Grundorientierung einnimmt.

Formel

Allgemein wird der Drehwinkel mit einem Polynom 5. Grades interpoliert:

$$\theta u = \theta_s + d_1 u + d_2 u^2 + d_3 u^3 + d_4 u^4 + d_5 u^5 \quad (14)$$

Für das Parameterintervall $0 \dots 1$ ergibt sich daraus für den linearen Koeffizienten:

$$d_1 = \theta_e - \theta_s - d_2 - d_3 - d_4 - d_5 \quad (15)$$

Interpolation des Drehvektors

Der programmierte Drehvektor kann mit den modal wirkenden G-Befehlen auf folgende Art interpoliert werden:

- **ORIROTA (orientation rotation absolute):**
Der Drehwinkel THETA wird bezüglich einer absolut festgelegten Richtung im Raum interpretiert. Die Festlegung der Grunddrehrichtung erfolgt mittels Maschinendaten.
- **ORIROTR (orientation rotation relative):**
Der Drehwinkel THETA wird relativ zur Ebene, die von Start- und Endorientierung aufgespannt wird, interpretiert.
- **ORIROTT (orientation rotation tangential):**
Der Drehwinkel THETA wird relativ zur Orientierungsänderung interpretiert. Das heißt der Drehvektor wird für $\text{THETA}=0$ tangential zur Orientierungsänderung interpoliert. Dies gibt nur dann einen Unterschied zu **ORIROTR**, falls die Orientierungsänderung nicht in einer Ebene abläuft. Dies ist der Fall, wenn für die Orientierung mindestens ein Polynom für den "Kippwinkel" PSI programmiert wurde. Durch einen zusätzlich programmierten Drehwinkel THETA kann dann z. B. der Drehvektor so interpoliert werden, dass er immer einen bestimmten Winkel zur Orientierungsänderung bildet.

Aktivierung der Drehung

Eine Drehung des Orientierungsvektors wird mit dem Bezeichner `THETA` programmiert. Dabei gibt es folgende Möglichkeiten der Programmierung:

<code>THETA=<Wert></code>	Programmierung eines Drehwinkels, der am Satzende erreicht wird.
<code>THETA = q_e</code>	Programmierte Winkel q_e kann sowohl absolut (G90 ist aktiv) als auch relativ (G91 ist aktiv Kettenmaß) interpretiert werden.
<code>THETA = AC (...)</code>	Satzweise auf Maßangabe absolut umschalten
<code>THETA = IC (...)</code>	Satzweise auf Kettenmaßangabe umschalten
<code>PO[TH] = (...)</code>	Programmierung eines Polynoms für den Drehwinkel <code>THETA</code> .

Die Programmierung des Winkels `THETA` erfolgt in Grad.

Mit den modalen G-Befehlen wird die Interpolation des Drehvektors festgelegt:

<code>ORIROTA</code>	Drehwinkel zu einer absolut vorgegebenen Drehrichtung
<code>ORIROTR</code>	Drehwinkel relativ zur Ebene zwischen Start- und Endorientierung
<code>ORIROTT</code>	Drehwinkel relativ zur Änderung des Orientierungsvektors tangentialer Drehvektor zur Orientierungsänderung
<code>ORIROTC</code>	Drehwinkel relativ zur Änderung des Orientierungsvektors tangentialer Drehvektor zur Bahntangente
<code>PL</code>	Länge des Parameterintervalls auf dem die Polynome definiert sind. Das Intervall beginnt immer bei 0. Ist kein PL programmiert, wirkt PL = 1.

Diese G-Befehle legen die Bezugsrichtung des Drehwinkels fest. Damit ändert sich auch die Bedeutung des programmierten Drehwinkels.

Randbedingungen

Nur wenn die Interpolationsart `ORIROTA` aktiv ist, kann der Drehwinkel bzw. Drehvektor auf alle vier Arten programmiert werden.

1. Rundachspositionen
2. Eulerwinkel über A_2 , B_2 , C_2
3. RPY-Winkel über A_2 , B_2 , C_2
4. Richtungsvektor über A_3 , B_3 , C_3

Falls `ORIROTR` oder `ORIROTT` aktiv sind, kann der Drehwinkel nur noch direkt mit `THETA` programmiert werden.

Die anderen Möglichkeiten der Programmierung müssen hier ausgeschlossen werden, da damit eine absolute Drehrichtung festgelegt wird, die im Widerspruch zur Interpretation des Drehwinkels in diesen Fällen steht. Die möglichen Kombinationen zur Programmierung werden überwacht, und gegebenenfalls ein Alarm ausgegeben.

Eine Drehung kann auch allein in einem Satz programmiert werden, ohne dass eine Orientierungsänderung stattfindet. Dabei haben `ORIROTR` und `ORIROTT` keine Bedeutung. In diesem Fall wird der Drehwinkel immer in Bezug zur absoluten Richtung interpretiert (`ORIROTA`).

Eine programmierbare Drehung des Orientierungsvektors ist nur möglich, wenn eine Orientierungstransformation (`TRAORI`) aktiv ist.

Nur für Maschinenkinematiken, die eine Drehung der Werkzeugorientierung ermöglichen (z. B. 6-Achs-Maschinen), wird eine programmierte Drehung der Orientierung tatsächlich interpoliert.

2.11.3 Erweiterte Interpolationen von Orientierungen

Funktionalität

Zur Ausführung von Orientierungsänderungen entlang sich einer im Raum befindlichen Kegelmantelfläche, ist eine erweiterte Interpolation des Orientierungsvektors erforderlich. Dabei muss der Vektor bekannt sein, um den die Werkzeugorientierung gedreht werden soll. Außerdem müssen die Start- und Endorientierung vorgegeben werden. Die Startorientierung ist durch den vorhergehenden Satz gegeben und die Endorientierung muss entweder programmiert werden, oder durch andere Bedingungen festgelegt werden.

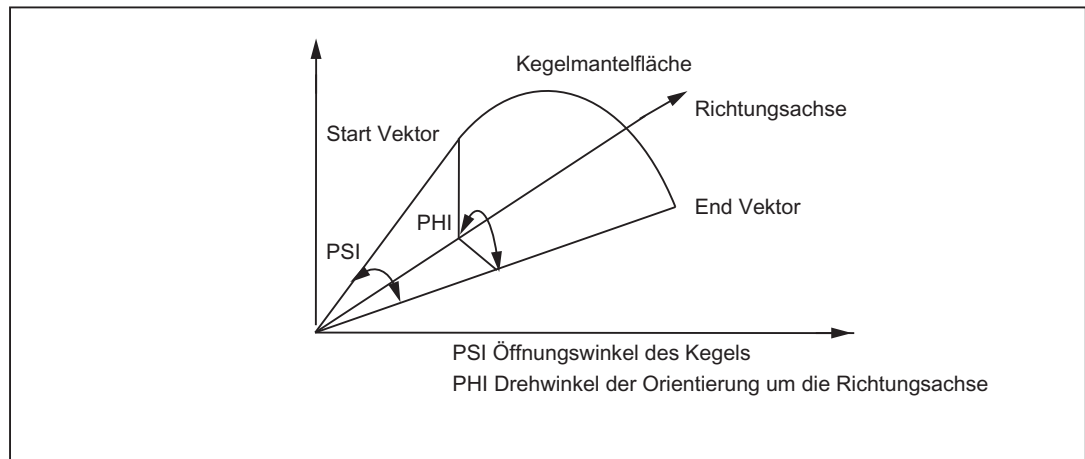


Bild 2-25 Orientierungsänderung auf einer im Raum befindlichen Kegelmantelfläche

Notwendige Festlegungen

Generell sind folgende Angaben notwendig:

- Die **Startorientierung** wird durch die Endorientierung des vorhergehenden Satzes festgelegt.
- Die **Endorientierung** wird entweder durch Angabe des Vektors (mit A3, B3, C3), der Euler- bzw. RPY-Winkel (mit A2, B2, C2) oder durch Programmierung der Rundachspositionen (mit A, B, C) festgelegt.
- Die **Drehachse des Kegels** wird als (normierter) Vektor mit A6, B6, C6 programmiert.

- Der **Öffnungswinkel des Kegels** wird mit dem Bezeichner NUT (**nut**ation angle) in Grad programmiert.
Der **Wertebereich** dieses Winkels ist auf das Intervall 0 Grad bis 180 Grad beschränkt. Dabei dürfen die Werte 0 Grad und 180 Grad nicht programmiert werden. Wird ein Winkel außerhalb des gültigen Intervalls programmiert erscheint ein Alarm.
Für den Spezialfall NUT = 90 Grad wird der Orientierungsvektor in einer Ebene senkrecht zum Richtungsvektor interpoliert (Großkreisinterpolation).
Mit dem Vorzeichen des programmierten Öffnungswinkel wird festgelegt, ob der Verfahrwinkel größer oder kleiner als 180 Grad sein soll.
Zur Festlegung des Kegels muss entweder dessen **Richtungsvektor** oder dessen **Öffnungswinkel** programmiert werden. Beide Angaben gleichzeitig sind nicht erlaubt.
- Eine weitere Möglichkeit besteht darin, dass eine **Zwischenorientierung**, die zwischen Start- und Endorientierung liegt programmiert wird.

Programmierung

ORIPLANE	orientation interpolation in a plane: Interpolation in einer Ebene (Großkreisinterpolation)
ORICONCW	orientation interpolation on a cone clockwise: Interpolation auf einer Kegelmantelfläche im Uhrzeigersinn
ORICONCCW	orientation interpolation on a cone counter clockwise: Interpolation auf einer Kegelmantelfläche gegen den Uhrzeigersinn.

Die Programmierung des **Richtungsvektors** erfolgt mit den Bezeichnern A6, B6, C6 und wird als (normierter) Vektor angegeben.

Hinweis

Die Programmierung einer Endorientierung ist hierbei **nicht unbedingt** erforderlich. Ist keine Endorientierung angegeben, dann wird ein voller Kegelmantel mit 360 Grad interpoliert.

Die Programmierung des **Öffnungswinkel des Kegels** erfolgt mit NUT= <winkel>, wobei die Angabe des Winkels in Grad erfolgt.

Hinweis

Hierbei ist die Angabe einer Endorientierung **zwingend** erforderlich. Ein vollständiger Kegelmantel mit 360 Grad kann auf diese Weise nicht interpoliert werden. Mit dem Vorzeichen des Öffnungswinkels wird festgelegt, ob der Verfahrwinkel größer oder kleiner als 180 Grad sein soll.

Dabei bedeuten:

- | | |
|------------|--|
| NUT = +... | Verfahrwinkel kleiner oder gleich 180 Grad |
| NUT = -... | Verfahrwinkel größer oder gleich 180 Grad |

Ein positives Vorzeichen kann bei der Programmierung weggelassen werden.

Angaben bei Zwischenorientierung

ORICONIO

orientation interpolation on a cone with intermediate orientation: Interpolation auf einer Kegelmantelfläche mit Angabe einer Zwischenorientierung

Ist dieser G-Befehl aktiv, dann ist die Angabe einer **Zwischenorientierung** mit A7, B7, C7 erforderlich, und wird als (normierter) Vektor angegeben.

Hinweis

Die Programmierung der Endorientierung ist hierbei **zwingend** erforderlich.

Die **Orientierungsänderung** und die **Drehrichtung** wird eindeutig durch die drei Vektoren Start-, End- und Zwischenorientierung festgelegt.

Alle drei Vektoren müssen hierbei voneinander unterschiedlich sein. Falls die programmierte Zwischenorientierung parallel zur Start- oder Endorientierung ist, wird eine lineare Großkreisinterpolation der Orientierung in der Ebene, die von Start- und Endvektor aufgespannt wird, durchgeführt.

Drehwinkel und Öffnungswinkel

Zusätzlich können hierbei für die beiden Winkel des Kegels programmiert werden:

PHI

Drehwinkel der Orientierung um die Richtungsachse

PSI

Öffnungswinkel des Kegels

Außerdem können Polynome von maximal 5. Grades wie folgt programmiert werden:

PO[PHI] = (a2, a3, a4, a5) Die konstanten und linearen Koeffizienten werden jeweils durch
PO[PSI] = (b2, b3, b4, b5) die Start- bzw. Endorientierung bestimmt.

Weitere Interpolationsmöglichkeiten

Es besteht die Möglichkeit, die Orientierung auf einem Kegel zu interpolieren, der tangential an die vorhergehende Orientierungsänderung anschließt. Diese Orientierungsinterpolation wird durch Programmierung des G-Befehls `ORICONTO` erreicht.

ORICONTO

orientation interpolation on a cone with tangential orientation: Interpolation auf einer Kegelmantelfläche mit tangenalem Übergang

Eine weitere Möglichkeit der Orientierungsinterpolation besteht darin, dass die Orientierungsänderung durch die Bahn eines 2. Kontaktpunktes auf dem Werkzeug beschrieben wird.

ORICURVE

orientation interpolation with a second curve: Interpolation der Orientierung mit Vorgabe der Bewegung zweier Kontaktpunkte des Werkzeugs.

Hierbei ist die Angabe der Koordinaten der Bewegung des 2. Kontaktpunktes des Werkzeugs notwendig. Diese zusätzliche Raumkurve wird programmiert mit XH, YH, ZH.

Außer den jeweiligen Endwerten können auch zusätzliche Polynome in der folgenden Form programmiert werden:

PO[XH] = (xe, x2, x3, x4, x5): (xe, ye, ze) der Endpunkt der Kurve, und

PO[YH] = (ye, y2, y3, y4, y5): xi, yi, zi sind die Koeffizienten der Polynome

PO[ZH] = (ze, z2, z3, z4, z5): von maximal 5. Grades.

Bei dieser Interpolationsart können für die beiden Raumkurven Punkte (G1) bzw. Polynome (POLY) programmiert werden.

Hinweis

Es sind insbesondere keine Kreise und Evolventen zulässig. Zusätzlich kann aber eine Splineinterpolation mit BSPLINE aktiviert werden. Dabei werden die programmierten Endpunkte beider Raumkurven als Knotenpunkte interpretiert.

Die anderen Splinearten (ASPLINE und CSPLINE) sowie die Aktivierung eines Kompressors (COMPON, COMPCURV, COMPCAD) sind hierbei nicht erlaubt.

Randbedingungen

Die erweiterte Interpolation von Orientierungen erfordert die Beachtung aller notwendigen Orientierungstransformationen, da diese zum Funktionsumfang gehören.

Aktivierung

Die Orientierungsänderungen auf einer beliebig im Raum befindlichen Kegelmantelfläche wird mit den G-Befehlen der Gruppe 51 durch erweiterte Interpolation des Orientierungsvektors mit folgenden Befehlen aktiviert:

ORIPLANE	Interpolation in einer Ebene, Angabe der Endorientierung (entspricht ORIVECT)
ORICONCW	Interpolation auf einer Kegelmantelfläche im Uhrzeigersinn Angabe der Endorientierung und der Kegelrichtung bzw. Öffnungswinkel des Kegels.
ORICONCCW	Interpolation auf einer Kegelmantelfläche gegen den Uhrzeigersinn. Angabe der Endorientierung und der Kegelrichtung bzw. Öffnungswinkel des Kegels.
ORICONIO	Interpolation auf einer Kegelmantelfläche mit Angabe der Endorientierung und einer Zwischenorientierung.
ORICONTO	Interpolation auf einer Kegelmantelfläche mit tangentialem Übergang, Angabe der Endorientierung.
ORICURVE	Interpolation der Orientierung mit Vorgabe der Bewegung zweier Kontaktpunkte des Werkzeugs.

ORIPATH	Werkzeugorientierung bezogen auf die Bahn.
ORIPATHS	Werkzeugorientierung bezogen auf die Bahn, wenn z. B. ein nick im Orientierungsverlauf z. B. an einer Ecke der Kontur geglättet werden soll (siehe Kapitel "Bahnrelative Orientierung (ORIPATH, ORIPATHS, ORIROT) (Seite 101)").

Beispiele

Im folgenden Programmbeispiel werden unterschiedliche Orientierungsänderungen programmiert:

Programmcode	Kommentar
...	
N10 G1 X0 Y0 F5000	
N20 TRAORI	; Orientierungstransformation aktiviert.
N30 ORIVECT	; WZ-Orientierung als Vektor interpolieren
N40 ORIPLANE	; Großkreisinterpolation auswählen
N50 A3=0 B3=0 C3=1	
N60 A3=0 B3=1 C3=1	; Orientierung in der Y/Z-Ebene um 45 ; Grad gedreht, am Satzende wird die ; Orientierung (0, 1 / (Wurzel aus 2), ; 1 / (Wurzel aus 2) erreicht.
N70 ORICONCW	; der Orientierungsvektor wird auf einem ; Kegelmantel mit der Richtung
N80 A6=0 B6=0 C6=1 A3=1 B3=0 C3=1;	(0, 0, 1) bis zur Orientierung ; (1 / (Wurzel aus 2), 0, 1 / (Wurzel aus 2) ; im Uhrzeigersinn interpoliert, ; der Drehwinkel beträgt hierbei 270 Grad. ; Die WZ-Orientierung durchläuft eine volle
N90 A6=0 B6=0 C6=1	; Umdrehung auf demselben Kegelmantel.
...	

2.12 Online-Werkzeuflängenkorrektur

Funktionalität

Mit der Online-Werkzeuflängenkorrektur können die effektiven Werkzeuflängen in Echtzeit so verändert werden, dass diese Längenänderungen auch bei Orientierungsänderungen des Werkzeugs berücksichtigt werden. Über die Systemvariable \$AA_TOFF[<Geometrieachsname>] werden Werkzeuflängenkorrekturen 3-dimensional, entsprechend den drei Werkzeugrichtungen eingerechnet.

Grundsätzlich werden keine Werkzeugparameter verändert. Die eigentliche Korrektur erfolgt intern über Transformationen mittels einer orientierbaren Werkzeuflängenkorrektur.

Es können nur so viele Korrekturrichtungen aktiv sein, wie Geometrieachsen aktiv sind. Alle Korrekturen können gleichzeitig aktiv sein.

Anwendung

Die Funktion Online-Werkzeuglängen-Korrektur ist anwendbar bei:

- Orientierungstransformationen (TRAORI)
 - Orientierbarer Werkzeugträger (TCARR)
-

Hinweis

Die Online-Werkzeuglängenkorrektur ist eine Option. Nur in Verbindung mit einer aktiven Orientierungstransformation oder einem aktiven Orientierbaren Werkzeugträger ist diese Funktion sinnvoll.

Literatur:

Funktionshandbuch Grundfunktionen; Werkzeugkorrektur, Kapitel: Orientierbare Werkzeugträger (W1)

Satzaufbereitung

Bei der Satzaufbereitung im Vorlauf wird der aktuelle im Hauptlauf wirksame Werkzeuglängenoffset mit berücksichtigt. Um die maximal zulässigen Achsgeschwindigkeiten weitestgehend auszunutzen, ist es erforderlich, die Satzaufbereitung mit einem Vorlaufstopp (STOPRE) anzuhalten, während ein Werkzeugoffset aufgebaut wird.

Der Werkzeugoffset ist zum Vorlaufzeitpunkt auch immer dann bekannt, wenn die Werkzeuglängenkorrekturen nach Programmstart nicht mehr verändert werden, oder wenn nach einer Veränderung der Werkzeuglängenkorrekturen mehr Sätze abgearbeitet wurden als der IPO-Buffer zwischen Vorlauf und Hauptlauf aufnehmen kann. Damit wird bei kurzen Sätzen die korrekte Berücksichtigung der Achsgeschwindigkeiten schnell wirksam.

Das Maß für die Differenz zwischen der aktuell im Interpolator wirksamen Korrektur und der Korrektur, die zum Zeitpunkt der Satzaufbereitung wirksam war, kann in der Systemvariable \$AA_TOFF_PREP_DIFF[] abgefragt werden.

Hinweis

Die Veränderung der effektiven Werkzeuglänge durch die Online-Werkzeuglängenkorrektur führt bei Orientierungsänderungen zu veränderten Ausgleichsbewegungen der an der Transformation beteiligten Achsen. Die resultierenden Geschwindigkeiten können dabei abhängig von der Maschinenkinematik und den aktuellen Achspositionen sowohl größer als auch kleiner werden.

MD21190 \$MC_TOFF_MODE (Wirkungsweise Korrektur in Werkzeugr.)

Über das folgende Maschinendatum kann eingestellt werden, ob der Inhalt der Synchronaktionsvariable \$AA_TOFF[] als absoluter Wert angefahren werden soll, oder ob ein integrierendes Verhalten erfolgen soll:

MD21190 \$MC_TOFF_MODE

Durch das integrierende Verhalten von \$AA_TOFF[] ist eine 3D-Abstandsregelung möglich. Der aufintegrierte Wert steht über die Systemvariable \$AA_TOFF_VAL[] zu Verfügung.

Zur Projektierung der Online-Werkzeuglängenkorrektur stehen folgende Maschinendaten und Settingdaten zur Verfügung:

Maschinendaten/Settingdatum	Bedeutung für Online-Werkzeuglängenkorrektur
MD21190 \$MC_TOFF_MODE	Der Inhalt von \$AA_TOFF[] wird als absoluter Wert herausgefahren oder aufintegriert
MD21194 \$MC_TOFF_VELO (Geschw. Online Korrektur in Werkzeugr.)	Geschwindigkeit der Online-WZL-Korrektur
MD21194 \$MC_TOFF_ACCEL (Beschl. Online Korrektur in Werkzeugr.)	Beschleunigung der Online-WZL-Korrektur
SD42970 \$SC_TOFF_LIMIT (Obergr. d. Korrekturw. \$AA_TOFF)	Obergrenze des Werkzeuglängenkorrekturwertes

Bei der Beschleunigungsreserve wird 20 Prozent für die überlagerte Bewegung der Online-Werkzeuglängenkorrektur vorgesehen, die über das folgende Maschinendatum veränderbar ist:

MD20610 \$MC_ADD_MOVE_ACCEL_RESERVE (Beschleunigungsreserve für überlagerte Bewegungen)

Aktivierung

Mit der Anweisung TOFFON kann die Online-Werkzeuglängenkorrektur für mindestens eine Werkzeugrichtung bei verfügbarer Option vom Teileprogramm aus aktiviert werden. Bei der Aktivierung kann für die entsprechende Korrekturrichtung ein Offsetwert angegeben werden, der sofort herausgefahren wird.

Beispiel: TOFFON(Z, 25).

Eine wiederholte Programmierung der Anweisung TOFFON() mit einem neuen Offsetwert führt dazu, dass der neue Offsetwert herausgefahren wird. Der Offsetwert wird hierbei als absoluter Wert zur Variablen \$AA_TOFF[] addiert.

Hinweis

Weitere Informationen zur Programmierung mit Beispielen entnehmen Sie bitte:

Literatur:

Programmierhandbuch Arbeitsvorbereitung; Transformationen

Solange die Online-Werkzeuglängenkorrektur aktiv ist, wird das VDI-Signal der Schnittstelle NC → PLC in folgendem Nahtstellensignal auf 1 gesetzt:

DB21, ... DBX318.2 (TOFF aktiv)

Während eine Korrekturbewegung aktiv ist, wird das VDI → Signal in folgendem Nahtstellensignal auf 1 gesetzt:

DB21, ... DBX318.3 (TOFF Bewegung aktiv)

Rücksetzen

Die Korrekturwerte können mit dem Befehl TOFFOF() rückgesetzt werden. Diese Anweisung löst einen Vorlaufstopp aus.

Die aufgebauten Werkzeuglängenkorrekturen werden abgelöscht und ins Basiskoordinatensystem übernommen. Der Vorlauf synchronisiert sich auf die aktuelle Position im Hauptlauf. Da hierbei keine Achsen verfahren werden, ändern sich die Werte von \$AA_IM[] nicht. Es werden nur die Werte der Variablen \$AA_IW[] und \$AA_IB[] verändert. Diese enthalten nun den abgewählten Anteil der Werkzeuglängenkorrektur.

Nach der Abwahl der "Online-Werkzeuglängenkorrektur" für eine Werkzeugrichtung ist der Wert der Systemvariablen \$AA_TOFF[] bzw. \$AA_TOFF_VAL[] für diese Werkzeugrichtung Null. Das folgende Nahtstellensignal wird auf 0 gesetzt:

DB21, ... DBX318.2 (TOFF aktiv)

Alarm 21670

Ein vorhandener Werkzeuglängenoffset muss vorher über TOFFOF() gelöscht werden, damit der Alarm 21670 "Kanal %1 Satz %2 unzulässige Änderung der Werkzeugrichtung wegen \$AA_TOFF aktiv" unterdrückt wird:

- Beim Abschalten der Transformation mit TRAFFOF
- Wenn von CP nach PTP-Fahren umgeschaltet wird.
- Wenn beim Geo-Achstausch ein Werkzeuglängenoffset in Richtung der Geometrieachse vorhanden ist.
- Wenn beim Ebenenwechsel ein Werkzeuglängenoffset vorhanden ist.
- Beim Wechsel vom achsspezifischen manuellen Verfahren in der Betriebsart JOG auf PTP solange eine Werkzeuglängenkorrektur aktiv ist. Der Wechsel nach PTP wird nicht ausgeführt.

Betriebsartenwechsel

Die Werkzeuglängenkorrektur bleibt auch beim Betriebsartenwechsel aktiv und kann in allen Betriebsartenausgeführt werden.

Wird beim Betriebsartenwechsel eine Werkzeuglängenkorrektur aufgrund von \$AA_TOFF[] interpoliert, kann die Betriebsartenumschaltung erst erfolgen, wenn die Interpolation der Werkzeuglängenkorrektur beendet ist. Es wird der Alarm 16907 "Kanal %1 Aktion %2 <ALNX> nur im Stopp-Zustand möglich" gemeldet.

Verhalten bei REF und Satzsuchlauf

Beim Anfahren des Referenzpunktes REF in der Betriebsart JOG wird die Werkzeuglängenkorrektur nicht berücksichtigt.

Die Anweisungen TOFFON() und TOFFOF() werden beim Satzsuchlauf nicht mit aufgesammelt und in einem Aktionssatz ausgegeben.

Systemvariable

Bei der Online-Werkzeuflängenkorrektur stehen dem Anwender folgende Systemvariable zur Verfügung:

Systemvariable	Bedeutung für Online-Werkzeuflängenkorrektur
\$AA_TOFF[]	Positionsoffset im Werkzeugkoordinatensystem
\$AA_TOFF_VAL[]	Integrierter Positionsoffset im WKS
\$AA_TOFF_LIMIT[]	Abfrage, ob sich der Werkzeuflängenkorrekturwert im Grenzbereich befindet
\$AA_TOFF_PREP_DIFF[]	Größe der Differenz zwischen dem aktuell wirksamen Wert von \$AA_TOFF[] und dem Wert, als der aktuelle Bewegungssatz präpariert wurde.

Literatur:

LHB Systemvariablen

Randbedingungen

Die Funktion Online-Werkzeuflängenkorrektur ist eine Option und steht standardmäßig bei der "Generischen 5-Achs-Transformation" und bei dem "Orientierbaren Werkzeugträger" zur Verfügung.

Steht das Werkzeug bei der Bearbeitung nicht senkrecht auf der Werkstückoberfläche, oder enthält die Kontur Krümmungen, deren Radius kleiner ist als das Korrekturmaß, ergeben sich Abweichungen von der eigentlichen Offsetfläche. Die Erzeugung exakter Offsetflächen ist mit einer Werkzeuflängenkorrektur allein nicht möglich.

2.13 Beispiele

2.13.1 Beispiel für eine 5-Achs-Transformation

```
CHANDATA(1)
```

```
$MA_IS_ROT_AX[AX5] = TRUE
$MA_SPIND_ASSIGN_TO_MACHAX[AX5] = 0
$MA_ROT_IS_MODULO[AX5] = 0
```

```
-----
; Allgemeine 5-Achs-Transformation
;
; Kinematik:
; 1. Rundachse ist parallel zu Z
; 2. Rundachse ist parallel zu X
```

Bewegliches Werkzeug

```
;  
-----  
;
```

\$MC_TRAFO_TYPE_1 = 20

\$MC_ORIENTATION_IS_EULER = TRUE

\$MC_TRAFO_AXES_IN_1[0] = 1

\$MC_TRAFO_AXES_IN_1[1] = 2

\$MC_TRAFO_AXES_IN_1[2] = 3

\$MC_TRAFO_AXES_IN_1[3] = 4

\$MC_TRAFO_AXES_IN_1[4] = 5

\$MC_TRAFO_GEOAX_ASSIGN_TAB_1[0]=1

\$MC_TRAFO_GEOAX_ASSIGN_TAB_1[1]=2

\$MC_TRAFO_GEOAX_ASSIGN_TAB_1[2]=3

\$MC_TRAFO5_PART_OFFSET_1[0] = 0

\$MC_TRAFO5_PART_OFFSET_1[1] = 0

\$MC_TRAFO5_PART_OFFSET_1[2] = 0

\$MC_TRAFO5_ROT_AX_OFFSET_1[0] = 0

\$MC_TRAFO5_ROT_AX_OFFSET_1[1] = 0

\$MC_TRAFO5_ROT_SIGN_IS_PLUS_1[0] = TRUE

\$MC_TRAFO5_ROT_SIGN_IS_PLUS_1[1] = TRUE

\$MC_TRAFO5_NON_POLE_LIMIT_1 = 2.0

\$MC_TRAFO5_POLE_LIMIT_1 = 2.0

\$MC_TRAFO5_BASE_TOOL_1[0] = 0.0

\$MC_TRAFO5_BASE_TOOL_1[1] = 0.0

\$MC_TRAFO5_BASE_TOOL_1[2] = 5.0

\$MC_TRAFO5_JOINT_OFFSET_1[0] = 0.0

\$MC_TRAFO5_JOINT_OFFSET_1[1] = 0.0

\$MC_TRAFO5_JOINT_OFFSET_1[2] = 0.0

CHANDATA(1)

M17

Beispielprogramm zur allgemeinen 5-Achs-Transformation:

Programmcode	Kommentar
; Definition des Werkzeugs T1	
\$TC_DP1[1,1] = 10	; Typ
\$TC_DP2[1,1] = 0	
\$TC_DP3[1,1] = ;z Längenkorrekturvektor G17	
\$TC_DP4[1,1] = 0.	; Y
\$TC_DP5[1,1] = 0.	; x
\$TC_DP6[1,1] = 0.	; Radius
\$TC_DP7[1,1] = 0	
\$TC_DP8[1,1] = 0	
\$TC_DP9[1,1] = 0	
\$TC_DP10[1,1] = 0	
\$TC_DP11[1,1] = 0	
\$TC_DP12[1,1] = 0	

Ausgangsstellung anfahren:

```
N100 G1 x1 y0 z0 a0 b0 F20000 G90 G64 T1 D1 G17 ADIS=.5 ADISPOS=3
```

Orientierungsvektorprogrammierung:

```
N110 TRAORI(1)
N120 ORIWKS
N130 G1 G90
N140 a3 = 0 b3 = 0 c3 = 1 x0
N150 a3 = 0 b3 = -1 c3 = 0
N160 a3 = 1 b3 = 0 c3 = 0
N170 a3 = 1 b3 = 0 c3 = 1
N180 a3 = 0 b3 = 1 c3 = 0
N190 a3 = 0 b3 = 0 c3 = 1
```

Eulerwinkelprogrammierung:

```
N200 ORIMKS
N210 G1 G90
N220 a2 = 0 b2 = 0 x0
N230 a2 = 0 b2 = 90
N240 a2 = 90 b2 = 90
N250 a2 = 90 b2 = 45
N260 a2 = 0 b2 = -90
N270 a2 = 0 b2 = 0
```

Achsprogrammierung:

```
N300 a0 b0 x0
N310 a45
N320 b30
```

TOFRAME:

```
N400 G0 a90 b90 x0 G90
N410 TOFRAME
N420 z5
N430 x3 y5
N440 G0 a0 b0 x1 y0 z0 G90

N500 TRAFOOF
m30
```

2.13.2 Beispiel für eine 3- und 4-Achs-Transformation

2.13.2.1 Beispiel für eine 3-Achs-Transformation

Beispiel: Für die schematisch dargestellte Maschine (siehe "Bild 2-1 Schematische Darstellung einer 3-Achs-Transformation (Seite 38)") kann die 3-Achs-Transformation folgendermaßen projiziert werden:

Programmcode	Kommentar
\$MC_TRAFO_TYPE_n = 18	
\$MC_TRAFO_GEOAX_ASSIGN_TAB_n[0] = 1	; Zuweisung der Kanalachsen zu Geo-Achsen
\$MC_TRAFO_GEOAX_ASSIGN_TAB_n[1] = 0	
\$MC_TRAFO_GEOAX_ASSIGN_TAB_n[2] = 3	
\$MC_TRAFO_AXES_IN_n[0] = 1	; x-Achse ist Kanalachse 1
\$MC_TRAFO_AXES_IN_n[1] = 0	; y-Achse wird nicht verwendet
\$MC_TRAFO_AXES_IN_n[2] = 3	; z-Achse ist Kanalachse 3
\$MC_TRAFO_AXES_IN_n[4] = 0	; es gibt keine 2. rotatorische Achse

2.13.2.2 Beispiel für eine 4-Achs-Transformation

Beispiel: Für die schematisch dargestellte Maschine (siehe "Bild 2-2 Schematische Darstellung einer 4-Achs-Transformation mit beweglichem Werkstück (Seite 38)"), jedoch mit einer zusätzlichen Achse (Y), kann die 4-Achs-Transformation folgendermaßen projiziert werden:

Programmcode	Kommentar
\$MC_TRAFO_TYPE_n = 18	
\$MC_TRAFO_GEOAX_ASSIGN_TAB_n[0] = 1	
\$MC_TRAFO_GEOAX_ASSIGN_TAB_n[1] = 2	
\$MC_TRAFO_GEOAX_ASSIGN_TAB_n[2] = 3	
\$MC_TRAFO_AXES_IN_n[0] = 1	; x-Achse ist Kanalachse 1
\$MC_TRAFO_AXES_IN_n[1] = 2	; y-Achse ist Kanalachse 2

Programmcode	Kommentar
\$MC_TRAFO_AXES_IN_n[2] = 3	; z-Achse ist Kanalachse 3
\$MC_TRAFO_AXES_IN_n[4] = 0	; es gibt keine 2. rotatorische Achse

2.13.3 Beispiel Kardanischer Fräskopf

Allgemeines

Die folgenden Ausschnitte geben die wesentlichen Aktionen wieder, die im Zusammenhang mit der Aktivierung einer Transformation für den kardanischen Fräskopf erforderlich sind.

Maschinendaten

; Maschinenkinematik CA' mit Orientierung des Werkzeugs in Nullstellung in Z-Richtung

\$MC_TRAFO_TYPE_1 = 148

\$MC_TRAFO_GEOAX_ASSIGN_TAB_1[0] = 1

\$MC_TRAFO_GEOAX_ASSIGN_TAB_1[1] = 2

\$MC_TRAFO_GEOAX_ASSIGN_TAB_1[2] = 3

;Winkel der 2. rotatorischen Achse

\$MC_TRAFO5_NUTATOR_AX_ANGLE_1 = 45

Programm

Programmcode	Kommentar
; Definition des Werkzeugs T1	
\$TC_DP1[1,1] = 120	; Typ
\$TC_DP2[1,1] = 0	;
\$TC_DP3[1,1] = 20	; Z Längenkorrekturvektor G17
\$TC_DP4[1,1] = 8.	; Y
\$TC_DP5[1,1] = 5.	; X
TRAORI(1)	; Aktivierung der Transformation
ORIMKS	; Bezug der Orientierung auf das MKS
G0 X1 Y0 Z0 A0 B0 F20000 G90 G64 T1 D1 G17	
;Programmierung des Richtungsvektors	
G1 G90	
a3 = 0 b3 = 1 c3 = 0	
;Programmierung in Eulerwinkeln	

2.13 Beispiele

Programmcode	Kommentar
G1 G90	
a2 = 0 b2 = 0 X0	
;Programmierung der Bewegung der Rundachsen	
G1 X10 Y5 Z20 A90 C90	
m30	

Literatur:
 Programmierhandbuch Grundlagen

2.13.4 Beispiel für Orientierungsachsen

Beispiel 1:

3 Orientierungsachsen für die 1. Orientierungstransformation für eine Kinematik mit 6 transformierten Achsen. Die Drehung soll in dieser Reihenfolge ablaufen:

- zuerst um die Z-Achse,
- danach um die Y-Achse und
- anschließend wieder um die Z-Achse.

Der Werkzeugvektor soll in X-Richtung zeigen.

Programmcode	Kommentar
CHANDATA (1)	
\$MC_TRAFO5_TOOL_VECTOR_1=0	; Werkzeugvektor in X-Richtung
\$MC_TRAFO5_ORIAX_ASSIGN_TAB_1[0]=4	; Kanalindex 1. Orient.achse
\$MC_TRAFO5_ORIAX_ASSIGN_TAB_1[1]=5	; Kanalindex 2. Orient.achse
\$MC_TRAFO5_ORIAX_ASSIGN_TAB_1[2]=6	; Kanalindex 3. Orient.achse
\$MC_ORIAX_TURN_TAB_1[0]=3	; Z-Richtung
\$MC_ORIAX_TURN_TAB_1[1]=2	; Y-Richtung
\$MC_ORIAX_TURN_TAB_1[2]=3	; Z-Richtung
CHANDATA (1)	
M17	

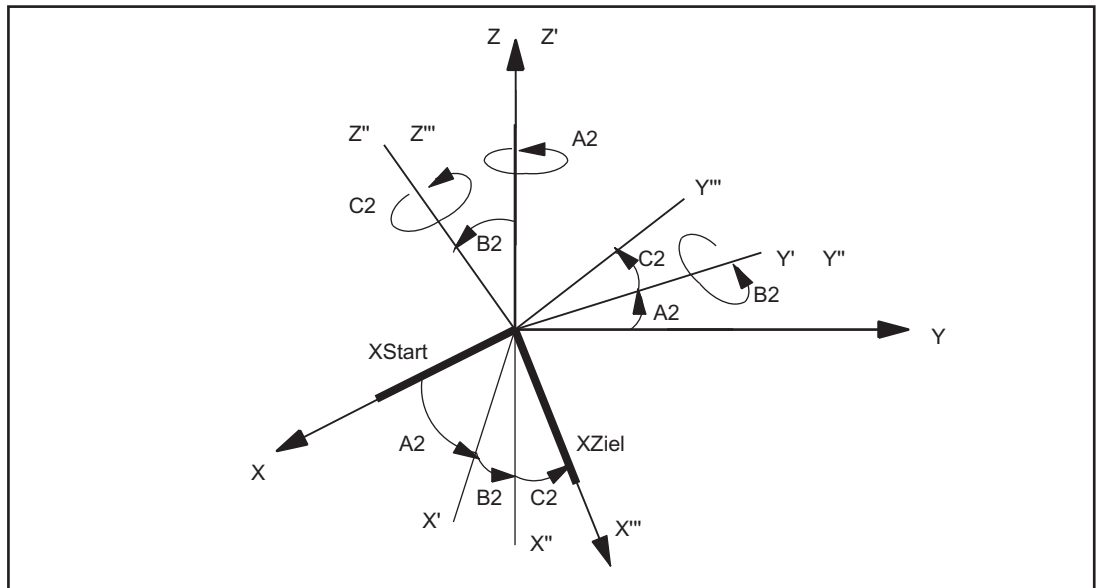


Bild 2-26 3 Orientierungsachsen für die 1. Orientierungstransformation für eine Kinematik mit 6 transformierten Achsen

Beispiel 2:

3 Orientierungsachsen für die 2. Orientierungstransformation für eine Kinematik mit 5 transformierten Achsen. Die Drehung soll in dieser Reihenfolge ablaufen:

- zuerst um die X-Achse,
- danach um die Y-Achse und
- anschließend um die Z-Achse.

Der Werkzeugvektor soll in Z-Richtung zeigen.

Programmcode	Kommentar
CHANDATA(1)	
\$MC_TRAFO5_TOOL_VECTOR_2=2	; Werkzeugvektor in Z-Richtung
\$MC_TRAFO5_ORIAX_ASSIGN_TAB_1[0]=4	; Kanalindex 1. Orient.achse
\$MC_TRAFO5_ORIAX_ASSIGN_TAB_1[1]=5	; Kanalindex 2. Orient.achse
\$MC_TRAFO5_ORIAX_ASSIGN_TAB_1[2]=0	; Kanalindex 3. Orient.achse
\$MC_ORIAX_TURN_TAB_1[0]=1	; X-Richtung
\$MC_ORIAX_TURN_TAB_1[1]=2	; Y-Richtung
\$MC_ORIAX_TURN_TAB_1[2]=3	; Z-Richtung
CHANDATA(1)	
M17	

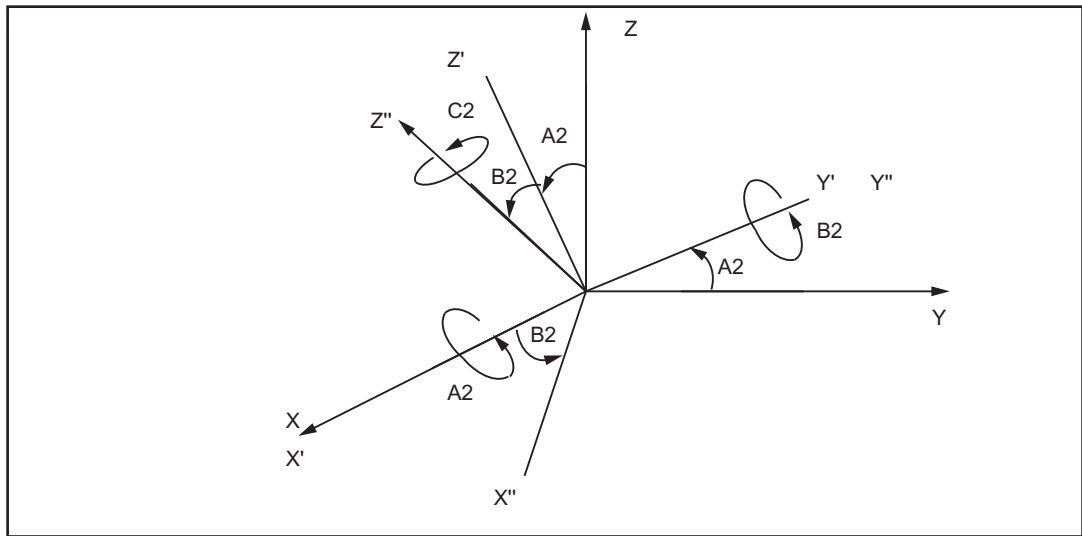


Bild 2-27 3 Orientierungsachsen für die 2. Orientierungstransformation für eine Kinematik mit 5 transformierten Achsen

Die Drehung um den Winkel C2 um die Z''-Achse entfällt in diesem Fall, da sich die Orientierung des Werkzeugvektors allein aus den Winkeln A2 und B2 bestimmen lässt und kein weiterer Freiheitsgrad an der Maschine vorhanden ist.

Literatur:

Programmierhandbuch Arbeitsvorbereitung

2.13.5 Beispiele zu Orientierungsvektoren

2.13.5.1 Beispiel für Polynominterpolation von Orientierungsvektoren

Orientierungsvektor in Z-X Ebene

In den nachfolgenden Beispielen wird der Orientierungsvektor direkt programmiert. Die sich daraus ergebenden Bewegungen der Rundachsen hängen von der jeweiligen Kinematik der Maschine ab.

Programmcode	Kommentar
N10 TRAORI	
N20 POLY	; Polynominterpolation wird ermöglicht.
N30 A3=0 B3=0 C3=1	; Orientierung in +Z-Richtung (Startvektor)
N40 A3=1 B3=0 C3=0	; Orientierung in +X-Richtung (Endvektor)

In N40 wird der Orientierungsvektor in der Z-X Ebene, die vom Start- und Endvektor aufgespannt wird, gedreht. Dabei wird der Winkel PHI in dieser Ebene zwischen den Werten 0 und 90 Grad linear interpoliert (Großkreisinterpolation).

Die zusätzliche Angabe der Polynome für die beiden Winke PHI und PSI bewirkt, dass der interpolierte Orientierungsvektor beliebig zwischen dem Start- und Endvektor liegen kann.

Winkel PHI mittels des Polynoms PHI

Im Unterschied zum obigen Beispiel wird der Winkel PHI mittels des Polynoms $\text{PHI}(u) = (90-10)u + 10 \cdot u^2$ zwischen den Werten 0 und 90 Grad interpoliert.

Der Winkel PSI ist ungleich Null und wird gemäß folgendem Polynom interpoliert:

$$\text{PSI}(u) = -10 \cdot u + 10 \cdot u^2$$

Die maximale "Auskipfung" des Orientierungsvektors aus der Ebene zwischen Start- und Endvektor erhält man in der Mitte des Satzes ($u = 1/2$).

Programmcode	Kommentar
N10 TRAORI	
N20 POLY	; Polynominterpolation wird ermöglicht.
N30 A3=0 B3=0 C3=1	; Orientierung in +Z-Richtung (Startvektor)
N40 A3=1 B3=0 C3=0 PO[PHI]=(10) PO[PSI]=(10)	; in +X-Richtung (Endvektor)

2.13.5.2 Beispiel für Drehungen des Orientierungsvektors

Drehungen mit dem Drehwinkel THETA

Im folgenden Beispiel wird der Drehwinkel vom Startwert 0 Grad zum Endwert 90 Grad linear interpoliert. Der Drehwinkel ändert gemäß einer Parabel oder es kann auch eine Drehung ohne eine Orientierungsänderung ausgeführt werden. Die Werkzeugorientierung wird von der Y-Richtung in die X-Richtung gedreht.

Programmcode	Kommentar
N10 TRAORI	; Aktivierung Orientierungstransformation
N20 G1 X0 Y0 Z0 F5000	; Orientierung des Werkzeugs
N30 A3=0 B3=0 C3=1 THETA=0	; in Z-Richtung mit Drehwinkel 0
N40 A3=1 B3=0 C3=0 THETA=90	; in X-Richtung und Drehung ; um 90 Grad
N50 A3=0 B3=1 C3=0 PO[THT]=(180,90)	; in Y-Richtung und Drehung ; auf 180 Grad
N60 A3=0 B3=1 C3=0 THETA=IC(-90)	; bleibt konstant und Drehung ; auf 90 Grad.
N70 ORIROTT	; Drehwinkel relativ zur ; Orientierungsänderung.
N80 A3=1 B3=0 C3=0 THETA=30	; Drehvektor im Winkel ; 30 Grad zur X-Y Ebene.

N40 Interpolation der Drehwinkel vom Startwert 0 Grad zum Endwert 90 Grad linear interpoliert.

N50 Der Drehwinkel ändert sich von 90 Grad auf 180 Grad gemäß der Parabel

$$\theta(u) = 90 + u^2$$

N60 Es kann auch eine Drehung ausgeführt werden, ohne dass eine Orientierungsänderung stattfindet.

N80 wird die Werkzeugorientierung von der Y-Richtung in X-Richtung gedreht. Dabei liegt die Orientierungsänderung in der X-Y Ebene und der Drehvektor bildet zu dieser Ebene einen Winkel von 30 Grad.

2.13.6 Beispiele für generische Achstransformationen

Im folgenden wird von einer Maschine mit drehbarem Werkzeug ausgegangen, deren erste Rundachse eine C-Achse und deren zweite Rundachse eine B-Achse ist (CB-Kinematik). Die in den Maschinendaten definierte Grundorientierung ist die Winkelhalbierende zwischen der X- und der Z-Achse.

Die relevanten Maschinendaten sehen aus wie folgt:

```

CHANDATA (1)
$MC_TRAFO_TYPE_1 = 24                ; Allgemeine 5-Achstransformation
                                       ; drehbares Werkzeug

$MC_TRAFO5_AXIS1_1[0] = 0.0
$MC_TRAFO5_AXIS1_1[1] = 0.0
$MC_TRAFO5_AXIS1_1[2] = 1.0          ; 1. Rundachse ist parallel zu Z.
$MC_TRAFO5_AXIS2_1[0] = 0.0
$MC_TRAFO5_AXIS2_1[1] = 1.0
$MC_TRAFO5_AXIS2_1[2] = 0.0          ; 2. Rundachse ist parallel zu Y.
$MC_TRAFO5_BASE_ORIENT_1[0] = 1.0
$MC_TRAFO5_BASE_ORIENT_1[1] = 0.0
$MC_TRAFO5_BASE_ORIENT_1[2] = 1.0
M30
    
```

Beispielprogramm:

Programmcode	Kommentar
N10 \$TC_DP1[1,1]=120	; Schaftfräser
N20 \$TC_DP3[1,1]= 0	; Längenkorrekturvektor
N30	
N40	; Definition des Werkzeugträgers
N50 \$TC_CARR7[1] = 1	; Komponente der 1. Drehachse ; in X-Richtung
N60 \$TC_CARR11[1] = 1	; Komponente der 2. Drehachse ; in Y-Richtung
N70 \$TC_CARR13[1] = -45	; Drehwinkel der 1. Achse
N80 \$TC_CARR14[1] = 0	; Drehwinkel der 2. Achse
N90	
N100 X0 Y0 Z0 B0 C0 F10000 ORIWKS G17	
N110 TRAORI()	; Anwahl der Trafo-Grundorientierung ; aus Maschinendaten
N120 C3=1	; Orientierung parallel zu Z

Programmcode	Kommentar
N130 T1 D1	; einstellen → B-45 C0
N140 C3=1	; Grundorientierung ist jetzt parallel Z
N150 G19	; Orientierung parallel zu Z
N160 C3=1	; einstellen → B0 C0
N170 G17 TCARR=1 TCOABS	; Grundorientierung ist jetzt Winkel-
N180 A3=1	; halbierende Y-Z
N190 B3=1 C3=1	; Orientierung parallel zu X
N200 TRAORI(,2.0, 3.0, 6.0)	; einstellen → B-90 C-135
N210 A3=2 B3=3 C3=6	; Orientierg. parallel zur
N220 TOFRAME	; Grundorientierg. → B0 C0
N230 G91 Z7	; Z-Achse zeigt in Richtung
N240 C3=1	; der Orientierung
N250 M30	; 7 mm in neue Z-Richtung
	; verfahren → X2 Y3 Z6
	; Orientierg. parallel zur
	; neuen Z-Achse → B0 C0

2.13.6.1 Beispiel für eine Generische 6-Achs-Transformation

Aufruf mit Parametrierung über Parametern

Aktivierung einer 6-Achs-Transformation mit anschließenden Orientierungsänderungen und Verfahrbewegungen:

Programmcode	Kommentar
N10 A0 B0 X0 Y0 Z0	
N20 TRAORI(1, , , , 0,0,0, 0,1,0)	; Anwahl mit Parametern:
	; Parameter 5, 6, 7: Orientierungsvektor
	; Parameter 8, 9, 10: Orientierungsnormalenvektor
N30 T1 D1 X10 Y20 Z30 A3=0.5	; Orientierungsänderung, Drehung
C3=1 BN3=1 ORIPLANE ORIWKS	; und Verfahrbewegung
N40 B3=0.5 C3=1 AN3=-1	; Drehung, Orientierung konstant
M30	

Aufruf mit Parametrierung über Werkzeugdaten

Definition eines Werkzeugs, bei dem die Orientierung mit einem Orientierungsvektor von (1,0 ; 0,0 ; 0,5) vom Standard abweicht. Der Orientierungsvektor liegt bei G17 in der X-Z-Ebene und ist ausgehend von der X-Achse in Richtung der Z-Achse um 26.565 Grad geneigt:

$$\tan^{-1}(\$TC_DPV5[2,2] / \$TC_DPV3[2,2]) = \tan^{-1}(0,5 / 1,01) = \tan^{-1}(0,5) = 26,565^\circ$$

Der Orientierungsnormalenvektor ist ebenfalls angegeben. Da nur \$TC_DPVN4[2,2] ungleich Null ist, zeigt er in Y-Richtung. Orientierungsvektor und Orientierungsnormalenvektor stehen senkrecht aufeinander. Eine Orthogonalisierung ist nicht notwendig. Der programmierte Orientierungsnormalenvektor wird nicht modifiziert.

Programmcode	Kommentar
N100 \$TC_DP1[2,2]=120	; Schaftfräser
N110 \$TC_DP3[2,2]= 20	; Längenkorrekturvektor
N120 \$TC_DPV[2,2]= 0	; Werkzeugschneidenorientierung ; Orientierungsvektor Werkzeugschneide
N130 \$TC_DPV3[2,2]= 1	; X-Komponente
N140 \$TC_DPV4[2,2]= 0	; Y-Komponente
N150 \$TC_DPV5[2,2]= 0.5	; Z-Komponente ; Orientierungsnormalenvektor
N160 \$TC_DPVN3[2,2]= 0	; X-Komponente
N170 \$TC_DPVN4[2,2]= 1	; Y-Komponente
N180 \$TC_DPVN5[2,2]= 0	; Z-Komponente
N200 TRAORI ()	; Aufruf mit Grundorientierung
N210 A3=5 C3=10 BN3=1	; Rundachsen in Grundstellung bringen
N220 C3=1	; Orientierung in Z-Richtung □ ; Werkzeug dreht um 26.565 Grad
N230 THETA=IC(90)	; Orientierungsnormalenvektor inkrementell ; um 90 Grad drehen. Vektor zeigt in negative X-Richtung
N240 M30	

2.13.6.2 Beispiel für eine generische 7-Achs-Transformation

Beispiel für eine generische 7-Achs-Transformation

Aktivierung einer 7-Achs-Transformation mit anschließenden Orientierungsänderungen und Verfahrbewegungen:

Programm	Kommentar
N10 TRAF00F	
N20 a0 b0 c0 x0 y0 z0 e=0	
N30 \$MC_TRAFO5_AXIS1_1[2] = 1	; 1. Rundachse zeigt in Z-Richtung
N40 \$MC_TRAFO5_AXIS1_2[0] = 1	; 2. Rundachse zeigt in X-Richtung
N50 \$MC_TRAFO5_AXIS1_3[2] = 1	; 3. Rundachse zeigt in Z-Richtung

Programm	Kommentar
N60 \$MC_TRAFO7_EXT_AXIS1_1[0] = 1	; 7. Achse zeigt in X-Richtung
N70 \$MC_TRAFO_BASE_ORIENT_1[2] = 1	; Orientierungsvektor
N80 \$MC_TRAFO_BASE_ORIENT_NORMAL_1[1] = 1	; Orientierungsnormalenvektor
N90 NEWCONF	
N100 traori()	
N110 G1 t1 d1 x10 y0 z50 c3=1 an3=1 bn3=1 orivect oriwks G19 F10000	
N120 G2 y50 z0 b3=1 e=DC(90) CR=50	; 1. Viertelkreis
N130 G2 y0 z-50 c3=-1 e=DC(180) CR=50	; 2. Viertelkreis
N140 G2 y-50 z0 b3=-1 e=DC(270) CR=50	; 3. Viertelkreis
N150 G2 y0 z50 c3=1 e=DC(0) CR=50	; 4. Viertelkreis
N200 M30	

Hinweis

Beim Verfahren der Viertelkreise im Beispiel dreht sich nur die 7. Achse E um 360° Grad. Die Maschine bleibt raumfest stehen.

2.13.6.3 Beispiel für die Modifikation der Rundachsbewegung

Es sei eine 5-Achs-Maschine vom Maschinentyp 1 (Zweiachsen-Schwenkkopf mit CA-Kinematik) gegeben, bei der beide Rundachsen das Werkzeug drehen (Trafotyp 24). Die erste Rundachse ist eine Modulo-Achse parallel zu Z (C-Achse), die zweite Rundachse ist parallel zu Y (B-Achse) und hat einen Verfahrbereich von -5 Grad bis +185 Grad.

Damit eine Veränderung immer zulässig ist, hat das folgende Maschinendatum den Wert 2:

MD21180 \$MC_ROT_AX_SWL_CHECK_MODE (Check Softwarelimits für Ori.-Achsen)

```
N10 X0 Y0 Z0 B0 C0
N20 TRAORI( ) ; Grundorientierung 5-Achs-Transformation
N30 B-1 C10 ; Rundachspositionen B-1 und C10
N40 A3=-1 C3=1 ORIWKS ; Großkreisinterpolation im WKS
N50 M30
```

Im Beispielprogramm steht die Maschine am Anfang des Satzes N40 auf den Rundachspositionen B-1 C10. Die programmierte Endorientierung ist mit den beiden Achsstellungen B-45 C0 (1. Lösung) oder B45 C180 (2. Lösung) zu erreichen.

Zunächst wird die erste Lösung ausgewählt, da sie der Ausgangsorientierung am nächsten liegt und im Gegensatz zur zweiten Lösung mittels Großkreisinterpolation (ORIWKS) erreichbar ist. Diese Position ist aber wegen der Achsgrenzen der B-Achse **nicht** erreichbar.

Es wird deshalb die zweite Lösung verwendet, d. h. die Endposition ist B45 C180. Die Endorientierung wird mittels Achsinterpolation erreicht. Die programmierte Orientierungsbahn kann nicht eingehalten werden.

2.14 Datenlisten

2.14.1 Maschinendaten

2.14.1.1 Allgemeine Maschinendaten

Nummer	Bezeichner: \$MN_	Beschreibung
10620	EULER_ANGLE_NAME_TAB	Name der Eulerwinkel oder Namen der Orientierungsachsen
10630	NORMAL_VECTOR_NAME_TAB	Name der Normalvektoren
10640	DIR_VECTOR_NAME_TAB	Name der Richtungsvektoren
10642	ROT_VECTOR_NAME_TAB	Name der Drehvektoren
10644	INTER_VECTOR_NAME_TAB	Name der Zwischenvektor-Komponente
10646	ORIENTATION_NAME_TAB	Bezeichner für die Programmierung einer zweiten Orientierungsbahn
10648	NUTATION_ANGLE_NAME	Name des Orientierungswinkels
10670	STAT_NAME	Name der Stellungsinformation
10672	TU_NAME	Name der Stellungsinformation der Achsen
10674	PO_WITHOUT_POLY	Erlaubt die Programmierung von PO[] ohne dass POLY aktiv ist

2.14.1.2 Kanal-spezifische Maschinendaten

Nummer	Bezeichner: \$MC_	Beschreibung
20150	GCODE_RESET_VALUES[n]	Löschstellung der G-Gruppen
20152	GCODE_RESET_MODE[n]	Einstellung nach Reset/Teileprogrammende
20482	COMPRESS_MODE	Mode des Kompressors
20621	HANDWH_ORIAX_MAX_INCR_SIZE	Begrenzung Handrad Inkrement
20623	HANDWH_ORIAX_MAX_INCR_VSIZE	Orientierungsgeschwindigkeitsüberlagerung
21094	ORIPATH_MODE	Einstellung für bahnrelative Orientierung
21100	ORIENTATION_IS_EULER	Winkeldefinition bei der Orientierungsprogrammierung
21102	ORI_DEF_WITH_G_CODE	Definition der Orientierungswinkel A2, B2, C2
21104	ORI_IPO_WITH_G_CODE	Definition der Interpolationsart für die Orientierung
21106	CART_JOG_SYSTEM	Koordinatensystem beim Kartesischen JOG
21108	POLE_ORI_MODE	Verhalten bei Großkreisinterpolation in Polposition
21120	ORIAX_TURN_TAB_1[n]	Zuordnung der Drehung der Orientierungsachsen um die Bezugsachsen Definition 1 [n = 0..2]
21130	ORIAX_TURN_TAB_2[n]	Zuordnung der Drehung der Orientierungsachsen um die Bezugsachsen Definition 2 [n = 0..2]
21132	ORI_DISP_IS_MODULO[n]	Modulo-Anzeige der Positionen von Orientierungsachsen [n = 0..2]

Nummer	Bezeichner: \$MC_	Beschreibung
21134	ORI_DISP_MODULO_RANGE	Größe des Modulo-Bereichs für Anzeige der Orientierungsachsen
21136	ORI_DISP_MODULO_RANGE_START	Startposition des Modulo-Bereichs für Anzeige der Orientierungsachsen
21150	JOG_VELO_RAPID_ORI[n]	Konventioneller Eilgang für Orientierungsachsen im Kanal [n = 0..2]
21155	JOG_VELO_ORI[n]	Konventionelle Orientierungsachsgeschwindigkeit [n = 0..2]
21160	JOG_VELO_RAPID_GEO[n]	Konventioneller Eilgang für Geometrieachsen im Kanal [n = 0..2]
21165	JOG_VELO_GEO[n]	Konventionelle Geometrieachsgeschwindigkeit [n = 0..2]
21170	ACCEL_ORI[n]	Beschleunigung für Orientierungsachsen [n = 0..2]
21180	ROT_AX_SWL_CHECK_MODE	Check Softwarelimits für Orientierungsachsen
21186	TOCARR_ROT_OFFSET_FROM_FR	Offset der TOCARR-Rundachsen
21190	TOFF_MODE	Wirkungsweise Online-Korrektur in Werkzeurichtung
21194	TOFF_VELO	Geschwindigkeit Online-Korrektur in Werkzeurichtung
21196	TOFF_ACCEL	Beschleunigung Online-Korrektur in Werkzeurichtung
24100	TRAFO_TYPE_1	Definition der Transformation 1 im Kanal
24110	TRAFO_AXES_IN_1[n]	Achszuordnung für Transformation 1 [Achsisindex]
24120	TRAFO_GEOAX_ASSIGN_TAB_1[n]	Zuordnung Geometrieachse zu Kanalachse für Transformation 1 [Geometrie-Nr.]
24200	TRAFO_TYPE_2	Definition der Transformation 2 im Kanal
24210	TRAFO_AXES_IN_2[n]	Achszuordnung für Transformation 2 [Achsisindex]
24220	TRAFO_GEOAX_ASSIGN_TAB_2[n]	Zuordnung Geometrieachse zu Kanalachse für Transformation 2 [Geometrie-Nr.]
24300	TRAFO_TYPE_3	Definition der Transformation 3 im Kanal
24310	TRAFO_AXES_IN_3[n]	Achszuordnung für Transformation 3 [Achsisindex]
24320	TRAFO_GEOAX_ASSIGN_TAB_3[n]	Zuordnung Geometrieachse zu Kanalachse für Transformation 3 [Geometrie-Nr.]
24400	TRAFO_TYPE_4	Definition der Transformation 4 im Kanal
24410	TRAFO_AXES_IN_4[n]	Achszuordnung für Transformation 4 [Achsisindex]
24420	TRAFO_GEOAX_ASSIGN_TAB_4[n]	Zuordnung Geometrieachse zu Kanalachse für Transformation 4 [Geometrie-Nr.]
24430	TRAFO_TYPE_5	Definition der Transformation 5 im Kanal
24432	TRAFO_AXES_IN_5[n]	Achszuordnung für Transformation 5 [Achsisindex]
24434	TRAFO_GEOAX_ASSIGN_TAB_5[n]	Zuordnung Geometrieachse zu Kanalachse für Transformation 5 [Geometrie-Nr.]
24440	TRAFO_TYPE_6	Definition der Transformation 6 im Kanal
24442	TRAFO_AXES_IN_6[n]	Achszuordnung für Transformation 6 [Achsisindex]
24444	TRAFO_GEOAX_ASSIGN_TAB_6[n]	Zuordnung Geometrieachse zu Kanalachse für Transformation 6 [Geometrie-Nr.]
24450	TRAFO_TYPE_7	Definition der Transformation 7 im Kanal
24452	TRAFO_AXES_IN_7[n]	Achszuordnung für Transformation 7 [Achsisindex]

Nummer	Bezeichner: \$MC_	Beschreibung
24454	TRAFO_GEOAX_ASSIGN_TAB_7[n]	Zuordnung Geometrieachse zu Kanalachse für Transformation 7 [Geometrie-Nr.]
24460	TRAFO_TYPE_8	Definition der Transformation 8 im Kanal
24462	TRAFO_AXES_IN_8[n]	Achszuordnung für Transformation 8 [Achsisindex]
24464	TRAFO_GEOAX_ASSIGN_TAB_8[n]	Zuordnung Geometrieachse zu Kanalachse für Transformation 8 [Geometrie-Nr.]
24470	TRAFO_TYPE_9	Definition der Transformation 9 im Kanal
24472	TRAFO_AXES_IN_9[n]	Achszuordnung für Transformation 9 [Achsisindex]
24474	TRAFO_GEOAX_ASSIGN_TAB_9[n]	Zuordnung Geometrieachse zu Kanalachse für Transformation 9 [Geometrie-Nr.]
24480	TRAFO_TYPE_10	Definition der Transformation 10 im Kanal
24482	TRAFO_AXES_IN_10[n]	Achszuordnung für Transformation 10 [Achsisindex]
24484	TRAFO_GEOAX_ASSIGN_TAB_10[n]	Zuordnung Geometrieachse zu Kanalachse für Transformation 10 [Geometrie-Nr.]
24500	TRAFO5_PART_OFFSET_1[n]	Verschiebungsvektor der 5-Achs-Transformation 1 [n = 0.. 2]
24510	TRAFO5_ROT_AX_OFFSET_1[n]	Positionsoffset der Rundachse 1/2 für die 5-Achs-Transformation 1 [Achsis-Nr.]
24520	TRAFO5_ROT_SIGN_IS_PLUS_1[n]	Vorzeichen der Rundachse 1/2 für die 5-Achs-Transformation 1 [Achsis-Nr.]
24530	TRAFO5_NON_POLE_LIMIT_1	Definition des Polbereichs für 5-Achs-Transformation 1
24540	TRAFO5_POLE_LIMIT_1	Endwinkeltoleranz bei Interpolation durch den Pol für 5-Achs-Transformation 1
24550	TRAFO5_BASE_TOOL_1[n]	Vektor des Basiswerkzeugs bei Aktivierung für die 5-Achs-Transformation 1 [0.. 2]
24558	TRAFO5_JOINT_OFFSET_PART_1[n]	Vektor kinematischer Versatz im Tisch für die 5-Achs-Transformation 1 [n = 0.. 2]
24560	TRAFO5_JOINT_OFFSET_1[n]	Vektor des kinematischen Versatzes der 5-Achs-Transformation 1 [n = 0.. 2]
24561	TRAFO6_JOINT_OFFSET_2_3_1[n]	Vektor des kinematischen Versatzes der 6-Achs-Transformation 2_3_1
24562	TRAFO5_TOOL_ROT_AX_OFFSET_1[n]	Offset des Schwenkpunktes der 1. 5-Achs-Transformation mit geschwenkter Linearachse
24564	TRAFO5_NUTATOR_AX_ANGLE_1	Winkel der 2. drehenden Achse für den kardanischen Fräskopf.
24570	TRAFO5_AXIS1_1[n]	Vektor für die erste Rundachse und die erste Orientierungstransform. [n = 0.. 2]
24572	TRAFO5_AXIS2_1[n]	Vektor für die zweite Rundachse und die erste Transformation [n = 0.. 2]
24673	TRAFO5_AXIS3_1[n]	Richtung dritte Rundachse bei der allgemeinen 6-Achs-Transformation (Trafotyp 24, 40, 56, 57)
24574	TRAFO5_BASE_ORIENT_1[n]	Grundorientierung für die erste Transformation [n = 0.. 2]
24576	TRAFO6_BASE_ORIENT_NORMAL_1[n]	Werkzeugnormalenvektor für die erste Transformation [n = 0.. 2]

Nummer	Bezeichner: \$MC_	Beschreibung
24580	TRAF05_TOOL_VECTOR_1	Richtung des Werkzeugvektors für die erste 5-Achs-Transformation 1
24582	TRAF05_TCARR_NO_1	TCARR-Nummer für die erste 5-Achs- Transformation 1
24585	TRAF05_ORIAX_ASSIGN_TAB_1[n]	Zuordnung der Orientierungsachsen zu den Kanalachsen für die Orientierungstransformation 1 [n = 0.. 2]
24590	TRAF05_ROT_OFFSET_FROM_FR_2	Offset der Trafo-Rundachsen aus NPV
24594	TRAF07_EXT_ROT_AX_OFFSET_1	Winkel-Offset der 1. externen Rundachse
24595	TRAF07_EXT_AXIS1_1	Richtung der 1. externen Rundachse
24600	TRAF05_PART_OFFSET_2[n]	Verschiebungsvektor der 5-Achs-Transformation 2 [n = 0.. 2]
24610	TRAF05_ROT_AX_OFFSET_2[n]	Positionsoffset der Rundachse 1/2 für die 5-Achs-Transformation 2 [Achs-Nr.]
24620	TRAF05_ROT_SIGN_IS_PLUS_2[n]	Vorzeichen der Rundachse 1/2 für die 5-Achs-Transformation 2 [Achs-Nr.]
24630	TRAF05_NON_POLE_LIMIT_2	Definition des Polbereichs für 5-Achs-Transformation 2
24640	TRAF05_POLE_LIMIT_2	Endwinkeltoleranz bei Interpolation durch den Pol für 5-Achs-Transformation 2
24650	TRAF05_BASE_TOOL_2[n]	Vektor des Basiswerkzeugs bei Aktivierung für die 5-Achs-Transformation 2 [n = 0.. 2]
24658	TRAF05_JOINT_OFFSET_PART_2[n]	Vektor kinematischer Versatz im Tisch für die 5-Achs-Transformation 2 [n = 0.. 2]
24660	TRAF05_JOINT_OFFSET_2[n]	Vektor des kinematischen Versatzes der 5-Achs-Transformation 2 [n = 0.. 2]
24661	TRAF06_JOINT_OFFSET_2_3_2[n]	Vektor des kinematischen Versatzes der 6-Achs-Transformation 2_3_2
24662	TRAF05_TOOL_ROT_AX_OFFSET_2[n]	Offset des Schwenkpunktes der 2. 5-Achs-Transformation mit geschwenkter Linearachse
24664	TRAF05_NUTATOR_AX_ANGLE_2	Winkel der 2. drehenden Achse für den kardanischen Fräskopf
24670	TRAF05_AXIS1_2[n]	Vektor für die erste Rundachse und die zweite Orientierungstransformation [n = 0.. 2]
24672	TRAF05_AXIS2_2[n]	Vektor für die zweite Rundachse u. die zweite Transformation [n = 0.. 2]
24673	TRAF05_AXIS3_2[n]	Richtung dritte Rundachse bei der allgemeinen 6-Achs-Transformation (Typ 24, 40, 56, 57)
24674	TRAF05_BASE_ORIENT_2[n]	Grundorientierung für die zweite Transformation [n = 0.. 2]
24676	TRAF06_BASE_ORIENT_NORMAL_2[n]	Werkzeugnormalenvektor für die zweite Transformation [n = 0.. 2]
24680	TRAF05_TOOL_VECTOR_2	Richtung des Werkzeugvektors für die zweite 5-Achs-Transformation 2
24682	TRAF05_TCARR_NO_2	TCARR-Nummer für die zweite 5-Achs- Transformation 2
24685	TRAF05_ORIAX_ASSIGN_TAB_2[n]	Zuordnung der Orientierungsachsen zu den Kanalachsen für die Orientierungstransformation 2 [n = 0.. 2]
24694	TRAF07_EXT_ROT_AX_OFFSET_2	Winkel-Offset der 2. externen Rundachse

Nummer	Bezeichner: \$MC_	Beschreibung
24695	TRAFO7_EXT_AXIS1_2	Richtung der 2. externen Rundachse
25294	TRAFO7_EXT_ROT_AX_OFFSET_3	Winkel-Offset der 3. externen Rundachse
25295	TRAFO7_EXT_AXIS1_3	Richtung der 3. externen Rundachse
25394	TRAFO7_EXT_ROT_AX_OFFSET_4	Winkel-Offset der 4. externen Rundachse
25395	TRAFO7_EXT_AXIS1_4	Richtung der 4. externen Rundachse
28580	MM_ORIPATH_CONFIG	Konfiguration für bahnrelative Orientierung ORIPATH

2.14.2 Settingdaten

2.14.2.1 Allgemeine Settingdaten

Nummer	Bezeichner: \$SN_	Beschreibung
41110	JOG_SET_VELO	Geometrieachsen
41130	JOG_ROT_AX_SET_VELO	Orientierungsachsen

2.14.2.2 Kanal-spezifische Settingdaten

Nummer	Bezeichner: \$SC_	Beschreibung
42475	COMPRESS_CONTOUR_TOL	Max. Konturabweichung beim Kompressor
42476	COMPRESS_ORI_TOL	Max. Winkelabweichung der Werkzeugorientierung beim Kompressor
42477	COMPRESS_ORI_ROT_TOL	Max. Winkelabweichung der Drehung des Werkzeugs beim Kompressor
42650	CART_JOG_MODE	Koordinatensystem für Kartesisches manuelles Verfahren
42660	ORI_JOG_MODE	Definition einer virtuellen Kinematik für JOG
42670	ORIPATH_SMOOTH_DIST	Wegstrecke zur Glättung der Orientierung
42672	ORIPATH_SMOOTH_TOL	Toleranz zur Glättung der Orientierung
42970	TOFF_LIMIT	Obergrenze des Korrekturwertes \$AA_TOFF

2.14.3 Signale

2.14.3.1 Signale an Kanal

Signalname	SINUMERIK 840D sl	SINUMERIK 828D
PTP-Fahren aktivieren	DB21,DBX29.4	-

2.14.3.2 Signale von Kanal

Signalname	SINUMERIK 840D sl	SINUMERIK 828D
Transformation aktiv	DB21,DBX33.6	DB330x.DBX1.6
Nummer der aktiven G-Befehle der G-Gruppe 25	DB21,DBB232	-
PTP-Fahren aktiv	DB21,DBX317.6	-
Online-Werkzeuglängenkorrektur (TOFF) aktiv	DB21,DBX318.2	-
Online-Werkzeuglängenkorrektur (TOFF): Korrekturbewegung aktiv	DB21,DBX318.3	-

G1: Gantry-Achsen

3.1 Kurzbeschreibung

Bei Portalmaschinen werden verschiedene Maschinenelemente, wie z.B. das Portal und der Querbalken, jeweils von mehreren parallel arbeitenden Achsen bewegt. Die Achsen, die gemeinsam ein Maschinenteil bewegen, werden als Gantry-Achsen bzw. Gantry-Verbund bezeichnet. Aufgrund des mechanischen Aufbaus sind die Gantry-Achsen starr miteinander verbunden und müssen demzufolge von der Steuerung immer synchron verfahren werden.

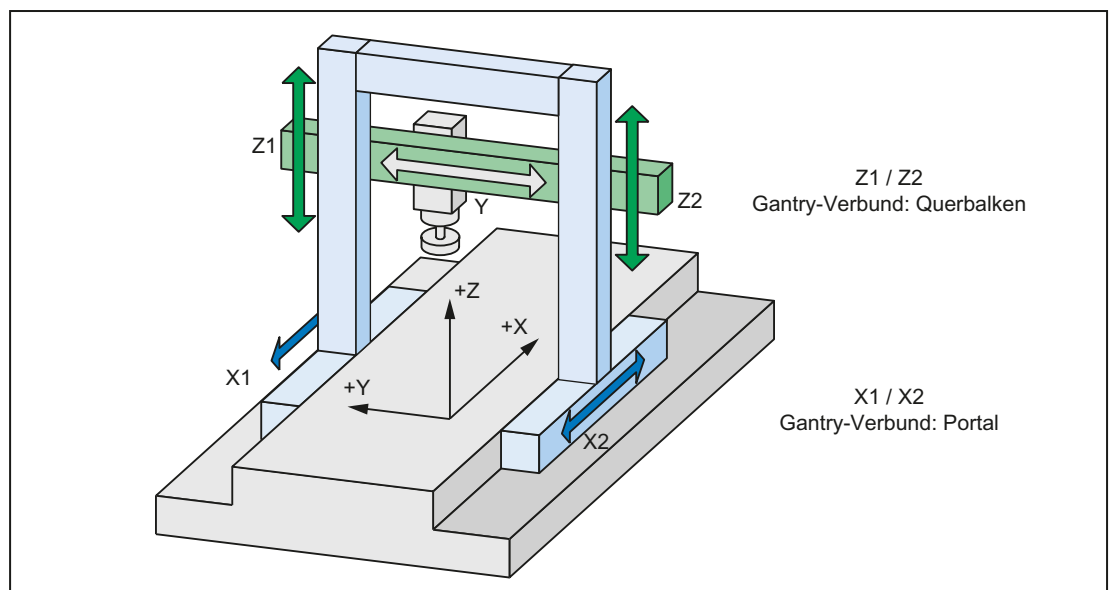


Bild 3-1 Beispiel: Portalfräsmaschine mit Portal und Querbalken

Führungsachse

Die Führungsachse des Gantry-Verbundes ist die Achse, die den Gantry-Verbund repräsentiert. Nur über sie werden die Verfahrbewegungen des Gantry-Verbundes programmiert.

Gleichlaufachsen

Die Gleichlaufachsen des Gantry-Verbundes sind die Achsen, die aufgrund ihrer Kopplung mit der Führungsachse von der Steuerung automatisch mit verfahren werden. Einer Führungsachse können beliebig viele Gleichlaufachsen zugeordnet werden.

Synchronlaufdifferenz

Die Synchronlaufdifferenz ist die Abweichung des axialen Istwerts einer Gleichlaufachse von seiner idealen Position bezogen auf den Istwert der Führungsachse. Die Synchronlaufdifferenz wird von der Steuerung kontinuierlich überwacht. Bei Überschreiten einer Warngrenze wird eine Meldung angezeigt. Bei Überschreiten der Alarmgrenze wird der gesamte Gantry-Verbund stillgesetzt. Die Grenzwerte sind über Maschinendaten parametrierbar.

3.2 Funktion "Gantry-Achsen"

3.2.1 Definition eines Gantry-Verbundes

Definition

Die Achsen eines Gantry-Verbundes werden über folgendes axiale Maschinendatum festgelegt:

MD37100 \$MA_GANTRY_AXIS_TYPE[AX1] = xy	
x	10er Dezimalstelle: Typ der Gantry-Achse (Führungs- oder Gleichlaufachse)
y	1er Dezimalstelle: ID des Gantry-Verbundes

Es können maximal 8 Gantry-Verbünde (Gantry-Verbund ID: 1 - 8) definiert werden. Die Gantry-Verbund ID muss entsprechend der zugeordneten Achsen kanal- oder NCU-übergreifend eindeutig sein.

Einem Gantry-Verbund können prinzipiell beliebig viele Gleichlaufachsen zugeordnet werden.

Beispiel

Definition eines Gantry-Verbundes mit ID=1, Führungssachse AX1 und Gleichlaufachse AX2

- MD37100 \$MA_GANTRY_AXIS_TYPE[AX1] = 01 (Führungssachse)
- MD37100 \$MA_GANTRY_AXIS_TYPE[AX2] = 11 (Gleichlaufachse)

Randbedingungen

Für einen Gantry-Verbund gelten folgende Randbedingungen:

- Ein Gantry-Verbund darf keine Spindel enthalten.
- Eine Gleichlaufachse darf keine konkurrierende POS-Achse sein.
- Eine Gleichlaufachse darf nicht zu einer Transformation gehören.
- Eine Gleichlaufachse darf nicht Folgeachse einer anderen Achskopplung sein.
- Eine Gleichlaufachse darf nicht Führungssachse einer anderen Achskopplung sein.
- Alle Achsen eines Gantry-Verbundes müssen vom gleichen Achstyp, Linear- oder Rundachse, sein: MD30300 \$MA_IS_ROT_AX (Rundachse/Spindel)

Hinweis

Antrieboptimierung

An einem Antriebsgerät SINAMICS S120 können maximal 3 Antriebe gleichzeitig optimiert bzw. vermessen (Drehzahlregleroptimierung/Funktionsgenerator) werden. Es wird daher dringend empfohlen bei einer Kopplung mit mehr als 3 gleichzeitig gekoppelten Antrieben, diese auf mehrere Antriebsgeräte zu verteilen.

3.2.2 Überwachung der Synchronlaufdifferenz

Grenzwerte für die Überwachung

Bezüglich der Synchronlaufdifferenz können 2 Grenzwerte vorgegeben werden.

Gantry-Warngrenze

Die Gantry-Warngrenze wird über folgendes Maschinendatum eingestellt:

MD37110 \$MA_GANTRY_POS_TOL_WARNING (Gantry-Warngrenze)

Überschreitet die Synchronlaufdifferenz die Gantry-Warngrenze, wird die Meldung "Warngrenze überschritten" angezeigt. Zusätzlich wird das NC/PLC-Nahtstellensignal gesetzt:

DB31, ... DBX101.3 = 1 (Gantry-Warngrenze überschritten)

Nach Unterschreitung der Warngrenze werden Meldung und Nahtstellensignal automatisch wieder zurückgesetzt.

Hinweis

Gantry-Warngrenze

Soll die Meldung "Warngrenze überschritten" nicht angezeigt werden, ist in das MD37110 der Wert 0 einzutragen.

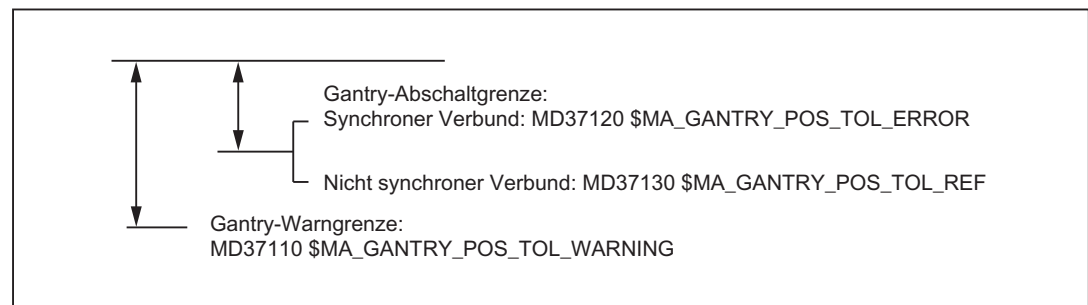
Gantry-Abschaltgrenze

Die Gantry-Abschaltgrenze wird über folgende Maschinendaten eingestellt:

- für den synchronisierten Gantry-Verbund:
MD37120 \$MA_GANTRY_POS_TOL_ERROR
- für den nicht synchronisierten Gantry-Verbund:
MD37130 \$MA_GANTRY_POS_TOL_REF

Überschreitet die Synchronlaufdifferenz die Gantry-Abschaltgrenze, wird der Alarm 10653 "Fehlergrenze überschritten" angezeigt. Zusätzlich wird das NC/PLC-Nahtstellensignal gesetzt:

DB31, ... DBX101.2 = 1 (Gantry-Abschaltgrenze überschritten)



Der Alarm wird auch angezeigt, wenn der Gantry-Verbund geklemmt ist (keine Reglerfreigabe, Gantry-Verbund im "Halten").

3.2.3 Erweiterte Überwachung der Synchronlaufdifferenz

Aktivierung der erweiterten Überwachung

Eine erweiterte Überwachung der Synchronlaufdifferenz kann über folgendes Maschinendatum aktiviert werden:

MD37150 \$MA_GANTRY_FUNCTION_MASK, Bit 0 = 1

Bei der erweiterten Überwachung wird auch eine während des Nachführens oder bei gelöstem Gantry-Verbund entstandene Synchronlaufdifferenz zwischen Führungs- und Gleichlaufachse berücksichtigt.

Die erweiterte Überwachung ist nach dem Hochlauf der NC nach dem ersten Referenzieren (Inkrementalgeber) bzw. Synchronisieren (Absolutwertgeber) wirksam.

Überschreitung der Gantry-Abschaltgrenze

Wird bei aktiver erweiterter Überwachung die Abschaltgrenze der Synchronlaufdifferenz überschritten wird der Alarm 10653 "Fehlergrenze überschritten" angezeigt.

Um den Alarm rücksetzen zu können, ist folgendermaßen vorzugehen:

1. Erweiterte Überwachung deaktivieren:
MD37150 \$MA_GANTRY_FUNCTION_MASK, Bit 0 = 0
2. Die im Maschinendatum angezeigte Synchronlaufdifferenz löschen:
MD37135 \$GANTRY_ACT_POS_TOL_ERROR = 0
3. Alarm löschen
4. Die Achsen des Gantry-Verbundes neu referenzieren bzw. synchronisieren
5. Erweiterte Überwachung wieder aktivieren:
MD37150 \$MA_GANTRY_FUNCTION_MASK, Bit 0 = 1

3.2.4 Referenzieren und Synchronisieren der Gantry-Achsen

Anwendungsfall

Wird bei der Führungs- oder Gleichlaufachse ein inkrementelles Messsystem verwendet, muss nach dem Hochlauf der NC ein Referenzieren der Messsysteme unter Beibehaltung der Achskopplung durchgeführt werden.

Nachdem jede Achse des Gantry-Verbundes ihren Referenzpunkt angefahren hat, muss anschließend eine eventuelle Schiefstellung der Gantry-Achsen untereinander ausgeglichen werden (Gantry-Synchronisationslauf). Ist dies erfolgt, wird das NC/PLC-Nahtstellensignal gesetzt:

DB31, ... DBX101.5 = 1 (Gantry-Verbund ist synchron)

Zum Ablauf des Referenzierens bzw. Synchronisierens von Gantry-Achsen (siehe Kapitel "Referenzieren und Synchronisieren von Gantry-Achsen (Seite 162)").

3.2.5 Regeldynamik

Anwendungsfall

Aus Anwendersicht wird ein Gantry-Verbund ausschließlich über die Führungssache verfahren. Die Sollwerte der Gleichlaufachsen werden von der NC zeitsynchron direkt aus den Sollwerten der Führungssache erzeugt und an diese ausgegeben. Zur Minimierung der Synchronlaufdifferenzen müssen dazu die Regeldynamiken aller Achsen eines Gantry-Verbundes gleich eingestellt werden (siehe Kapitel "Inbetriebnahme der Gantry-Achsen (Seite 172)").

Hinweis

Für alle Achsen eines Gantry-Verbunds müssen identische Regeldynamiken eingestellt werden.

Störverhalten

Bei Störungen, die zum Stillsetzen einer Achse des Gantry-Verbundes führen, wird immer der gesamte Gantry-Verbund stillgesetzt.

3.2.6 Gantry-Verbund auftrennen

Beschreibung

Die Achskopplung innerhalb eines Gantry-Verbunds kann über folgendes Maschinendatum aufgehoben werden:

MD37140 \$MA_GANTRY_BREAK_UP = 1 (Gantry-Verbund lösen)

 **VORSICHT**

Kein Synchronlauf

Falls die Gantry-Achsen weiterhin mechanisch verbunden sind, kann in diesem Betriebszustand beim Verfahren der Führungs- oder Gleichlaufachse die Maschine beschädigt werden!

Mit Wirksamwerden der Einstellung können die Achsen des Gantry-Verbundes in den Betriebsarten JOG, AUTOMATIK und MDA einzeln verfahren werden.

Die Überwachung der Synchronlaufdifferenz bzw. die Warn- und Abschaltgrenzen sind nicht aktiv.

Das NC/PLC-Nahtstellensignal "Gantry-Verbund ist synchronisiert" wird zurückgesetzt:

DB31, ... DBX101.5 = 0

3.3 Referenzieren und Synchronisieren von Gantry-Achsen

3.3.1 Einführung

Schiefelage beim Einschalten

Beim Einschalten der Maschine kann die Idealstellung zwischen Führungsachse und Gleichlaufachse verschoben sein (z. B. Schiefelage eines Portals). In der Regel ist diese Verschiebung relativ klein, so dass damit die Gantry-Achsen referenziert werden können.

In Sonderfällen (z. B. Gantry-Achsen wurden durch eine Störung, Netzausfall oder NOT-AUS stillgesetzt) muss vor dem Verfahren der Maßversatz auf zulässige Toleranzwerte überprüft und ggf. eine Ausgleichsbewegung vorgenommen werden.

Für diesen Vorgang muss der Gantry-Verbund mit dem folgenden Maschinendatum aufgelöst werden :

MD37140 \$MA_GANTRY_BREAK_UP (Gantry-Verbund lösen)

Gantry-Synchronisationslauf

Alle Gantry-Achsen müssen nach dem Einschalten der Steuerung zunächst referenziert und anschließend synchronisiert werden. Beim Gantry-Synchronisationslauf fahren alle Gantry-Achsen **ohne Achskopplung die Bezugsposition des Gantry-Verbundes** an. Die Bezugsposition des Gantry-Verbundes ist beim Referenzieren der Gantry-Achsen die **Referenzposition der Führungsachse**:

MD34100 \$MA_REFP_SET_POS (Referenzpunktwert/Zielpunkt bei abstandskodiertem System)

Ansonsten ist die Bezugsposition die **aktuelle Istposition der Führungsachse**.

Diese Vorgänge für das Referenzieren und Synchronisieren der Gantry-Achsen laufen automatisch nach einem speziellen Ablaufschema ab.

Referenziervorgang

Folgendes Ablaufschema ergibt sich beim Referenziervorgang von Gantry-Achsen bei einem inkrementellen Messsystem:

Abschnitt 1:

Referenzieren der Führungsachse

Das achsspezifische Referenzieren der Gantry-Achsen wird mit dem Nahtstellensignal der Führungssachse vom PLC-Anwenderprogramm bei aktiver Maschinenfunktion REF gestartet: DB31, ... DBX4.7/4.6 (Verfahrtaste plus/minus).

Die Führungssachse fährt den Referenzpunkt an (Ablauf entsprechend Referenzpunktfahren).

Literatur:

Funktionshandbuch Grundfunktionen; Referenzpunktfahren (R1)

Die zugehörigen Gleichlaufachsen verfahren synchron mit. Das Erreichen des Referenzpunktes wird mit Nahtstellensignal "Referenziert/Synchronisiert" der Führungssachse angezeigt.

Abschnitt 2:

Referenzieren der Gleichlaufachsen

Sobald die Führungssachse ihren Referenzpunkt angefahren hat, wird **automatisch** die Gleichlaufachse referenziert (entsprechend Referenzpunktfahren).

Literatur:

Funktionshandbuch Grundfunktionen; Referenzpunktfahren (R1)

Für diese Phase wird die Abhängigkeit zwischen Führungs- und Gleichlaufachse steuerungsintern umgeschaltet, so dass jetzt die Führungssachse synchron mit der Gleichlaufachse verfährt. Das Erreichen des Referenzpunktes wird mit NST "Referenziert/Synchronisiert" der Gleichlaufachse angezeigt. Damit ist wieder die bisherige Gantry-Achsabhängigkeit wirksam. Falls im Gantry-Verbund eine weitere Gleichlaufachse definiert ist, wird diese ebenfalls in entsprechender Weise referenziert.

Abschnitt 3:

Gantry-Synchronisationslauf

Nachdem alle Achsen des Gantry-Verbundes referenziert sind, müssen diese auf die festgelegte Bezugsposition synchronisiert werden. Es wird zunächst für jede Gleichlaufachse ein Vergleich der Istposition mit der festgelegten Referenzposition der Führungssachse durchgeführt.

Abhängig von der ermittelten Istwertdifferenz zwischen Führungs- und Gleichlaufachse werden folgende Fälle unterschieden:

- Differenz ist **kleiner** als Gantry-Warngrenze:
MD37110 \$MA_GANTRY_POS_TOL_WARNING (Gantry-Warngrenze)
Der Gantry-Synchronisationslauf wird **automatisch** gestartet. Dabei wird die Meldung "Synchronisation läuft Gantry-Verbund x" angezeigt.
Die Meldung "Synchronisation läuft Gantry-Verbund x" kann unterdrückt werden mit:
MD37150 \$MA_GANTRY_FUNCTION_MASK Bit 2 = 1
Alle Gantry-Achsen fahren auf einen bestimmten Positionswert **ohne Achskopplung** mit der eingestellten Geschwindigkeit im Maschinendatum:
MD34040 \$MA_REFP_VELO_SEARCH_MARKER (Abschaltgeschwindigkeit)
Der Positionswert wurde festgelegt bei der Führungsachse:
MD34100 \$MA_REFP_SET_POS (Referenzpunkt/Zielpunkt bei abstandskoordiniertem System)
Bei Absolutgebern und abstandscodierten Gebern der Führungsachse wird wahlweise auf die momentane Istlage der Führungsachse oder auf den Referenzpunktwert gefahren, eingestellt durch das folgende Maschinendatum:
MD34330 \$MA_REFP_STOP_AT_ABS_MARKER (Abstandscodiertes Längenmesssystem ohne Zielpunkt)
Die dabei verwendete Geschwindigkeit entspricht der Referenzpunkteinfahrtgeschwindigkeit:
MD34070 \$MA_REFP_VELO_POS (Referenzpunkteinfahrtgeschwindigkeit)
Sobald alle Gantry-Achsen ihre Zielposition (Ideallage) erreicht haben, wird das NST "Gantry-Verbund ist synchronisiert" auf 1-Signal gesetzt und die Gantry-Achskopplung wieder aktiviert. Der Lageistwert muss nun bei allen Achsen des Gantry-Verbundes identisch sein. Damit ist der Gantry-Synchronisationslauf beendet.
- Bei mindestens einer Gleichlaufachse ist die Differenz **größer** als die Gantry-Warngrenze:
Es wird das NST "Gantry-Synchronisationslauf startbereit" auf 1-Signal gesetzt und die Meldung "Warte auf Synchronisations-Start des Gantry-Verbunds x" ausgegeben. Der Gantry-Synchronisationslauf wird in diesem Fall nicht automatisch gestartet, sondern muss vom Bediener bzw. vom PLC-Anwenderprogramm explizit gestartet werden. Der Anstoß erfolgt durch das NST "Gantry-Synchronisationslauf starten" bei der Führungsachse. Das Signal wird bei der Führungsachse gesetzt. Der Ablauf ist analog wie oben beschrieben.

Im nachfolgenden Ablaufplan ist der Referenzier- und Synchronisationsvorgang graphisch dargestellt.

3.3 Referenzieren und Synchronisieren von Gantry-Achsen

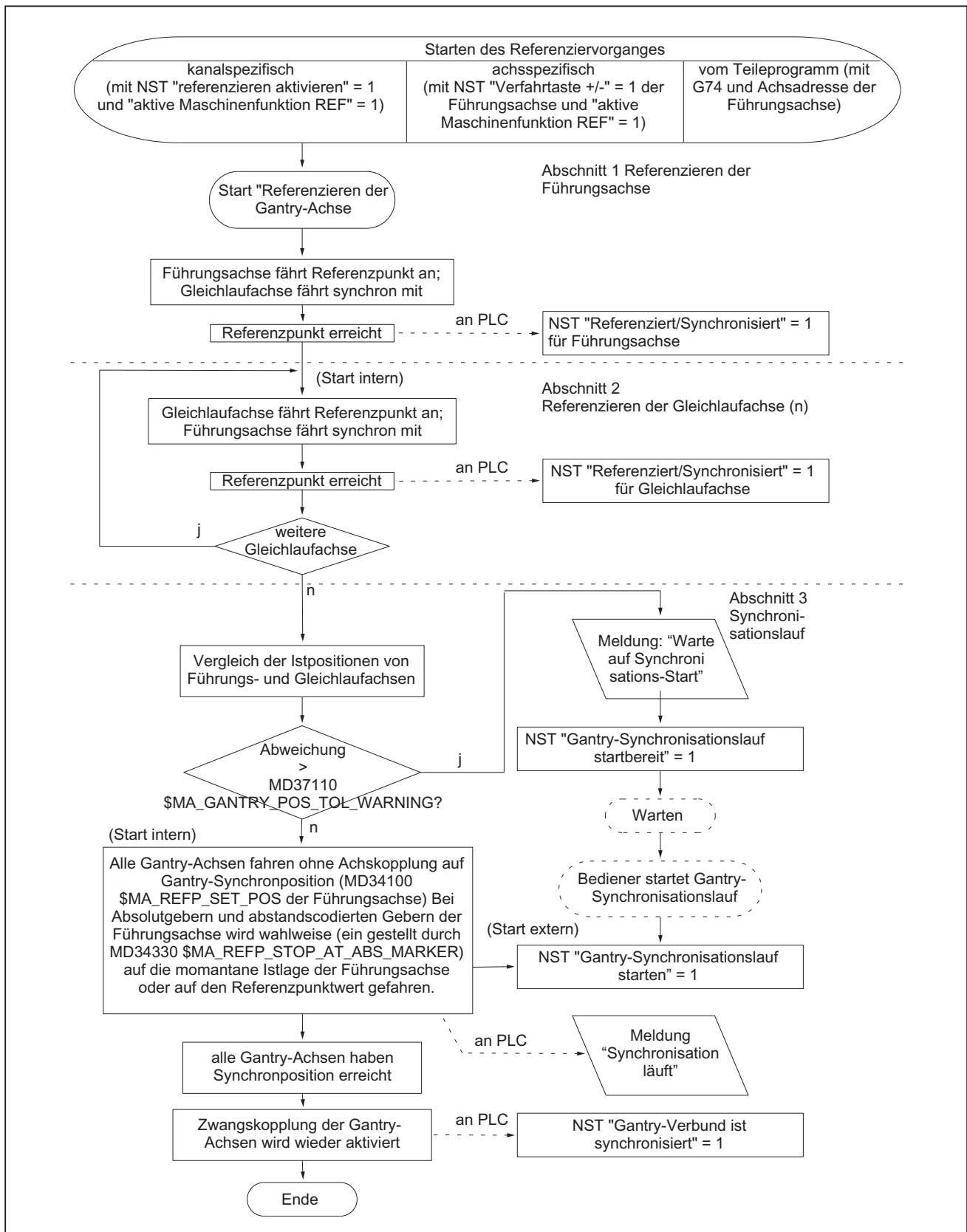


Bild 3-2 Ablaufplan für Referenzier- und Synchronisationsvorgang von Gantry-Achsen

Synchronisationslauf

Ein Synchronisationslauf ist in folgenden Fällen immer nötig:

- nach dem Referenzpunktfahren aller zum Verbund gehörigen Achsen,
- wenn die Synchronisation verloren geht (s unten).

Ablaufunterbrechung

Falls der o. g. Referenziervorgang aufgrund von Störungen bzw. RESET unterbrochen wird, ist wie folgt zu verfahren:

- **Abbruch innerhalb von Abschnitt 1 oder 2:**
Mit Führung Achse erneut Referenzpunkt anfahren (siehe Abschnitt 1)
- **Abbruch bei Abschnitt 3:**
Sofern die Gantry-Achsen noch referenziert sind (NST "Referenziert/Synchronisiert" = 1), kann der Gantry-Synchronisationslauf mit dem NST "Gantry-Verbund synchronisieren" erneut angestoßen werden.

Gantry-Synchronisationslauf erneut starten

Die Synchronisation der Gantry-Achsen kann mit dem NST "Gantry-Synchronisationslauf starten" nur unter folgenden Bedingungen gestartet werden:

- Die Maschinenfunktion JOG/REF muss aktiv sein. Das folgende Nahtstellensignal muss gesetzt sein:
DB11, ... DBX5.2 = 1 (aktive Maschinenfunktion REF)
- DB31, ... DBX 101.5 = 0 (Gantry-Verbund ist synchronisiert)
- Alle Achsen des Verbundes befinden sich innerhalb des Toleranzfensters:
DB31, ... DBX 101.4 = 1 (Gantry-Synchronisationslauf startbereit)
- Im jeweiligen NC-Kanal wird keine Achse referenziert
DB21, ... DBX33.0 = 0 (Referenzieren aktiv)

Wird der Gantry-Synchronisationslauf **nicht aus dem Referenziervorgang** mit dem NST "Gantry-Synchronisationslauf starten" gestartet, so wird für die Gleichlaufachsen als Zielposition nicht die Referenzposition vorgegeben:

MD34100 \$MA_REFP_SET_POS (Referenzpunktwert/Zeitpunkt bei abstandskodiertem System)

Stattdessen wird **die aktuelle Istposition der Führungssachse** als Zielposition vorgegeben und ohne Achskopplung angefahren.

Hinweis

Das automatische Synchronisieren kann bei der Führungssachse durch das folgende NC/PLC-Nahtstellensignal verriegelt werden:

DB31, ... DBX29.5 = 1 (Kein automatischer Synchronisationslauf)

Das ist immer dann sinnvoll, wenn die Achsen noch keine Achsfreigabe haben. In diesem Fall sollte auch der Synchronisationslauf gezielt gestartet werden mit dem NC/PLC-Nahtstellensignal:

DB31, ... DBX29.4 = 1 (Gantry-Synchronisationslauf starten)

Verlust der Synchronisation

Die Synchronisation des Gantry-Verbunds geht verloren, durch:

- "Nachführen" der Gantry-Achsen
- Verlust der Referenzposition einer Gantry-Achse z. B. durch "Parken" (kein Messsystem aktiv)
- erneutes Referenzieren einer Gantry-Achse
- Auflösen des Gantry-Verbunds durch:
MD37140 \$MA_GANTRY_BREAK_UP = 0 (Gantry-Achsverbund lösen)

Die entsprechend NC/PLC-Nahtstellensignal werden zurückgesetzt:

- DB31, ... DBX60.4 bzw. DBX60.5 == 0 (Referenziert/Synchronisiert 1 bzw. 2)
- DB31, ... DBX 101.5 == 0 (Gantry-Verbund ist synchronisiert)

Geht die Synchronisation des Gantry-Verbundes während des Betriebes aufgrund einer Störung verloren, kann der Synchronisationslauf mit dem NC/PLC-Nahtstellensignal erneut gestartet werden:

DB31, ... DBX29.4 == 1 (Gantry-Synchronisationslauf starten)

Voraussetzung ist, dass für alle Achsen des Gantry-Verbunds gilt:

DB31, ... DBX60.4 bzw. DBX60.5 = 1 (Referenziert/Synchronisiert 1 bzw. 2)

In diesem Fall verfahren die Gleichlaufachsen ohne Achskopplung auf die aktuelle Istposition der Führungssachse.

Wird bei fahrendem Gantry-Verbund das Signal "Not-Halt" (DB10, DBX56.2) gesetzt und wieder rückgesetzt, und die Gantry-Achsen sind weniger als die Stillstandstoleranz der Gleichlaufachsen auseinander gedriftet, werden diese automatisch wieder synchronisiert. Das automatische Synchronisieren kann mit dem NC/PLC-Nahtstellensignal für die Führungssachse unterbunden werden:

DB31, ... DBX29.5 = 1 (Kein automatischer Synchronisationslauf)

Wahl des Referenzpunkts

Um beim Referenzieren der Gantry-Achsen möglichst kurze Wege zu verfahren, sollten die Referenzpunktswerte von Führungs- und Gleichlaufachsen gleich sein im Maschinendatum:

MD34100 \$MA_REFP_SET_POS (Referenzpunktwert/Zielpunkt bei abstandskodiertem System)

Distanzabweichungen zwischen der Nullmarke und dem Referenzpunkt sind achsspezifisch zu berücksichtigen über die Maschinendaten:

MD34080 \$MA_REFP_MOVE_DIST (Referenzpunktabstand)

MD34090 \$MA_REFP_MOVE_DIST_CORR (Referenzpunktverschiebung/ Absolutverschiebung)

Wahl der Richtung für Referenzieren

Die Nullmarkensuchrichtung der Gleichlaufachse kann festgelegt werden über das Maschinendatum:

MD37150 \$MA_GANTRY_FUNCTION_MASK, Bit 1

Bit	Wert	Bedeutung
1	0	Nullmarkensuchrichtung der Gleichlaufachse analog zu Maschinendatum: MD34010 \$MA_REFP_CAM_DIR_IS_MINUS
	1	Nullmarkensuchrichtung der Gleichlaufachse gleich wie Führungssachse

Innerhalb des Referenziervorgangs wird bei der Synchronisations-Ausgleichsbewegung für alle Achsen des Gantry-Verbunds als Zielposition der Referenzpunktwert der Führungssachse vorgegeben und ohne Achskopplung verfahren. Bei Absolutgebern und abstandscodierten Gebern der Führungssachse wird wahlweise auf die momentane Istlage der Führungssachse oder auf den Referenzpunktwert gefahren, eingestellt durch das Maschinendatum:

MD34330 \$MA_REFP_STOP_AT_ABS_MARKER (Abstandscodiertes Längenmesssystem ohne Zielpunkt)

Falls für die Führungs- und Gleichlaufachsen nur ein Referenznocken verwendet wird, muss dies vom PLC-Anwenderprogramm berücksichtigt werden.

3.3.2 Automatisches Synchronisieren

Automatisches Synchronisieren kann erfolgen:

- im Referenziermodus (siehe Kapitel "Einführung (Seite 162)")
- in anderen Modi, wie im Folgenden beschrieben:

Wird ein Gantry-Verbund ins Nachführen geschaltet, so ist die Überwachung der Istwerte zwischen Führungssachse und Gleichlaufachsen nicht aktiv. Daher ist der Verbund nicht mehr synchron. Unabhängig von den Positionen der Achsen wird das folgende Nahtstellensignal auf 0 gesetzt (von Führungssachse)

DB31, ... DBX101.5 (Gantry-Verbund ist synchron)

Wird der Gantry-Verbund aus dem Nachführen kommend in Lageregelung geschaltet, so wird die Synchronität automatisch wiederhergestellt werden, falls bei der Überwachung der Istwerte zwischen den Positionen von Führungssachse und Gleichlaufachse(n) eine kleinere Differenz festgestellt wird als im Maschinendatum:

MD36030 \$MA_STANDSTILL_POS_TOL (Stillstandstoleranz)

In diesem Fall wird den Gleichlaufachse(n) ohne Interpolation ein neuer Sollwert vorgegeben. Die vorher festgestellte Differenz wird dann vom Lageregler ausgeregelt. Dabei bewegen sich nur die Gleichlaufachse(n).

Der Bewegungsablauf der Gleichlaufachse(n) ist analog zu der Situation, wenn aus dem Zustand "Halten" kommend die Lageregelung eingeschaltet wird. In diesem Fall wird die vor dem Halten gegebene Position vom Lageregler wieder eingestellt, falls nicht zwischenzeitlich von der Stillstandsüberwachung ein Alarm 25040 (mit Nachführen als Alarmreaktion) ausgelöst worden ist.

Bei dieser Art des automatischen Synchronisierens wird das gleiche Toleranzfenster verwendet wie bei der Stillstandsüberwachung:

MD36030 \$MA_STANDSTILL_POS_TOL (Stillstandstoleranz)

Parametersatzabhängig beaufschlagt mit dem Maschinendatum:

MD36012 \$MA_STOP_LIMIT_FACTOR (Faktor Genauhalt grob/fein und Stillstand)

Hinweis

Das folgende Nahtstellensignal blockiert das automatische Synchronisieren in jeder Betriebsart, außer im Referenziermodus:

DB31, ... DBX29.5 (Kein automatischer Synchronisationslauf)

Soll das automatische Synchronisieren hier aktiviert werden, so ist das folgende Nahtstellensignal vorzugeben:

DB31, ... DBX29.5 = 0 (Kein automatischer Synchronisationslauf)

Danach ist eine der Achsen im Gantry-Verband aus dem Nachführbetrieb in den lagegeregelten Betrieb zu schalten. Dies wird erreicht mit den Nahtstellensignalen:

DB31, ... DBX1.4 = 1 (Nachführbetrieb)

DB31, ... DBX2.1 = 1 (Reglerfreigabe)

3.3.3 Besonderheiten

2. Lagemesssysteme je Gantry-Achse

An den Gantry-Achsen eines Verbundes können unterschiedliche Lagemesssysteme angebaut sein. Des Weiteren kann jede Gantry-Achse auch zwei Lagemesssysteme verarbeiten, zwischen denen jederzeit umgeschaltet werden kann:

DB31, ... DBX1.5 (Lagemesssystem 1)

DB31, ... DBX1.6 (Lagemesssystem 2)

Die maximale Toleranz bei Lageistwertumschaltung sollte kleiner als die Gantry-Warngrenze gewählt werden:

MD36500 \$MA_ENC_CHANGE_TOL (Toleranz bei Lageistwertumschaltung)

Zuvor müssen allerdings die beiden Lagemesssysteme referenziert worden sein. Vor dem Referenzvorgang ist das jeweilige Messsystem anzuwählen. Der Ablauf ist analog wie oben beschrieben.

Kanalspezifisches Referenzieren

Gantry-Achsen können mit dem folgenden Nahtstellensignal auch kanalspezifisch referenziert werden:

DB21, ... DBX1.0 (Referenzieren aktivieren)

Für die Achsreihenfolge beim kanalspezifischen Referenzieren wird der Wert des axialen Maschinendatums der Führungssachse verwendet:

MD34110 \$MA_REFP_CYCLE_NR (Achsreihenfolge beim kanalspezifischen Referenzieren)

Nach Erreichen des Referenzpunktes der Führungssachse werden wie oben beschrieben zunächst die Gleichlaufachsen referenziert.

Referenzieren vom Teileprogramm aus mit G74

Der Referenzier- und Synchronisationsvorgang der Gantry-Achsen kann auch vom Teileprogramm mit dem Befehl G74 angestoßen werden. Dabei darf nur der Achsname der Führungssachse programmiert werden. Der Ablauf ist analog wie beim achsspezifischen Referenzieren beschrieben.

Lagemesssystem mit abstandscodierten Referenzmarken

Damit beim Referenzpunktfahren nicht große Wege abgefahren werden müssen, besteht die Möglichkeit als alleiniges bzw. als 2. Messsystem ein Lagemesssystem mit abstandscodierten Referenzmarken bei Gantry-Achsen zu verwenden. Dadurch wird erreicht, dass das Messsystem bereits nach einen kurzen Verfahrweg (z. B. 20 mm) referenziert ist. Die Vorgehensweise beim Referenzieren der Gantry-Achsen ist die gleiche wie beim normalen inkrementellen Messsystem.

Literatur:

Funktionshandbuch Grundfunktionen; Referenzpunktfahren (R1)

Absolutgeber

Bei der Synchronisations-Ausgleichsbewegung verfahren auch hier alle Achsen des Gantry-Achsverbundes ohne Achskopplung auf den Referenzpunktwert der Führungssachse, festgelegt mit dem Maschinendatum:

MD34100 \$MA_REFP_SET_POS (Referenzpunktwert/Zielpunkt bei abstandskodiertem System)

Bei Absolutgebern und abstandscodierten Gebern der Führungssachse wird wahlweise auf die momentane Istlage der Führungssachse oder auf den Referenzpunktwert gefahren, eingestellt durch das Maschinendatum:

MD34330 \$MA_REFP_STOP_AT_ABS_MARKER (Abstandscodiertes Längenmesssystem ohne Zielpunkt)

Aktivierung von Achs-Kompensationen

Sowohl bei der Führungs- als auch bei den Gleichlaufachsen können Kompensationsfunktionen aktiviert werden. Die Kompensationswerte werden für jede Gantry-Achse getrennt berücksichtigt. Bei der Inbetriebnahme sind daher die Kompensationswerte für die Führungs- und die Gleichlaufachse zu bestimmen und einzugeben.

Die Kompensationen werden steuerungsintern erst wirksam, wenn die Achse referenziert bzw. der Gantry-Verbund synchronisiert ist. Dabei gilt:

Kompensationsart	Wirksamkeit wenn	PLC-Nahtstellensignal
Losekompensation	Achse referenziert ist	"Referenziert/Synchronisiert"
SSFK	Achse referenziert ist	"Referenziert/Synchronisiert"
Durchhangkompensation	Gantry-Verbund synchronisiert ist	"Gantry-Verbund ist synchronisiert"
Temperaturkompensation	Gantry-Verbund synchronisiert ist	"Gantry-Verbund ist synchronisiert"

Wenn eine aktive Kompensation die Ursache ist für eine Bewegung der Gleichlaufachse(n), wird für die Gleichlaufachse(n) ein Fahrbefehl unabhängig von der Führungssachse angezeigt.

Überwachungen wirksam

Bei Gantry-Achsen werden analog wie bei normalen NC-Achsen die nachfolgenden Überwachungen erst mit Erreichen des Referenzpunktes (NST "Referenziert/Synchronisiert") wirksam:

- Arbeitsfeldbegrenzungen
- Software-Endschalter
- Schutzbereiche

Als Überwachungsgrenzwerte werden auch bei den Gleichlaufachsen die axialen Maschinendatenwerte verwendet.

Satzsuchlauf mehrkanalig

Mit dem kanalübergreifenden Satzsuchlauf im Modus Programmtest (SERUPRO "Search-Run by Programmtest") können Gantry-Achsverbände simuliert verfahren werden.

Hinweis

Weitere Informationen zum mehrkanaligen Satzsuchlauf SERUPRO entnehmen Sie bitte:

Literatur:

Funktionshandbuch Grundfunktionen; BAG, Kanal, Programmbetrieb (K1),
Kapitel: Programmtest

3.4 Inbetriebnahme der Gantry-Achsen

Allgemeines

Bedingt durch die in der Regel bei Gantry-Achsen vorhandene Zwangskopplung zwischen Führungs- und Gleichlaufachsen muss die Inbetriebnahme des Gantry-Verbundes wie eine Achseinheit vorgenommen werden. Daher sind stets die axialen Maschinendaten für Führungs- und Gleichlaufachsen gemeinsam festzulegen und einzutragen.

Wird die Gleichlaufachse aufgrund einer zur Führungssachse schwächeren Dynamik überlastet, wird Alarm 10656 angezeigt.

Nachfolgend wird auf die Besonderheiten bei der Inbetriebnahme von Gantry-Achsen eingegangen.

Verfahrrichtung

Bei der Inbetriebnahme der Gantry-Achsen muss kontrolliert werden, ob die Drehrichtung des Motors mit der gewünschten Verfahrrichtung der Achse übereinstimmen. Korrektur sind mit folgendem axialen Maschinendatum vorzunehmen:

MD32100 \$MA_AX_MOTION_DIR (Verfahrrichtung)

Gantry-Abschaltgrenzen eingeben

Für die Überwachung der Lageistwertabweichung der Gleichlaufachse zur Istposition der Führungssachse sind die Grenzwerte für Abschaltung sowohl bei der Führungs- als auch bei der Gleichlaufachse entsprechend den Angaben des Maschinenherstellers einzugeben:

- MD37120 \$MA_GANTRY_POS_TOL_ERROR (Gantry-Abschaltgrenze)
- MD37130 \$MA_GANTRY_POS_TOL_REF (Gantry-Abschaltgrenze beim Referenzieren)

Hinweis

Anschließend muss die Steuerung aus- und wieder eingeschaltet werden, da die Gantry-Achsdefinition und die Gantry-Abschaltgrenzwerte erst mit Wiederanlauf (POWER ON) wirksam werden.

Führungs- und Störverhalten

Aufgrund des guten Führungs- und Störverhaltens der digitalen Antriebe kann auf eine Ausgleichsregelung zwischen den Gantry-Achsen verzichtet werden. Voraussetzung für einen exakten Gleichlauf der Gantry-Achsen ist allerdings, dass die Parameter für die Regelkreise von Führungssachse und Gleichlaufachse **dynamisch gleich** eingestellt werden.

Um den bestmöglichen Gleichlauf zu erzielen, müssen Führungs- und Gleichlaufachse für das **Führungsverhalten die gleiche Dynamik** aufweisen. Die axialen Regelkreise (Lage-, Drehzahl- und Stromregler) sollten jeweils **optimal** eingestellt werden, um Störungen möglichst schnell und effizient ausregeln zu können. Zur Anpassung von unterschiedlichen Achsdynamiken ohne Verlust von Regelgüte dient die **Dynamikanpassung** im Sollwertzweig.

Axiale Optimierung

Folgende Regelungsparameter sind für Führungs- und Gleichlaufachse jeweils axial optimal einzustellen:

- MD32200 \$MA_POSCTRL_GAIN (KV-Faktor)
- MD32620 \$MA_FFW_MODE (Vorsteuerungsparameter)
- MD32610 \$MA_VELO_FFW_WEIGHT (Vorsteuerfaktor für Geschwindigkeits-/ Drehzahlvorsteuerung)
- MD32650 \$MA_AX_INERTIA (Trägheit für Drehmomentvorsteuerung)
- MD32800 \$MA_EQUIV_CURRCTRL_TIME (Ersatzzeitkonstante Stromregelkreis für Vorsteuerung)
- MD32810 \$MA_EQUIV_SPEEDCTRL_TIME (Ersatzzeitkonstante Drehzahlregelkreis für Vorsteuerung)

Literatur

Funktionshandbuch Erweiterungsfunktionen; Kompensationen (K3)

Gleiche Einstellungen

Folgende Regelungsparameter müssen für Führungs- und Gleichlaufachse gleich eingestellt werden:

- MD33000 \$MA_FIPO_TYPE (Feininterpolatortyp)
- MD32400 \$MA_AX_JERK_ENABLE (Axiale Ruckbegrenzung)
- MD32410 \$MA_AX_JERK_TIME (Zeitkonstante für den axialen Ruckfilter)
- MD32420 \$MA_JOG_AND_POS_JERK_ENABLE (Grundeinstellung der axialen Ruckbegrenzung)
- MD32430 \$MA_JOG_AND_POS_MAX_JERK (Axialer Ruck)

Literatur

Funktionshandbuch Grundfunktionen; Geschwindigkeiten/Soll-/Istwertssysteme, Regelung (G2)

Dynamik-Anpassung

Die Führungssachse und die gekoppelten Gleichlaufachsen müssen für das Führungsverhalten die gleiche Dynamik aufweisen. Gleiche Dynamik heißt: die Schleppabstände sind bei gleicher Drehzahl gleich groß.

Mit der Dynamikanpassung im Sollwertzweig lässt sich eine sehr gute Angleichung des Führungsverhaltens von dynamisch unterschiedlichen Achsen (Regelkreise) erzielen. Als Zeitkonstante der Dynamikanpassung ist die Differenz der Ersatzzeitkonstanten zwischen der dynamisch "schwächsten" Achse zu der jeweiligen anderen Achse einzugeben.

Beispiel

Bei aktiver Drehzahlvorsteuerung wird die Dynamik im Wesentlichen durch die Ersatzzeitkonstanten des "langsamsten" Drehzahlregelkreises bestimmt.

Führungssachse

MD32810 \$MA_EQUIV_SPEEDCTRL_TIME [n] = 5ms (Ersatzzeitkonstante Drehzahlregelkreis für Vorsteuerung)

Gleichlaufachse

MD32810 \$MA_EQUIV_SPEEDCTRL_TIME [n] = 3ms

- Zeitkonstante der Dynamikanpassung für Gleichlaufachse:
MD32910 \$MA_DYN_MATCH_TIME [n] = 5ms - 3ms = 2ms (Zeitkonstante der Dynamikanpassung)

Die Dynamikanpassung muss aktiviert werden axial mit dem Maschinendatum:

MD32900 \$MA_DYN_MATCH_ENABLE (Dynamikanpassung)

Kontrolle der Dynamikanpassung:

Die Schleppabstände von Führungs- und Gleichlaufachse müssen bei gleicher Geschwindigkeit gleich groß sein!

Zur Feinabstimmung kann es notwendig sein, K_V -Faktoren oder Vorsteuerparameter geringfügig anzugleichen, um ein optimales Ergebnis zu erzielen.

Gantry-Achsen referenzieren

Zunächst sind die Positionen der Referenzpunkte von Führungs- und Gleichlaufachsen näherungsweise identisch einzustellen.

Damit die Synchronisations-Ausgleichsbewegung der Gantry-Achsen nicht automatisch gestartet wird, ist bei der Erstinbetriebnahme vor dem Referenzieren die Gantry-Warngrenze auf den Wert 0 zu setzen:

MD37100 \$MA_GANTRY_POS_TOL_WARNING (Gantry-Achsdefiniton)

Dadurch ist während der Verfahrbewegung die Warnmeldung unwirksam.

Falls die momentane Schiefelage zwischen Führungs- und Gleichlaufachsen ein zu hohes zusätzliches Moment auf die Antriebe bewirkt, muss der Gantry-Verbund vor der

Verfahrensbewegung ausgerichtet werden. Danach sind die Gantry-Achsen zu referenzieren. Siehe dazu:

- Kapitel "Referenzieren und Synchronisieren von Gantry-Achsen (Seite 162)"
- **Literatur:** Funktionshandbuch Grundfunktionen; Referenzpunktfahren (R1)

Nachdem Führungs- und Gleichlaufachsen referenziert sind, ist die Differenz zwischen diesen durch Vergleich der Lageistwert zu ermitteln (HMI: Bedienbereich "Diagnose" > "Service Achsen") und als Referenzpunktverschiebung zu berücksichtigen:

- MD34080 \$MA_REFP_MOVE_DIST (Referenzpunktabstand)
- MD34090 \$MA_REFP_MOVE_DIST_CORR (Referenzpunktverschiebung/
Absolutverschiebung)

Des Weiteren sind für jede Gantry-Achse die Distanzabweichungen zwischen der Nullmarke und dem Referenzpunkt zu ermitteln. Sie sind über die folgenden Maschinendaten so anzupassen, dass die Lageistwerte von Führungs- und Gleichlaufachsen nach der Ausgleichsbewegung identisch sind:

- MD34080 \$MA_REFP_MOVE_DIST (Referenzpunktabstand)
- MD34090 \$MA_REFP_MOVE_DIST_CORR (Referenzpunktverschiebung/
Absolutverschiebung)

Gantry-Achsen synchronisieren

Der Gantry-Synchronisationslauf ist über das NC/PLC-Nahtstellensignal zu aktivieren (siehe Kapitel "Referenzieren und Synchronisieren der Gantry-Achsen (Seite 160)"):

DB31, ... DBX29.4 = 1 (Start Synchronisation Gantry)

Die Beendigung des Synchronisationsvorgang wird angezeigt über das NC/PLC-Nahtstellensignal:

DB31, ... DBX101.5 == 1 (Gantry-Verbund ist synchron)

Nach Beendigung des Synchronisationsvorganges ist zu kontrollieren, dass der Maßversatz zwischen Führungs- und Gleichlaufachsen gleich 0 ist. Gegebenenfalls sind Korrekturen in den oben genannten Maschinendaten vorzunehmen.

Gantry-Warngrenze eingeben

Nachdem die Referenzpunktwerte für Führungs- und Gleichlaufachsen optimal eingestellt sind, so dass die Gantry-Achsen nach dem Synchronisationsvorgang ohne Schiefelage gegenüberstehen, ist für Führungs- und Gleichlaufachse der Warngrenzwert im folgenden Maschinendatum einzugeben:

MD37110 \$MA_GANTRY_POS_TOL_WARNING (Gantry-Warngrenze)

Dabei ist der Wert stufenweise so lange zu erhöhen, dass gerade keine Meldung (Grenze überschritten) erscheint. Vor allem sind die Beschleunigungsphasen zu kontrollieren.

Mit diesem Grenzwert wird auch festgelegt, bei welcher Lageabweichung der Gantry-Synchronisationslauf steuerungsintern automatisch gestartet wird.

Kompensationen ermitteln und aktivieren

Falls bei den Gantry-Achsen Kompensationen (Lose-, Durchhang-, Temperatur- oder Spindelsteigungsfehler-Kompensation) erforderlich sind, müssen die Korrekturwerte für Führungs- **und** Gleichlaufachse ermittelt und in die entsprechenden Parameter bzw. Tabellen eingegeben werden.

Literatur

Funktionshandbuch Erweiterungsfunktionen; Kompensationen (K3)

Funktionsgenerator/Messfunktion

Die Aktivierung von Funktionsgenerator und Messfunktion bei einer Gleichlaufachse wird mit Fehlermeldung abgebrochen. Ist ein Anregen der Gleichlaufachse unbedingt erforderlich, z. B. um die Maschine zu vermessen, müssen Führungs- und Gleichlaufachse vorübergehend getauscht werden.

Sonderfälle

Wenn eine Anregung **einzelner** Achsen benötigt wird, muss der Gantry-Verbund vorübergehend aufgelöst werden. Da dann die andere Achse nicht mehr synchron mitfährt, muss sichergestellt werden, dass die angeregte Achse nicht mehr als die Positionstoleranz bewegt wird.

Das bedeutet bei aufgelöstem Gantry-Verbund:

- Verfahrbereichsgrenzen immer aktivieren und sehr knapp einstellen (Positionstoleranz)
- Möglichst zuerst den Gantry-Verbund synchronisieren und dann einen POWER-ON-RESET machen, **ohne** nochmals zu referenzieren. Dadurch beziehen sich die Verfahrbereichsgrenzen immer auf die gleiche Position (die beim POWER ON).
- Sprungfunktion meiden. Nur Lagesprünge sind zulässig, wenn sie kleiner als die Toleranz bleiben.
- Bei Funktionsgenerator und Messfunktion immer Offset 0 verwenden, entgegen den Empfehlungen für gewöhnliche Achsen.
- Amplituden für Funktionsgenerator und Messfunktion so klein wählen, dass die angeregte Achse weniger als die Positionstoleranz verfährt. Zur Kontrolle immer die Verfahrbereichsgrenzen aktivieren (s. o.).

Literatur

Funktionshandbuch Antriebsfunktionen; Drehzahlregelkreis (DD2)

Inbetriebnahme-Unterstützung von Gantry-Verbänden

Die Inbetriebnahmefunktionen Funktionsgenerator und Messen werden über PI-Dienste parametrisiert. Die Verfahrbewegung startet für alle parametrisierten Achsen mit NC-Start in der Betriebsart JOG.

In der Bedienoberfläche "Messfunktion und Funktionsgenerator im Gantry-Verbund" blendet die Bedienoberfläche ein Bild auf, in dem zwei Amplitudenwerte und je ein Offset und eine

Bandbreite einzugeben sind. Der erste Amplitudenwert gilt für die Messachse, der zweite für die restlichen gekoppelten Achsen.

3.5 Parametrierung: Verhalten im Fehlerfall

Impulslöschung

Das Verhalten des Gantry-Verbundes bezüglich Fehlern die Impulslöschung auslösen, kann über folgende achsspezifische Maschinendaten eingestellt werden:

MD30455 \$MA_MISC_FUNCTION_MASK, Bit 9 = <Wert>

<Wert>	Bedeutung
0	Beim Auftreten eines Fehlers der Impulslöschung auslöst (z.B. Messkreisfehler), werden die Impulse auch in allen anderen Achsen des Gantry-Verbundes gelöscht. Folge: Austrudeln aller Achsen des Gantry-Verbundes.
1	Beim Auftreten eines Fehlers der Impulslöschung auslöst (z.B. Messkreisfehler), werden die Impulse nur für diese Achse gelöscht. Die anderen Achsen des Gantry-Verbundes werden stillgesetzt. Danach werden auch bei diesen Achsen die Impulse gelöscht. Damit die Impulse dieser Achsen nicht vorzeitig gelöscht werden, sind, speziell bei Achsen die in der Stillstandsposition gehalten werden sollen (hängende Achsen), folgende Antriebsparameter zu berücksichtigen: <ul style="list-style-type: none"> • p1135[0...n] AUS3 Rücklaufzeit • p1217 Motorhaltebremse Schließzeit • p1227 Stillstandserkennung Überwachungszeit • p1228 Impulslöschung Verzögerungszeit Die ausführliche Beschreibung der Antriebsparameter findet sich in: Literatur: SINAMICS S120/S150 Listenhandbuch

3.6 PLC-Nahtstellensignale bei Gantry-Achsen

Spezielle Nahtstellensignale für Gantry-Achsen

Die speziellen NC/PLC-Nahtstellensignale der gekoppelten Gantry-Achsen werden über die axiale NC/PLC-Nahtstelle der Führungs- oder Gleichlaufachse geführt. In der nachfolgenden Tabelle sind alle speziellen Gantry- NC/PLC-Nahtstellensignale sowie die Kennzeichnung, ob das NST bei der Führungs- oder Gleichlaufachse ausgewertet wird, dargestellt.

NC/PLC-Nahtstellensignal	Übertragungsrichtung	DB31, ... DBX...	Führungsachse	Gleichlaufachse
Gantry-Synchronisationslauf starten	PLC → NC	29.4	X	
Kein automatischer Synchronisationslauf	PLC → NC	29.5	X	

NC/PLC-Nahtstellensignal	Übertragungsrichtung	DB31, ... DBX...	Führungssachse	Gleichlaufachse
Gantry-Achse	NC → PLC	101.7	1	1
Gantry-Führungssachse	NC → PLC	101.6	1	0
Gantry-Verbund ist synchronisiert	NC → PLC	101.5	X	
Gantry-Synchronisationslauf startbereit	NC → PLC	101.4	X	
Gantry-Warngrenze überschritten	NC → PLC	101.3		X
Gantry-Abschaltgrenze überschritten	NC → PLC	101.2		X

Wirkung der axialen Nahtstellensignale bei Gantry-Achsen

a) axiale Nahtstellensignale von PLC an Achse (PLC → NC)

Grundsätzlich werden die axialen Nahtstellensignale von PLC an Achse immer auf alle Gantry-Achsen des Gantry-Verbundes bezogen. Dabei ist jede Gantry-Achse (Führungs- und Gleichlaufachse) gleichberechtigt.

Es werden beispielsweise alle Achsen des Gantry-Verbundes zum gleichen Zeitpunkt stillgesetzt, wenn von der Führungssachse das folgende Nahtstellensignal auf "0" gesetzt wird:

DB31, ... DBX2.1 (Reglerfreigabe)

In der nachfolgenden Tabelle ist die Wirkung einzelner NST (von PLC an Achse) bei Gantry-Achsen dargestellt:

NC/PLC-Nahtstellensignal	DB31, ... DBX ...	Wirksamkeit bei	
		Führungssachse	Gleichlaufachse
Achsen-/Spindelsperre	1.3	auf alle Achsen des Gantry-Verbundes	ohne Wirkung
Lagemesssystem 1/2	1.5 und 1.6	axial ¹⁾	axial ¹⁾
Reglerfreigabe	2.1	auf alle Achsen des Gantry-Verbundes ²⁾	
Restweg löschen (axial)	2.2	axial	ohne Wirkung
Klemmvorgang läuft	2.3	axial	axial
Referenzpunktwert 1 - 4	2.4 - 2.7	axial	axial
Vorschub Halt	4.3	auf alle Achsen des Gantry-Verbundes	
Hardware-Endschalter plus / minus	12.0 und 12.1	Alarm axial; Bremsanforderung auf alle Achsen des Gantry-Verbundes	
2. Software-Endschalter plus / minus	12.2 und 12.3	axial	axial
Hochlaufgeber-Schnellstopp (HLGSS)	20.1	auf alle Achsen des Gantry-Verbundes	
Anwahl Antriebsparametersatz	21.0 - 21.2	axial	axial
Impulsfreigabe	21.7	axial	axial
¹⁾ DB31, ... DBX1.5 und 1.6 (Lagemesssystem 1/2) Die Umschaltung zwischen den Lagemess-Systemen 1 und 2 wird bei jeder Gantry-Achse einzeln berücksichtigt. Jedoch wird ein Inaktivschalten beider Lagemess-Systeme (sog. Parkstellung) bei allen Gantry-Achsen gemeinsam berücksichtigt. ²⁾ DB31, ... DBX2.1 (Reglerfreigabe) Bei Wegnahme der Reglerfreigabe einer Gantry-Achse werden alle Achsen des Gantry-Verbundes gleichzeitig stillgesetzt. Die Art des Stillsetzens (z. B. mit Schnellstopp) ist identisch für alle Gantry-Achsen.			

Für alle Gantry-Achsen wird entweder der Zustand "Nachführen" (Nahtstellensignal einer Gantry-Achse = 1) oder "Halten" (Nahtstellensignale aller Gantry-Achsen = 0) aktiviert abhängig vom Nahtstellensignal:

DB31, ... DBX1.4 (Nachführbetrieb)

b) axiale Nahtstellensignale von Achse an PLC (NC → PLC)

Grundsätzlich werden die axialen Nahtstellensignale von Achse an PLC für die Gleichlaufachse und die Führungssachse jeweils achsspezifisch gesetzt und an die PLC ausgegeben.

Beispiel:

DB31, ... DBX60.4 bzw. 60.5 (Referenziert/Synchronisiert 1/2).

Ausnahme:

Beim Verfahren der Führungssachse werden auch bei der Gleichlaufachse die Nahtstellensignale gesetzt:

DB31, ... DBX64.6 und 64.7 (Fahrbefehl plus bzw. minus)

3.7 Sonstiges bei Gantry-Achsen

manuelles Verfahren

Eine Gleichlaufachse kann nicht direkt von Hand in der Betriebsart JOG verfahren werden. Bei Betätigung der Verfahrtasten der Gleichlaufachse werden diese steuerungsintern ignoriert. Ebenso bleibt ein Verdrehen des Handrades bei der Gleichlaufachse ohne Wirkung.

Handradüberlagerung

Eine überlagerte Bewegung durch Handrad kann im gekoppelten Achsbetrieb nur auf die Führungssachse angewendet werden. Die Gleichlaufachsen fahren dabei synchron mit.

DRF-Verschiebung

Eine DRF-Verschiebung kann nur auf die Führungssachse angewendet werden. Die Gleichlaufachsen fahren dabei synchron mit.

Programmierung im Teileprogramm

Im Teileprogramm darf nur die Führungssachse eines Gantry-Achverbundes programmiert werden. Bei Programmierung einer Gleichlaufachse wird ein Alarm erzeugt, auch bei gelöstem Gantry-Achverbund (MD37140 \$MA_GANTRY_BREAK_UP = 1).

PLC- oder Kommandoachsen

Nur die Führungssachse des Gantry-Verbundes kann von PLC mittels FC 18 oder als Kommandoachsen durch Synchronaktionen gefahren werden.

Literatur:

- Funktionshandbuch Grundfunktionen, PLC-Grundprogramm (P3)
- Funktionshandbuch Synchronaktionen

PRESET

Die PRESET-Funktion kann nur auf die Führungssachse angewendet werden. Steuerungsintern werden mit PRESET alle Achsen des Gantry-Verbundes neu bewertet. Die Gantry-Achsen verlieren damit ihre Referenz und auch die Synchronisation:

DB31, ... DBX101.5 (Gantry-Verbund ist synchron) = 0

Kanalzuordnung der Gantry-Achsen

Es ist zu beachten, dass für einen Gantry-Verbund, dessen Führungssachse in mehreren Kanälen bekannt ist, auch dessen Gleichlaufachsen in diesen Kanälen bekannt sind. Ist dies nicht der Fall, wird der Alarm 10651 mit dem Beanstandungsgrund 60XX ausgegeben (XX ist der beanstandete Gantry-Verbund).

Achstausch

Mit RELEASE (Führungssachse) werden automatisch alle Achsen des Gantry-Verbundes frei gegeben.

Ein Achstausch der Führungssachse eines geschlossenen Gantry-Verbundes ist nur möglich, wenn im aufnehmenden Kanal alle Achsen des Verbundes bekannt sind, andernfalls wird der Alarm 10658 gemeldet.

Bei dem Versuch, einen mit MD37140 \$MA_GANTRY_BREAK_UP = 1 gelösten Gantry-Verbund wieder zu schließen, wird kein automatischer Achstausch und keine automatische Anpassung der Achszustände der Gantry-Achsen vorgenommen. Der Anwender ist selbst dafür verantwortlich. Es erfolgt allerdings eine Überprüfung der Achszustände nach Break-Up und bei Bedarf ein entsprechender Alarm 10658.

Hinweis

Soll ein Gantry-Verbund wieder geschlossen werden, so hat der Anwender dafür zu sorgen, dass alle Achsen des Verbundes mit einem entsprechenden Achszustand in einem Kanal sind.

Voreinstellung bei RESET

Bei aktivem Gantry-Verbund wird für die Gleichlaufachsen folgende Maschinendatenparametrierung ignoriert:

MD30450 \$MA_IS_CONCURRENT_POS_AX = 1 (Voreinstellung bei Reset: neutrale Achse/ Kanalachse)

Es wird der Zustand der Führungssachse angenommen. Der Anwender wird durch einen Anzeigearm 4300 auf die nicht sinnvolle Projektierung hingewiesen.

Anzeigedaten

Bei der Lageistwert-Anzeige werden die Istwerte der Führungssachse als auch die der Gleichlaufachsen angezeigt. Entsprechendes gilt auch für die Service-Anzeigewerte im Bedienbereich "Diagnose".

Software-Endschalter

Die Software-Endschalterüberwachung wird nur für die Führungssachse bearbeitet. Beim Überschreiten werden alle Achsen des Gantry-Verbundes abgebremst.

Unterschiede zur Funktion "Mitschleppen"

Folgende wesentliche Unterschiede sind zwischen den Funktionen "Gantry-Achsen" und "Mitschleppen" vorhanden:

- Die Achskopplung der Gantry-Achsen muss stets bestehen. Eine Auftrennung der Achskopplung per Teileprogramm ist somit bei Gantry-Achsen nicht möglich. Dagegen kann der Mitschleppverbund per Teileprogramm aufgelöst und die Achsen getrennt verfahren werden.
- Bei der Funktion "Gantry-Achsen" wird die Differenz der Lageistwerte von Führungs- und Gleichlaufachse stets überwacht und bei unzulässigen Abweichungen die Verfahrbewegung stillgesetzt. Bei der Funktion "Mitschleppen" erfolgt keine Überwachung.
- Gantry-Achsen müssen auch beim Referenzieren gekoppelt bleiben. Deshalb existieren spezielle Ablaufvorgänge für das Referenzpunktfahren der Gantry-Achsen. Mitschlepp-Achsen werden dagegen getrennt referenziert.
- Damit die Gantry-Achsen ohne mechanischen Versatz verfahren, müssen die Gleichlaufachsen regeldynamisch wie die Führungssachse eingestellt werden. Beim "Mitschleppen" sind dagegen auch Achsen mit unterschiedlicher Regeldynamik möglich.

Literatur:

Funktionshandbuch Sonderfunktionen, Mitschleppen (M3)

Satzsuchlauf bei aktiver Kopplung

Hinweis

Es wird empfohlen bei aktiver Kopplung für einen Satzsuchlauf ausschließlich den Suchlauftyp 5, "Satzsuchlauf über Programmtest" (SERUPRO), zu verwenden.

3.8 Beispiele

3.8.1 Gantry-Verband erstellen

Einführung

Das Einrichten eines Gantry-Verbands, das Referenzieren seiner Achsen, das Ausrichten eventueller Verschiebungen und schließlich das Synchronisieren der beteiligten Achsen ist ein aufwendigerer Vorgang. Die erforderlichen einzelnen Schritte werden im Folgenden an einer Beispielkonstellation beschrieben.

Konstellation

Maschinenachse 1 = Gantry Führungssachse, Messsystem inkrementell

Maschinenachse 3 = Gantry Gleichlaufachse, Messsystem inkrementell

Maschinendaten

Die folgenden Maschinendaten beschreiben die Ausgangswerte zu Beginn der Prozedur. Einzelne Werte sind später noch nach den unten folgenden Vorgaben zu korrigieren bzw. zu ergänzen.

Gantry Maschinendaten

Achse 1

MD37100 \$MA_GANTRY_AXIS_TYPE = 1 (Gantry-Achsdefinition)

MD37110 \$MA_GANTRY_POS_TOL_WARNING = 0 (Gantry-Warnngrenze)

MD37120 \$MA_GANTRY_POS_TOL_ERROR = z. B. 1 mm (Gantry-Abschaltgrenze)

MD37130 \$MA_GANTRY_POS_TOL_REF = z. B. 100 mm (max. Schiefelage) (Gantry-Abschaltgrenze beim Referenzieren)

MD37140 \$MA_GANTRY_BREAK_UP = 0 (Gantry-Achsverbund lösen)

Achse 3

MD37100 \$MA_GANTRY_AXIS_TYPE = 11

MD37110 \$MA_GANTRY_POS_TOL_WARNING = 0

MD37120 \$MA_GANTRY_POS_TOL_ERROR = z. B. 1mm

MD37130 \$MA_GANTRY_POS_TOL_REF = z. B. 100 mm (max. Schiefelage)

MD37140 \$MA_GANTRY_BREAK_UP = 0

Referenzpunktmaschinendaten (jeweils für den ersten Geber)

Achse 1

MD34000 \$MA_REFP_CAM_IS_ACTIVE = TRUE

MD34010 \$MA_REFP_CAM_DIR_IS_MINUS = z. B. FALSE

MD34020 \$MA_REFP_VELO_SEARCH_CAM =
MD34030 \$MA_REFP_MAX_CAM_DIST = entspricht max. Verfahrstrecke
MD34040 \$MA_REFP_VELO_SEARCH_MARKER =
MD34050 \$MA_REFP_SEARCH_MARKER_REVERSE = z. B. FALSE
MD34060 \$MA_REFP_MAX_MARKER_DIST = Differenz zw. Nockenflanke und 0-Marke
MD34070 \$MA_REFP_VELO_POS =
MD34080 \$MA_REFP_MOVE_DIST = 0
MD34090 \$MA_REFP_MOVE_DIST_CORR = 0
MD34092 \$MA_REFP_CAM_SHIFT = 0
MD34100 \$MA_REFP_SET_POS = 0
MD34200 \$MA_ENC_REFP_MODE = 1

Die Referenzpunktmaschinendaten (für den ersten Geber) der Achse 3 sind sinngemäß anzugeben.

3.8.2 Einstellung der NC-PLC Nahtstelle

Einführung

Ein automatischer Synchronisationslauf beim Referenzieren der Achsen muss zunächst unterdrückt werden, um zu vermeiden, dass ein ggf. unausgerichteter Verband zerstört wird.

Verhinderung automatischer Synchronisation

Das PLC-Anwenderprogramm setzt Folgendes für den Achsdatenbaustein der Achse 1:

DB31, ... DBX29.4 = 0 (Gantry-Synchronisation nicht starten)

DB31, ... DBX29.5 = 1 (Kein automatischer Synchronisationslauf)

Der NC setzt als Bestätigung im Achsbaustein der Achse 1:

DB31, ... DBB101.4 = 0 (Synchronisationslauf nicht startbereit)

DB31, ... DBB101.6 = 1 (Führungssachse LA)

DB31, ... DBB101.7 = 1 (Gantry-Achse)

Das PLC-Anwenderprogramm setzt für den Achsdatenbaustein der Achse 3:

DB31, ... DBX29.4 = 0 (Gantry-Synchronisation nicht starten)

Der NC setzt als Bestätigung im Achsbaustein der Achse 3:

DB31, ... DBB101.4 = 0 (Synchronisationslauf nicht startbereit)

DB31, ... DBB101.6 = 0 (Gleichlaufachse GA)

DB31, ... DBB101.7 = 1 (Gantry-Achse)

3.8.3 Beginn der Inbetriebnahme

Referenzieren

Folgende Schritte sind auszuführen:

- Betriebsart "REF" anwählen
- Referenzieren für Achse 1 (Masterachse) starten
- Warten bis Meldung "10654 Kanal 1 Warte auf Synchronisationsstart" erscheint.

Zu diesem Zeitpunkt hat der NC die Synchronisationsbereitschaft für Achse 1 hergestellt und meldet dies am Nahtstellensignal:

DB31 DB31, ... DBB101.4 = 1 (Synchronisationslauf startbereit)

DB31, ... DBB101.6 = 1 (Führungssachse LA)

DB31, ... DBB101.7 = 1 (Gantry-Achse)

Im Weiteren sind folgende Schritte auszuführen:

- RESET
- Ablesen der Werte im Maschinenkoordinatensystem:
z. B.
X = 0.941
Y = 0.000
XF = 0.000
- Tragen Sie den X-Wert der Masterachse 1 mit umgekehrtem Vorzeichen in das Maschinendatum der Folgeachse 3 ein:
MD34090 \$MA_REFP_MOVE_DIST_CORR = - 0.941 (Referenzpunktverschiebung/
Absolutverschiebung)

Hinweis

Dieses Maschinendatum ist POWER-On wirksam. Um vorab einen POWER-On zu vermeiden, kann dieser Wert auch in das folgende Maschinendatum eingetragen werden:

MD34080 \$MA_REFP_MOVE_DIST (Referenzpunktabstand)

Dieses Maschinendatum ist nach RESET gültig.

- Erneut Referenzieren für Achse 1 (Masterachse) mit den modifizierten Maschinendaten starten

- Warten, bis Meldung "10654 Kanal 1 Warte auf Synchronisationsstart" erscheint
- Zu diesem Zeitpunkt hat der NC die Synchronisationsbereitschaft für Achse 1 hergestellt und meldet dies am Nahtstellensignal:
DB31 DB31, ... DBB101.4 = 1 (Synchronisationslauf startbereit)
DB31, ... DBB101.6 = 1 (Führungssachse LA)
DB31, ... DBB101.7 = 1 (Gantry-Achse)
- Istpositionen der Maschine betrachten. Es können die Fälle A oder B vorliegen:

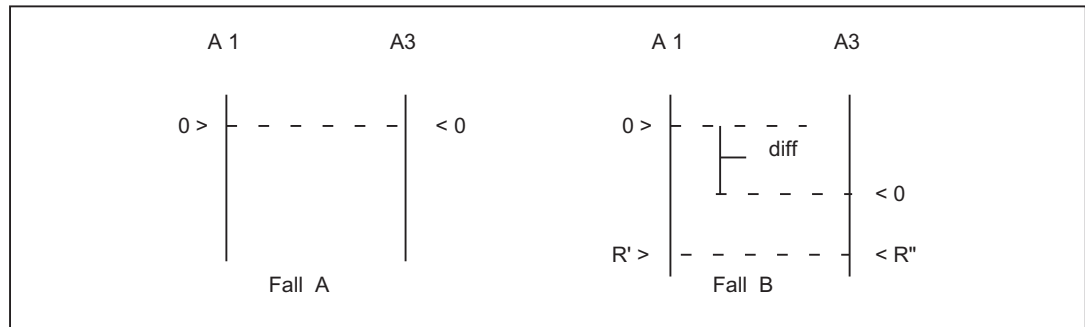


Bild 3-3 Mögliche Ergebnisse nach Referenzieren Achse 1 (Masterachse)

Liegt Fall A vor, kann sofort der Synchronisationslauf gestartet werden (siehe Schritt "Synchronlauf starten"). Liegt Fall B vor, muss der Versatz "diff" ermittelt und berücksichtigt werden:

- Messen von diff
- Durch zwei rechtwinklig angebrachte Referenzpunkte R' und R'' im Maschinenbett (rechts im Bild) kann die Positionsdivergenz im JOG abgefahren werden. Der Versatz diff kann dann als Unterschied der Positionsanzeige abgelesen werden. Der Versatz diff muss eingetragen werden in das Maschinendatum der Achse 3 (Gleichlaufachse):
MD34100 \$MA_REFP_SET_POS
Weiter mit 1. Schritt (siehe oben).
- Synchronisationslauf starten. PLC setzt:
DB31, ... DBX29.4= 1 (Start Synchronisation Gantry)

3.8.4 Warn- und Fehlergrenzen einstellen

Ist der Gantry-Verbund eingestellt und synchronisiert, müssen abschließend noch folgende Maschinendaten eingestellt werden:

MD37110 \$MA_GANTRY_POS_TOL_WARNING (Gantry-Warngrenze)

MD37120 \$MA_GANTRY_POS_TOL_ERROR (Gantry-Abschaltgrenze)

Vorgehensweise

- Stellen Sie das Maschinendatum für alle Achsen zunächst groß ein:
MD37120 \$MA_GANTRY_POS_TOL_ERROR (Gantry-Abschaltgrenze)
- Belegen Sie mit einem sehr kleinen Wert das Maschinendatum:
MD37110 \$MA_GANTRY_POS_TOL_WARNING (Gantry-Warngrenze)
Wenn Sie jetzt die Achsen dynamisch stark belasten, wird immer wieder der selbstlöschende Alarm "10652 Kanal %1 Achse %2 Gantry-Warngrenze überschritten" ausgegeben.
- Erhöhen Sie jetzt das folgende Maschinendatum:
MD37110 \$MA_GANTRY_POS_TOL_WARNING (Gantry-Warngrenze)
Wiederholen Sie dies solange, bis der Alarm nicht mehr erscheint. Die Nahtstelle zeigt den Zustand:
DB31, ... DBX101.2 = 0 (Gantry-Abschaltgrenze nicht überschritten)
DB31, ... DBX101.3 = 0 (Gantry-Warngrenze nicht überschritten)
DB31, ... DBX101.4 = 0 (Gantry-Synchronisationslauf nicht startbereit)
DB31, ... DBX101.5 = 1 (Gantry-Verbund ist synchron)
DB31, ... DBX101.6 = 1 (Gantry-Führungsachse)
DB31, ... DBX101.7 = 1 (Gantry-Achse)

Falls die Überwachung nur noch sehr sporadisch anschlägt, ist es möglich im PLC-Anwenderprogramm einen Flankenmerker zu programmieren.

- Tragen Sie den für die Warngrenze erhaltenen Wert und einen kleinen Sicherheitsaufschlag in das folgende Maschinendatum ein:
MD37120 \$MA_GANTRY_POS_TOL_ERROR (Gantry-Abschaltgrenze)

Fehlergrenzwerte

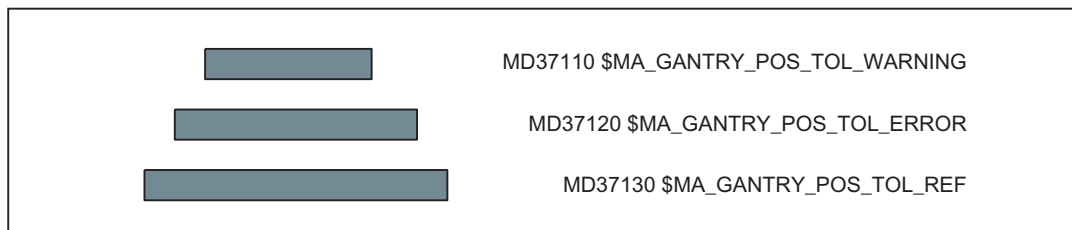
In folgenden Maschinendaten wurden Werte hinterlegt:

MD37110 \$MA_GANTRY_POS_TOL_WARNING (Gantry-Warngrenze)

MD37120 \$MA_GANTRY_POS_TOL_ERROR (Gantry-Abschaltgrenze)

MD37130 \$MA_GANTRY_POS_TOL_REF (Gantry-Abschaltgrenze beim Referenzieren)

Diese sollten am Abschluss des Anpassvorganges die folgenden Größenverhältnisse haben:



Hinweis

Bei der Inbetriebnahme eines Gantry-Verbands, bei dem die verbundenen Achsen von **Linearmotoren** und zugehörigen Messsystemen betrieben werden, ist sinngemäß zu verfahren.

Die eingegebenen Fehlergrenzen in den Maschinendaten MD37110 und MD37120 verstehen sich, wenn das Nahtstellenbit "Gantry ist synchron" nicht ansteht, als zusätzliche Toleranz der Istwertdifferenz von Leit und Folgeachse (z. B. nach dem Löschen von Alarmen ohne Gantry neu zu synchronisieren).

3.9 Datenlisten

3.9.1 Maschinendaten

3.9.1.1 Achs-/Spindel-spezifische Maschinendaten

Nummer	Bezeichner: \$MA_	Beschreibung
30300	IS_ROT_AX	Rundachse
32200	POSCTRL_GAIN	K _V -Faktor
32400	AX_JERK_ENABLE	Axiale Ruckbegrenzung
32410	AX_JERK_TIME	Zeitkonstante für axialen Ruckfilter
32420	JOG_AND_POS_JERK_ENABLE	Grundeinstellung der axialen Ruckbegrenzung
32430	JOG_AND_POS_MAX_JERK	Axialer Ruck
32610	VELO_FFW_WEIGHT	Vorsteuerfaktor für Drehzahlvorsteuerung
32620	FFW_MODE	Vorsteuerungsart
32650	AX_INERTIA	Trägheitsmoment für Drehmomentvorsteuerung
32800	EQUIV_CURRCTRL_TIME	Ersatzzeitkonst. Stromregelkreis für Vorsteuerung
32810	EQUIV_SPEEDCTRL_TIME	Ersatzzeitkonst. Drehzahlregelkreis für Vorsteuerung
32900	DYN_MATCH_ENABLE	Dynamikanpassung
32910	DYN_MATCH_TIME	Zeitkonstante für Dynamikanpassung
33000	FIPO_TYPE	Feininterpolatortyp
34040	REFP_VELO_SEARCH_MARKER	Abschaltgeschwindigkeit
34070	REFP_VELO_POS	Referenzpunkteinfahrgeschwindigkeit
34080	REFP_MOVE_DIST	Referenzpunktabstand
34090	REFP_MOVE_DIST_CORR	Referenzpunktverschiebung
34100	REFP_SET_POS	Referenzpunktwert
34110	REFP_CYCLE_NR	Achsreihenfolge beim kanalspezifischen Referenzieren

Nummer	Bezeichner: \$MA_	Beschreibung
34330	REFP_STOP_AT_ABS_MARKER	Abstandscodiertes Längenmesssystem ohne Zielpunkt
36012	STOP_LIMIT_FACTOR	Faktor Genauhalt grob/fein und Stillstand
36030	STANDSTILL_POS_TOL	Stillstandstoleranz
36500	ENC_CHANGE_TOL	Maximale Toleranz bei Lageistwertumschaltung
37100	GANTRY_AXIS_TYPE	Gantry-Achsdefinition
37110	GANTRY_POS_TOL_WARNING	Gantry-Warngrenze
37120	GANTRY_POS_TOL_ERROR	Gantry-Abschaltgrenze
37130	GANTRY_POS_TOL_REF	Gantry-Abschaltgrenze beim Referenzieren
37140	GANTRY_BREAK_UP	Gantry-Achsverbund lösen

3.9.2 Signale

3.9.2.1 Signale von BAG

Signalname	SINUMERIK 840D sl	SINUMERIK 828D
Aktive Maschinenfunktion REF	DB11.DBX5.2	DB3100.DBX1.2

3.9.2.2 Signale von Kanal

Signalname	SINUMERIK 840D sl	SINUMERIK 828D
Referenzieren aktiv	DB21, ... DBX33.0	DB330x.DBX1.0

3.9.2.3 Signale an Achse/Spindel

Signalname	SINUMERIK 840D sl	SINUMERIK 828D
Gantry-Synchronisationslauf starten	DB31, ... DBX29.4	DB380x.DBX5005.4
Kein automatischer Synchronisationslauf	DB31, ... DBX29.5	DB380x.DBX5005.5

3.9.2.4 Signale von Achse/Spindel

Signalname	SINUMERIK 840D sl	SINUMERIK 828D
Referenziert/Synchronisiert 1, Referenziert/Synchronisiert 2	DB31, ... DBX60.4/5	DB390x.DBX0.4/5
Gantry-Abschaltgrenze überschritten	DB31, ... DBX101.2	DB390x.DBX5005.2
Gantry-Warngrenze überschritten	DB31, ... DBX101.3	DB390x.DBX5005.3
Gantry-Synchronisationslauf startbereit	DB31, ... DBX101.4	DB390x.DBX5005.4
Gantry-Verbund ist synchron	DB31, ... DBX101.5	DB390x.DBX5005.5

Signalname	SINUMERIK 840D sl	SINUMERIK 828D
Gantry-Führungssachse	DB31, ... DBX101.6	DB390x.DBX5005.6
Gantry-Achse	DB31, ... DBX101.7	DB390x.DBX5005.7

K6: Konturtunnel-Überwachung

4.1 Kurzbeschreibung

4.1.1 Konturtunnel-Überwachung - nur 840D sl

Funktion

Es wird die absolute Bewegung der Werkzeugspitze im Raum überwacht. Die Funktion arbeitet kanalspezifisch.

Modell

Über der programmierten Bahn einer Bearbeitung wird ein runder Tunnel definiert, dessen Durchmesser vorgegeben werden kann. Achsbewegungen werden optional angehalten, wenn Bahnabweichungen der Werkzeugspitze durch Achsfehler größer als der definierte Tunnel werden.

Reaktion

Bei erkannter Abweichung reagiert das System schnellstmöglich. Es vergeht jedoch mindestens ein Interpolatortakt, bevor eine der beiden folgenden Reaktionen erfolgt:

- Das Verlassen des Tunnels löst einen Alarm aus und die Achsen fahren weiter.
- Das Verlassen des Tunnels löst einen Alarm aus und die Achsbewegungen werden abgebremst.

Bremsmethoden

Wenn der Überwachungstunnel verlassen wird, kann wahlweise über eine der beiden folgenden Methoden abgebremst werden:

- Bremsrampe
- Drehzahlsollwert Null und Nachführbetrieb

Einsatz

Die Funktion kann bei 2D- und 3D-Bahnen eingesetzt werden. Bei 2D ist die Überwachungsfläche durch parallele Linien zur programmierten Bahn definiert. Der Überwachungsbereich wird durch 2 oder 3 Geometrieachsen bestimmt.

Die Überwachung von Synchronachsen, Positionierachsen usw., die nicht Geometrieachsen sind, erfolgt direkt auf Maschinenachsebene mit der "Konturüberwachung".

Beispiel

Das folgende Bild zeigt schematisch an einem einfachen Beispiel die Gestalt des Überwachungsbereichs.

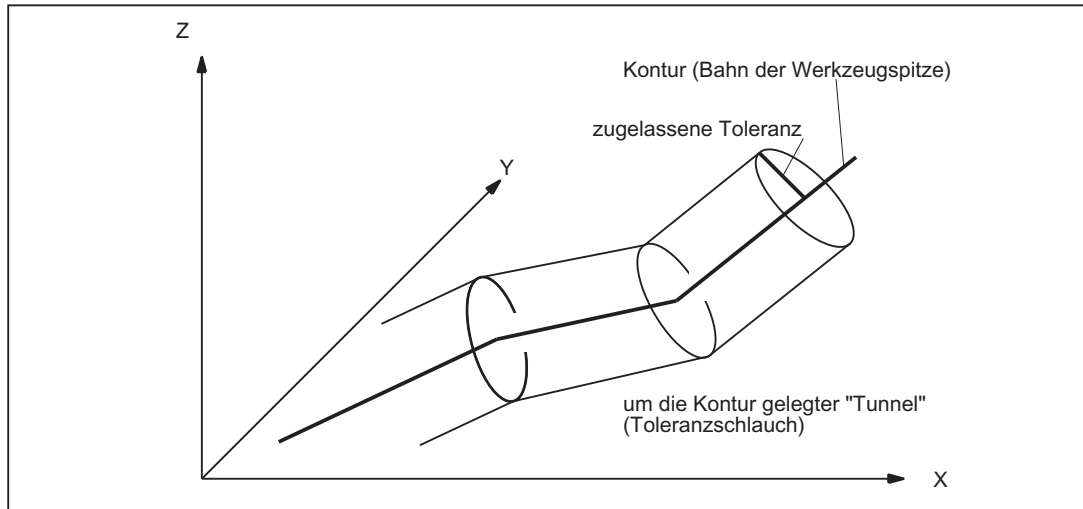


Bild 4-1 Lage des Konturtunnels um programmierte Bahn

Solange die errechnete Ist-Position der Werkzeugspitze innerhalb des skizzierten Tunnels bleibt, wird die Bewegung normal fortgesetzt. Verlässt die errechnete Ist-Position den Tunnel, wird (in der Standardeinstellung) ein Alarm ausgelöst und die Achsen werden mit "Rampenstopp" stillgesetzt. Über Maschinendaten lässt sich die Reaktion beim Verlassen des Tunnels auch abschalten (Alarm, aber Bewegung fortsetzen) oder verschärfen (Schnellstopp).

Analyse

Der errechnete Abstand zwischen programmierter Bahn und Ist-Werten kann auf einen Analogausgang gelegt werden, um im normalen Betrieb den Verlauf der Konturfehler analysieren zu können (Qualitätskontrolle).

4.1.2 Programmierbare Konturgenauigkeit

Funktion

Als Alternative zu der unter "Konturtunnel-Überwachung" beschriebenen Möglichkeit, die Bearbeitungsgenauigkeit zu überwachen und bei zu großen Abweichungen die Bearbeitung abubrechen, steht auch die Möglichkeit zur Verfügung, eine gewählte Genauigkeit in jedem Fall zu erzielen, wobei nötigenfalls die Bahngeschwindigkeit herabgesetzt wird. Die Details zu dieser Funktion finden sich unter dem Thema "Programmierbare Konturgenauigkeit".

4.2 Konturtunnel-Überwachung - nur 840D sl

Überwachungsziel

Ziel der Überwachung ist es, die Bewegung der Achsen still zu setzen, wenn wegen Achsabweichungen die Distanz zwischen Werkzeugspitze (Istwert) und der programmierten Bahn (Sollwert) einen vorgegebenen Wert (Tunnelradius) überschreitet.

Tunnelgröße

Für die Überwachungsfunktion ist die Angabe des Radius des zu überwachenden Konturtunnels um die programmierte Bahn erforderlich:

MD21050 \$MC_CONTOUR_TUNNEL_TOL (Ansprechschwelle für Konturtunnel-Überwachung)

Wird das Maschinendatum mit 0.0 besetzt, so erfolgt keine Überwachung. Der Wert des Maschinendatums wird von der Steuerung bei Neukonfiguration übernommen.

Parametrierbares Bremsverhalten

Das Bremsverhalten beim Ansprechen der Überwachung kann über das folgende Maschinendatum eingestellt werden:

MD21060 \$MC_CONTOUR_TUNNEL_REACTION (Reaktion bei Ansprechen der Konturtunnel-Überwachung)

Wert	Bedeutung
0	Alarm anzeigen und die Bearbeitung fortsetzen
1	Bremsen gemäß Bremsrampe (Standardvorbereitung)
2	Schnellstopp (Drehzahlsollwert = 0)

Geberumschaltung

Die Umschaltung zwischen zwei Gebersystemen ergibt in der Regel eine sprunghafte Veränderung der Ist-Position der Werkzeugspitze. Die Veränderung allein durch die Geberumschaltung darf nicht so groß werden, dass die Werkzeugspitze den Überwachungstunnel durchbricht. Der im MD21050 vorgegebene Radius muss größer sein als die zulässige Toleranz bei der Istwertgeberumschaltung:

MD36500 \$MA_ENC_CHANGE_TOL (Toleranz bei Lageistwertumschaltung)

Aktivierung

Die Überwachung wird nur unter folgenden Bedingungen aktiv:

- MD21050 ist größer als 0.0.
- Es sind mindestens zwei Geometrieachsen definiert.

Stillsetzen

Die Überwachung kann stillgesetzt werden durch Wirksamsetzen der Maschinendaten-Einstellung:

MD21050 = 0.0.

Analyse-Ausgang

Die Werte der Abweichung des Istwerts der Werkzeugspitze von der programmierten Bahn können zur Analyse auf einem schnellen Analogausgang ausgegeben werden (Genauigkeitsüberwachung).

Die Zuordnung eines Analogausgangs für die Ausgabe des Konturfehlers erfolgt im Maschinendatum:

MD21070 \$MC_CONTOUR_ASSIGN_FASTOUT

Wert	Bedeutung
0	keine Ausgabe (Standardvorbesetzung)
1	Ausgabe auf Ausgang 1
2	Ausgabe auf Ausgang 2
...	...
8	Ausgabe auf Ausgang 8

Maßstab

Der im MD21050 hinterlegte Tunnelradius entspricht der Spannung 10 V am Ausgang.

4.3 Programmierbare Konturgenauigkeit

Funktion

Die Funktion "Programmierbare Konturgenauigkeit" begrenzt den Konturfehler aufgrund von Regelverhalten und Ruckfilter auf einen vorgegebenen Wert, indem sie die Bahngeschwindigkeit an gekrümmten Konturen im notwendigen Maße reduziert. Sie ermöglicht dadurch dem Anwender die Einstellung eines Kompromisses zwischen Genauigkeit und Produktivität einer Bearbeitung.

Hinweis

Die Funktion "LookAhead" stellt sicher, dass an keiner Stelle der Bahn eine größere Geschwindigkeit gefahren wird, als zur Einhaltung der geforderten Konturgenauigkeit möglich ist.

Projektierung

Die Wirkungsweise und Parametrierung der Funktion wird bestimmt durch das Maschinendatum:

MD20470 \$MC_CPREC_WITH_FFWD (Programmierbare Konturgenauigkeit)

Wert	Bedeutung
0	Die Funktion "Programmierbare Konturgenauigkeit" ist bei gleichzeitig aktiver Vorsteuerung unwirksam.
1	Die Funktion "Programmierbare Konturgenauigkeit" ist auch bei Vorsteuerung wirksam. Die Absenkung der Bahngeschwindigkeit wird bei aktiver Vorsteuerung auf Grundlage des effektiven K_V -Faktors mit Vorsteuerung berechnet.
2	Wie 1, die Funktion wird aber mit MD32415 \$MA_EQUIV_CPREC_TIME (Zeitkonstante für die programmierbare Konturgenauigkeit) parametrierung. Das Ruckfilter wird korrekt eingerechnet. Das Settingdatum SD42450 \$SC_CONTPREC bestimmt den erlaubten Konturfehler (siehe "Parametrierung").
3	Wie 2, aber eine eventuell mit CTOL programmierte Konturgenauigkeit hat Vorrang vor SD42450 \$SC_CONTPREC. Das Ruckfilter wird korrekt eingerechnet. Die programmierte Konturgenauigkeit CTOL bestimmt den erlaubten Konturfehler (siehe "Parametrierung"). \$SC_CONTPREC ist nur dann relevant, wenn CTOL nicht programmiert wurde.
4	Die Funktion "Programmierbare Konturgenauigkeit" ist unabhängig von Vorsteuerung und Ruckfilter wirksam. Es wird ausschließlich MD32415 \$MA_EQUIV_CPREC_TIME in der Berechnung des Konturfehlers betrachtet. Alle Zeitkonstanten, die Einflüsse auf den Konturfehler haben, müssen aufsummiert im MD32415 \$MA_EQUIV_CPREC_TIME eingetragen werden.
5	Wie 4, aber eine eventuell mit CTOL programmierte Konturgenauigkeit hat Vorrang vor SD42450 \$SC_CONTPREC.

Bei den Funktionsvarianten MD20470 = 2 oder 3 geht die Steuerung davon aus, dass es eine Ruckfilter-Zeitkonstante (MD32410 \$MA_AX_JERK_TIME) gibt, bei der die Einstellung der Regelstrecke mit Vorsteuerung einen vernachlässigbar kleinen Konturfehler erzeugt. Dieser Wert muss in das Maschinendatum MD32415 \$MA_EQUIV_CPREC_TIME (siehe "Parametrierung") eingetragen werden.

Zur Berechnung des Konturfehlers auf Grundlage des eingestellten Ruckfiltertyps (MD32402 \$MA_AX_JERK_MODE) wird folgender Wert verwendet:

- bei aktiver Vorsteuerung die Differenz:
MD32410 \$MA_AX_JERK_TIME - MD32415 \$MA_EQUIV_CPREC_TIME
- ohne Vorsteuerung der volle Wert aus MD32410 \$MA_AX_JERK_TIME

Dieses Vorgehen erlaubt dem Inbetriebnehmer, von einer zunächst genauen, aber möglicherweise zu harten Einstellung durch eine Erhöhung der Ruckfilter-Zeitkonstanten zu einer weicheren Einstellung mit kontrolliertem Genauigkeitsverlust zu wechseln.

Soll die Funktion "Programmierbare Konturgenauigkeit" zusammen mit dem Ruckfiltertyp "FIR-Tiefpass" (MD32402 \$MA_AX_JERK_MODE = 5) verwendet werden, muss bei der Inbetriebnahme über das speicherkonfigurierende Maschinendatum MD38020 \$MA_MM_CPREC_FIR_POINTS Speicherplatz für die Kennlinienapproximation von FIR-Filtern bereitgestellt werden. Ohne spezifischen Speicherplatz (MD38020 = 0) ist die Ausführung der Funktion nicht möglich und es wird Alarm 10990 ausgegeben.

Einschränkungen:

- Die Funktion arbeitet nicht mit dem Ruckfiltertyp "Bandsperre" (MD32402 \$MA_AX_JERK_MODE = 3).
- Die Funktionsvarianten MD20470 = 2 oder 3 sind in erster Linie für die Verwendung mit Vorsteuerung gedacht. Ist eine der beiden Funktionsvarianten bei ausgeschalteter Vorsteuerung aktiv, wird ein Konturfehler addiert, der sich aus dem K_V -Faktor ergibt. Die Bahngeschwindigkeit wird dabei wesentlich stärker reduziert.

Hinweis

Die Funktionsvarianten MD20470 = 0 bzw. 1 werden nicht mehr empfohlen. Sie stellen lediglich noch die Kompatibilität zu älteren Softwareständen her.

Parametrierung

Konturgenauigkeit

Der maximale Konturfehler für die Bahn der Geometrieachsen auf gekrümmten Konturen wird bestimmt:

- bei MD20470 \$MC_CPREC_WITH_FFW = 2 durch das Settingdatum: SD42450 \$SC_CONTPREC (Konturgenauigkeit)
- bei MD20470 \$MC_CPREC_WITH_FFW = 3 durch die mit CTOL programmierte Konturtoleranz.

Je kleiner der Wert und je kleiner der K_V -Faktor der Geometrieachsen, umso stärker wird der Bahnvorschub auf gekrümmten Konturen abgesenkt.

Mindestbahnvorschub

Über das folgende Settingdatum kann der Anwender für die Funktion "Programmierbare Konturgenauigkeit" einen Mindestbahnvorschub vorgeben:

SD42460 \$SC_MINFEED (Mindestbahnvorschub bei CPRECON)

Der Vorschub wird nicht unter diesen Wert begrenzt, es sei denn, ein niedrigerer F-Wert wurde programmiert oder die dynamischen Begrenzungen der Achsen erzwingen eine niedrigere Bahngeschwindigkeit.

Zeitkonstante für die programmierbare Konturgenauigkeit

Die Ersatzzeitkonstante für die Funktionsvarianten MD20470 = 2 oder 3 (siehe "Projektierung") wird eingetragen in das Maschinendatum:

MD32415 \$MA_EQUIV_CPREC_TIME (Zeitkonstante für die programmierbare Konturgenauigkeit)

MD32415 muss diejenige Ruckfilter-Zeitkonstante (MD32410 \$MA_AX_JERK_TIME) enthalten, bei der der Konturfehler bei aktiver Vorsteuerung vernachlässigbar klein ist.

Programmierung

Die "Programmierbare Konturgenauigkeit" kann mit den modalen G-Befehlen CPRECON und CPRECOF im Teileprogramm ein- und ausgeschaltet werden.

Beispiel:

Programmcode	Kommentar
N10 G0 X0 Y0	
N20 CPRECON	; Einschalten der "Programmierbaren Konturgenauigkeit".
N30 G1 G64 X100 F10000	; Bearbeitung mit 10 m/min im Bahnsteuerbetrieb.
N40 G3 Y20 J10	; Automatische Vorschubbegrenzung im Kreissatz.
N50 G1 X0	; Vorschub wieder ohne Begrenzung (10 m/min).
...	
N100 CPRECOF	; Ausschalten der "Programmierbaren Konturgenauigkeit".
N110 G0 ...	

Die beiden modalen G-Befehle CPRECON und CPRECOF bilden die G-Gruppe 39 (Programmierbare Konturgenauigkeit).

Verhalten bei Teileprogrammstart und nach Reset/Teileprogrammende

Bei Teileprogrammstart und nach Reset/Teileprogrammende wird die für die G-Gruppe 39 projektierte Steuerungs-Grundstellung wirksam:

MD20110 \$MC_RESET_MODE_MASK (Festlegung der Steuerungs-Grundstellung nach Reset/TP-Ende)

MD20112 \$MC_START_MODE_MASK (Festlegung der Steuerungs-Grundstellung nach Teileprogrammstart)

Randbedingungen

Positionierachsen

Die Funktion betrachtet nur die Geometrieachsen der Bahn. Auf die Geschwindigkeiten von Positionierachsen hat sie keinen Einfluss.

Literatur

Informationen zu MD32402 \$MA_AX_JERK_MODE (Filtertyp für axiale Ruckbegrenzung) und MD32410 \$MA_AX_JERK_TIME (Zeitkonstante für den axialen Ruckfilter) siehe:

- Funktionshandbuch Grundfunktionen; Beschleunigung (B2), Kapitel: "Funktionen" > "Ruckfilter (Lagesollwertfilter, achsspezifisch)"

Informationen zu CTOL siehe:

- Funktionshandbuch Grundfunktionen; Bahnsteuerbetrieb, Genauhalt, LookAhead (B1), Kapitel: "Kontur-/Orientierungstoleranz"

4.4 Randbedingungen

Verfügbarkeit der Funktion "Konturtunnel-Überwachung"

Die Funktion ist eine Option ("Konturüberwachung durch Tunnelfunktion"), die über das Lizenzmanagement der Hardware zugeordnet werden muss.

Mitschleppen

Wird bei aktiver Konturtunnel-Überwachung eine Mitschleppkopplung zwischen zwei Geometrieachsen programmiert, hat dies immer ein Ansprechen der Konturtunnel-Überwachung zur Folge. Die Konturtunnel-Überwachung muss in diesem Fall vor Programmierung der Mitschleppkopplung ausgeschaltet werden:

MD21050 \$MC_CONTOUR_TUNNEL_TOL = 0.0

4.5 Datenlisten

4.5.1 Maschinendaten

4.5.1.1 Kanal-spezifische Maschinendaten

Nummer	Bezeichner: \$MC_	Beschreibung
20110	RESET_MODE_MASK	Festlegung der Steuerungs-Grundstellung nach Reset/TP-Ende
20112	START_MODE_MASK	Festlegung der Steuerungs-Grundstellung nach Teilprogrammstart
20470	CPREC_WITH_FFW	Programmierbare Konturgenauigkeit
21050	CONTOUR_TUNNEL_TOL	Ansprechschwelle für Konturtunnel-Überwachung
21060	CONTOUR_TUNNEL_REACTION	Reaktion bei Ansprechen der Konturtunnel-Überwachung
21070	CONTOUR_ASSIGN_FASTOUT	Zuordnung eines Analogausgangs für die Ausgabe des Konturfehlers

4.5.1.2 Achs-/Spindel-spezifische Maschinendaten

Nummer	Bezeichner: \$MA_	Beschreibung
32402	AX_JERK_MODE	Filtertyp für axiale Ruckbegrenzung
32410	AX_JERK_TIME	Zeitkonstante für den axialen Ruckfilter

Nummer	Bezeichner: \$MA_	Beschreibung
32415	EQUIV_CPREC_TIME	Zeitkonstante für die programmierbare Konturgenauigkeit
36500	ENC_CHANGE_TOL	Maximale Toleranz bei Lageistwertumschaltung

4.5.2 Settingdaten

4.5.2.1 Kanal-spezifische Settingdaten

Nummer	Bezeichner: \$SC_	Beschreibung
42450	CONTPREC	Konturgenauigkeit
42460	MINFEED	Mindestbahnvorschub bei CPRECON

K7: Kinematische Kette

5.1 Funktionsbeschreibung

5.1.1 Merkmale

Im vorliegenden Kapitel wird beschrieben, wie für NC-Funktionen wie "Kollisionsvermeidung" oder "Kinematische Transformation" die kinematische Struktur einer Maschine mittels einer kinematischen Kette abgebildet und in der Steuerung über Systemvariablen parametrisiert wird.

Die Systemvariablen werden in der NC remanent gespeichert und können über SINUMERIK Operate mittels Inbetriebnahme-archiv als "NC-Daten" archiviert bzw. eingelesen werden.

Für Funktionen wie z.B. die "Kollisionsvermeidung", die auch die Beschreibung der Maschinengeometrie benötigen, siehe Kapitel "K8: Geometrische Maschinenmodellierung (Seite 235)".

Hinweis

Grafischer Editor

Die Maschinenmodellierung kann, alternativ zum Schreiben der Systemvariablen in einem Teileprogramm, über die Bedienoberfläche SINUMERIK Operate erfolgen:

Bedienbereich: "Inbetriebnahme" > "NC" > "Maschinenmodell"

Änderungen am Maschinenmodell

Direkt an den Systemvariablen vorgenommene Änderungen am Maschinenmodell werden erst nach einer expliziten Anforderung zum Neuberechnen des Maschinenmodells durch Aufruf der Funktion PROTA() (Seite 305) bzw. PROTS() (Seite 306) an der Bedienoberfläche sichtbar.

Über die Bedienoberfläche vorgenommene Änderungen am Maschinenmodell werden sofort in die Systemvariablen der NC übernommen. Die Änderungen werden aber erst nach einer expliziten Anforderung zum Neuberechnen des Maschinenmodells durch Aufruf der Funktion PROTA() (Seite 305) bzw. PROTS() (Seite 306) aktiv.

Kinematische Struktur

Die kinematische Struktur einer Maschine setzt sich zusammen aus:

- Anzahl und Typ der Maschinenachsen: Linear- oder Rundachsen
- Anordnung der Maschinenachsen: Lage und Orientierung
- Abhängigkeiten der Maschinenachsen untereinander: Welche Maschinenachse verfährt mit welcher anderen Maschinenachse mit.

Kinematische Kette

Die Beschreibung der kinematischen Struktur einer Maschine erfolgt mittels einer kinematischen Kette mit folgenden Eigenschaften:

- Eine kinematische Kette besteht aus einer beliebigen Anzahl miteinander verbundener Elemente.
- Von einer kinematischen Kette können parallele Teilketten abzweigen.
- In der Steuerung liegt immer nur eine aktive kinematische Kette vor.
- Die aktive kinematische Kette beginnt mit dem Root-Element.
- Parametrierte Elemente oder Teilketten, die nicht mit dem Root-Element verbunden sind oder deren Verbindung zum Root-Element hin zeigt, sind nicht Bestandteil der aktuell wirksamen kinematischen Kette.
- Eine kinematische Kette wird in den raumfesten Koordinaten des Weltkoordinatensystems definiert.

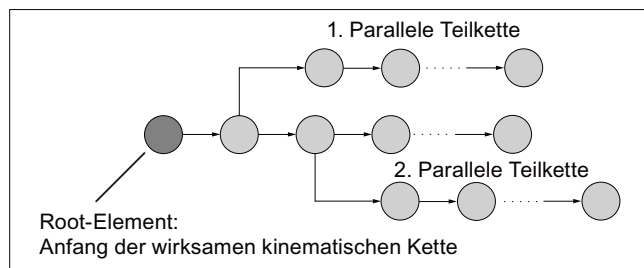


Bild 5-1 Beispiel einer kinematischen Kette

Element

Ein Element einer kinematischen Kette beschreibt im Wesentlichen die Transformation, mit der das lokale Koordinatensystem des vorhergehenden Elements in das lokale Koordinatensystem des aktuellen Elements abgebildet wird:

$$K_{n-1} \Rightarrow T_n \Rightarrow K_n$$

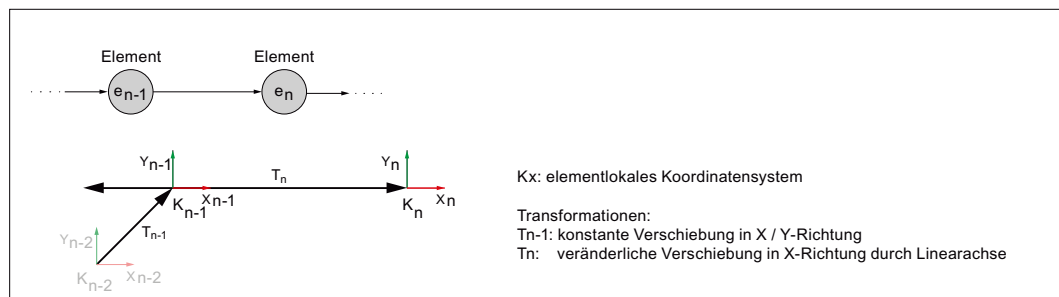


Bild 5-2 Elementlokale Koordinatensysteme

Folgende konstante Transformationen sind möglich:

- Verschiebung (Typ: OFFSET (Seite 221))
- Drehung (Typ: ROT_CONST (Seite 219))

Folgende veränderliche Transformationen, basierend auf den aktuellen Positionswerten der dem Element zugeordneten Maschinenachse (Linearachse / Rundachse), sind möglich:

- Verschiebung (Typ: AXIS_LIN (Seite 212))
- Drehung (Typ: AXIS_ROT (Seite 215))

Eine Positions- oder Orientierungsänderung in einem Element, z.B. durch Positionsänderung der zugehörigen Maschinenachse, wirkt sich auf alle **nachfolgenden** Elemente der Kette oder parallelen Teilketten aus.

Die maximale Anzahl möglicher Elemente ist über Maschinendaten (Seite 207) parametrierbar.

Parallele Teilketten

Zweigt von einem Element e_n eine parallele Teilkette ab, erfolgt das Abzweigen der Teilkette kinematisch immer **vor** dem Element. Eine Veränderung im Element e_n , z.B. eine Positionsänderung der zugehörigen Maschinenachse, wirkt sich dadurch **nicht** auf die abzweigende Teilkette aus.

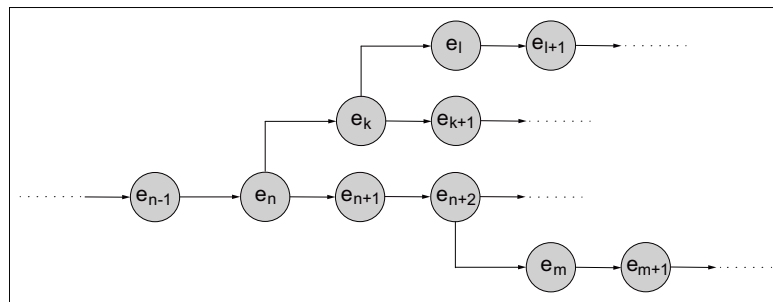


Bild 5-3 Von den Elementen e_n und e_{n+2} abzweigende parallele Teilketten

Schalter

Ein Schalter ist ein spezielles, kinematisch nicht wirksames Element (Typ: SWITCH (Seite 222)) in einer kinematischen Kette, das die Zustände EIN und AUS einnehmen kann.

Im Zustand **AUS** ist die Verbindung vom vorhergehenden zum nachfolgenden Element unterbrochen.

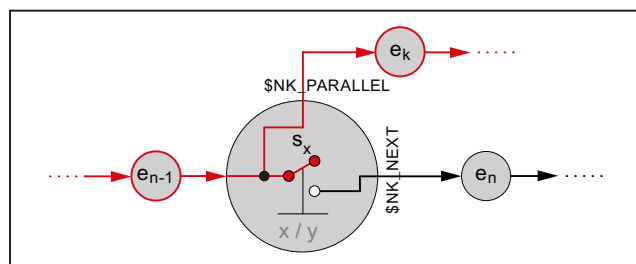


Bild 5-4 Zustand AUS

Im Zustand **EIN** ist das vorhergehende Element mit dem nachfolgenden Element verbunden.

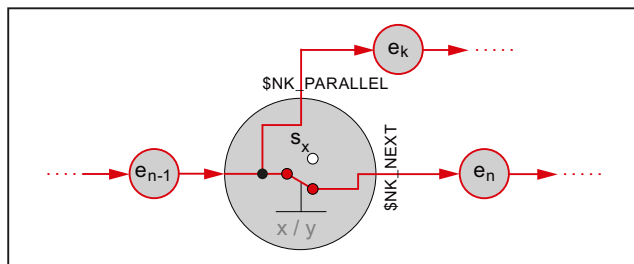


Bild 5-5 Zustand EIN

Durch den Schalter wird die Verbindung zu einem parallelen Element nicht beeinflusst.

Die maximale Anzahl möglicher Schalter ist über Maschinendaten (Seite 207) parametrierbar.

Hinweis

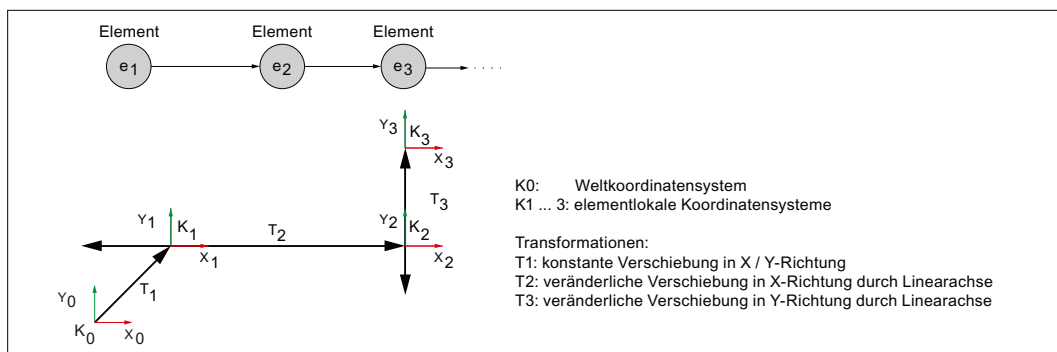
Lokales Koordinatensystem

Das lokale Koordinatensystem eines Schalters ist gegenüber dem Weltkoordinatensystem nicht gedreht.

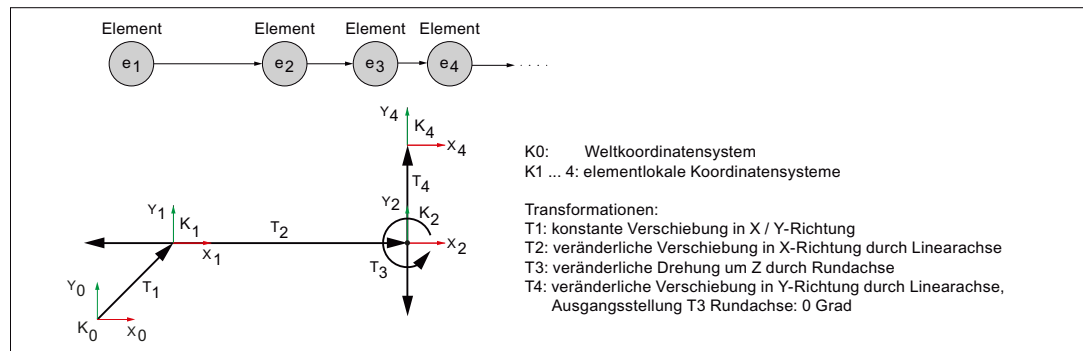
Weltkoordinatensystem

Um die kinematische Struktur einer Maschine eindeutig beschreiben zu können, werden für alle Elemente der kinematischen Kette sowohl die Orientierungs- und Verschiebungsvektoren der Maschinenachsen als auch der konstanten Drehungen / Verschiebungen, auf das Weltkoordinatensystem bezogen.

Elemente mit **Linearachsen** ändern die Orientierung der nachfolgenden Elemente nicht. Daher können die Orientierungs- und Verschiebungsvektoren der nachfolgenden Elemente ohne weitere Randbedingungen auf das Weltkoordinatensystem bezogen werden.



Elemente mit **Rundachsen** ändern die Orientierung der nachfolgenden Elemente. Daher müssen die Orientierungs- und Verschiebungsvektoren der nachfolgenden Elemente bei einer definierten Ausgangsstellung des vorhergehenden orientierungsverändernden Elementes auf das Weltkoordinatensystem bezogen werden.



Ursprung und Orientierung des Weltkoordinatensystems sind frei wählbar in Ketten-Elementen, die vor dem Root-Element definiert werden. Für das wirksame Koordinatensystem ab dem Root-Element ist folgende Anordnung erforderlich:

- Ursprung des Weltkoordinatensystems im Maschinennullpunkt
- Orientierung des Weltkoordinatensystems so, dass die Koordinatenachsen in positiver Verfahrrichtung der linearen Hauptachsen der Maschine angeordnet sind

Richtungsvektoren

Innerhalb einer kinematischen Kette werden die Richtungsvektoren, über die die Ausrichtung der Maschinenachsen angegeben wird, immer absolut, d.h. bezogen auf das Weltkoordinatensystem angegeben.

5.2 Inbetriebnahme

5.2.1 Allgemein

5.2.1.1 Übersicht

Die Inbetriebnahme der Funktion "Kinematische Kette" erfolgt mittels:

- Maschinendaten
 - Vorgabe des Mengengerüsts
 - Festlegung des ersten Elements der kinematischen Kette
- Systemvariablen
 - Festlegung der kinematischen Eigenschaften eines Elements
 - Verbindung der Elemente zur kinematischen Kette

5.2.1.2 Aufbau der Systemvariablen

Die Systemvariablen sind nach folgendem Schema aufgebaut:

- **\$NK_<Name>[<Index_1>]**
- **\$NK_<Name>[<Index_1>, <Index_2>]**

Allgemein

Die Systemvariablen zur Beschreibung der Elemente von kinematischen Ketten haben folgende Eigenschaften:

- Der Präfix für alle Systemvariablen der kinematischen Kette ist **\$NK_**, (N für NC, K für Kinematik).
- Die Systemvariablen sind über NC-Programme les- und schreibbar.
- Die Systemvariablen können über Archive gesichert und wieder in die NC eingelesen werden.

Datentyp

STRING

Alle Systemvariablen vom Datentyp STRING haben folgende Eigenschaften:

- Maximale String-Länge: 31 Zeichen
- Es wird keine Unterscheidung zwischen Groß- und Kleinschreibung gemacht
Beispiel: "Achse1" identisch mit "ACHSE1"
- Leer- und Sonderzeichen sind zulässig
Beispiel: "Achse1" nicht identisch mit " Achse 1"
- Namen, die mit **zwei** Unterstrichen "___" **beginnen**, sind für Systemzwecke reserviert und dürfen **nicht** für anwenderdefinierte Namen verwendet werden.

Hinweis

Führendes Leerzeichen

Da Leerzeichen gültige und der Unterscheidung dienende Zeichen sind, dürfen Namen, die mit einem **Leerzeichen**, gefolgt von **zwei** Unterstrichen "___" **beginnen**, prinzipiell für anwenderdefinierte Namen verwendet werden. Aufgrund der Verwechslungsgefahr mit Systemnamen wird dieses Vorgehen **nicht** empfohlen.

Index_1

Über Index_1 werden die einzelnen Elemente adressiert. Index 0 → 1. Element, Index 1 → 2. Element, ... n → (n+1) Element, mit n = (\$MN_MM_MAXNUM_KIN_CHAIN_ELEM - 1)

Alle Systemvariablen eines Elements haben den gleichen Index.

Index_2

Bei Systemvariablen, die einen Vektor enthalten, werden über Index_2 die Koordinaten des Vektors adressiert.

- 0 → X-Achse
- 1 → Y-Achse
- 2 → Z-Achse

5.2.2 Maschinendaten

5.2.2.1 Maximale Anzahl Elemente

Mit dem Maschinendatum wird die maximale Anzahl von Elementen für kinematische Ketten eingestellt:

MD18880 \$MN_MM_MAXNUM_KIN_CHAIN_ELEM = <Anzahl>

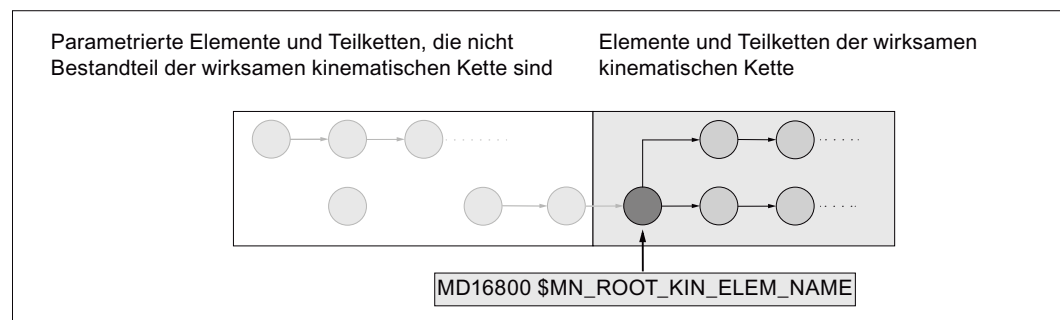
5.2.2.2 Root-Element

Mit dem Maschinendatum wird das Root-Element, d.h. das erste Element der aktuell wirksamen kinematischen Kette, festgelegt.

MD16800 \$MN_ROOT_KIN_ELEM_NAME = "<Element_Name>"

Alle parametrisierten Elemente und Teilketten, die so mit dem Root-Element verbunden sind, dass die Verbindung vom Root-Element ausgeht, sind Bestandteil der aktuell wirksamen kinematischen Kette.

Parametrisierte Elemente oder Teilketten, die nicht mit dem Root-Element verbunden sind oder deren Verbindung zum Root-Element hin zeigt, sind nicht Bestandteil der aktuell wirksamen kinematischen Kette.



5.2.2.3 Maximale Anzahl Schalter

Mit dem Maschinendatum wird die maximale Anzahl von Schaltern für kinematische Ketten eingestellt

MD18882 \$MN_MM_MAXNUM_KIN_SWITCHES = <Anzahl>

5.2.3 Systemvariablen

5.2.3.1 Übersicht

Elementunabhängige Systemvariablen

Systemvariable	Bedeutung
\$NK_SWITCH	Schaltervariable zum Ein- und Ausschalten der Schalter

Elementspezifische Systemvariablen

Die elementspezifischen Systemvariablen unterteilen sich in typunabhängige und typabhängige Variablen:

- Typunabhängige Variablen

Systemvariable	Bedeutung
\$NK_NAME	Name des aktuellen Elements e_n
\$NK_NEXT	Name des nächsten Elements e_{n+1}
\$NK_PARALLEL	Name eines vor dem aktuellen Element e_n abzweigenden parallelen Elements e_p
\$NK_TYPE	Typ des Elements

- Typabhängige Variablen

Systemvariable	Bedeutung
\$NK_OFF_DIR	Verschiebungs- oder Richtungsvektor
\$NK_AXIS	Maschinenachs- bzw. Objektname
\$NK_A_OFF	Nullpunktverschiebung bei Linear - bzw. Rundachsen
\$NK_SWITCH_INDEX	Schalterindex
\$NK_SWITCH_POS	Schalterstellung "EIN"

Die typabhängigen Variablen werden bei folgenden Typen ausgewertet:

Systemvariable	Typ				
	OFFSET	AXIS_LIN	AXIS_ROT	ROT_CON ST	SWITCH
\$NK_OFF_DIR	x	x	x	x	-
\$NK_AXIS	-	x	x	-	-
\$NK_A_OFF	-	x	x	x	-
\$NK_SWITCH_INDEX	-	-	-	-	x
\$NK_SWITCH_POS	-	-	-	-	x

Die Systemvariablen sind in den nachfolgenden Kapiteln ausführlich beschrieben.

Hinweis

Definierten Ausgangszustand herstellen

Es wird empfohlen, vor Parametrierung der kinematischen Kette einen definierten Ausgangszustand zu erzeugen. Dazu sind die Systemvariablen der kinematischen Kette mit der Funktion DELOBJ() (Seite 225) auf ihren Defaultwert zu setzen.

Ändern von Systemvariablenwerten

Wird der Wert einer der oben aufgeführten Systemvariablen geändert, wird die Änderung auf der Bedienoberfläche, z.B. SINUMERIK Operate, sofort sichtbar. Das Maschinenmodell der NC wird aber erst nach einer expliziten Anforderung zum Neuberechnen des Maschinenmodells durch Aufruf der Funktion PROTA() (Seite 305) bzw. PROTS() (Seite 306) aktualisiert.

5.2.3.2 \$NK_NAME

Funktion

In die Systemvariable ist der NC-weit eindeutige Name des Elements einzutragen. Über diesen Namen wird das Element, z.B. innerhalb von kinematischen Ketten, referenziert. Der Name wird auch im grafischen Editor von SINUMERIK Operate angezeigt.

Syntax

```
$NK_NAME [<n>] = "<Name>"
```

Bedeutung

\$NK_NAME:	Name des Elements	
	Datentyp:	STRING
	Defaultwert:	"" (Leerstring)
<n>:	Systemvariablen- bzw. Element-Index	
	Datentyp:	INT
	Wertebereich:	0, 1, 2, ... (\$MN_MM_MAXNUM_KIN_CHAIN_ELEM - 1)
<Name>:	Element-Name, max. Stringlänge: 31 Zeichen	
	Datentyp:	STRING

Beispiel

Dem 9. kinematischen Elementes wird der Name "B-Achse" zugewiesen:

Programmcode	Kommentar
N100 \$NK_NAME[8] = "B-Achse"	; 9. kin. Element, ; Name = "B-Achse"

5.2.3.3 \$NK_NEXT

Funktion

Ist das Element Bestandteil einer kinematischen Kette, ist in die Systemvariable der Namen des nachfolgenden Elements einzutragen.

Syntax

`$NK_NEXT [<n>] = "<Name>"`

Bedeutung

NK_NEXT:	Name des nachfolgenden Elements	
	Datentyp:	STRING
	Wertebereich:	Alle in \$NK_NAME (Seite 209) enthaltenen Namen
	Defaultwert:	"" (Leerstring)
<n>:	Systemvariablen- bzw. Element-Index	
	Datentyp:	INT
	Wertebereich:	0, 1, 2, ... (\$MN_MM_MAXNUM_KIN_CHAIN_ELEM - 1)
<Name>:	Element-Name, max. Stringlänge: 31 Zeichen	
	Datentyp:	STRING

Beispiel

Das 9. kinematischen Elementes hat kein nachfolgendes Element:

Programmcode	Kommentar
N100 \$NK_NEXT[8] = ""	; 9. kin. Element, ; Nachfolgendes Element = ""

5.2.3.4 \$NK_PARALLELE

Funktion

In die Systemvariable ist der Name des Elements einzutragen, das **vor** dem aktuellen Element abzweigt. Das abzweigende Element liegt parallel zum aktuellen Element. Änderungen im aktuellen Element, z.B. Positionsänderungen der zugeordneten Maschinenachse, haben keine Auswirkung auf das parallele Element.

Syntax

`$NK_PARALLELE [<n>] = "<Name>"`

Bedeutung

\$NK_PARALLEL:	Name des parallelen Elements	
	Datentyp:	STRING
	Wertebereich:	Alle in \$NK_NAME (Seite 209) enthaltenen Namen
	Defaultwert:	"" (Leerstring)
<n>:	Systemvariablen- bzw. Element-Index	
	Datentyp:	INT
	Wertebereich:	0, 1, 2, ... (\$MN_MM_MAXNUM_KIN_CHAIN_ELEM - 1)
<Name>:	Element-Name, max. Stringlänge: 31 Zeichen	
	Datentyp:	STRING

Beispiel

Das Element "Offset zum Rundtisch" ist parallel zum 9. kinematischen Element angeordnet:

Programmcode	Kommentar
N100 \$NK_PARALLEL[8] = "Offset zum Rundtisch"	; Parallel zum 9. kin. Element, ; Name = "Offset zum Rundtisch"

5.2.3.5 \$NK_TYPE

Funktion

In die Systemvariable ist der Typ des Elements einzutragen:

Typ	Beschreibung
AXIS_LIN (Seite 212)	Das Element beschreibt eine lineare Maschinenachse (Linearachse) mit dem Richtungsvektor \$NK_OFF_DIR und der Nullpunktverschiebung \$NK_A_OFF
AXIS_ROT (Seite 215)	Das Element beschreibt eine rotatorische Maschinenachse (Rundachse) mit dem Richtungsvektor \$NK_OFF_DIR und der Nullpunktverschiebung \$NK_A_OFF oder eine Spindel. Hinweis Die Position einer Spindel wird, abhängig von der Funktion welche die kinematische Kette verwendet, unterschiedlich berücksichtigt: <ul style="list-style-type: none"> • Kollisionsvermeidung: unbestimmte Position • Kinematische Transformation: abhängig von der Einstellung in \$NT_CNTRL, Bit 1-3, wird die Position entsprechend \$NK_A_OFF oder der Drehzahlsollwert berücksichtigt
ROT_CONST (Seite 219)	Das Element beschreibt eine konstante Drehung um den Richtungsvektor \$NK_OFF_DIR und dem Winkel \$NK_A_OFF
OFFSET (Seite 221)	Das Element beschreibt eine konstante lineare Verschiebung mit dem Verschiebungsvektor \$NK_OFF_DIR
SWITCH (Seite 222)	Das Element beschreibt einen Schalter der über die Systemvariable \$NK_SWITCH (Seite 224) geschaltet wird.

Syntax

`$NK_TYPE [<n>] = "<Typ>"`

Bedeutung

\$NK_TYPE:	Typ des Elements	
	Datentyp:	STRING
	Defaultwert:	"" (Leerstring)
<n>:	Systemvariablen- bzw. Element-Index	
	Datentyp:	INT
	Wertebereich:	0, 1, 2, ... (\$MN_MM_MAXNUM_KIN_CHAIN_ELEM - 1)
<Typ>:	Element-Name, max. Stringlänge: 31 Zeichen	
	Datentyp:	STRING
	Wertebereich:	"AXIS_LIN", "AXIS_ROT", "ROT_CONST", "OFFSET", "SWITCH"

Beispiel

Das 9. kinematischen Element ist eine Rundachse:

Programmcode	Kommentar
<code>N100 \$NK_TYPE[8] = "AXIS_ROT"</code>	<code>; 9. kin. Element</code> <code>; Type = Rundachse</code>

5.2.3.6 Typabhängige Variablen bei \$NK_TYPE = "AXIS_LIN"

\$NK_OFF_DIR

Funktion

In die Systemvariable ist der Richtungsvektor einzutragen, entlang dessen sich die dem Element zugeordnete Linearachse \$NK_AXIS bewegt. Das Ausgangskordinatensystem ergibt sich somit aus dem Eingangskordinatensystem, verschoben um den aktuellen Positionswert der Linearachse und der in \$NK_A_OFF angegebenen Nullpunktverschiebung.

Randbedingungen:

- Der Richtungsvektor ist absolut, d.h. bezogen auf das **Weltkoordinatensystem**, anzugeben.
- Der Richtungsvektor ist so anzugeben, dass er in die positive Verfahrrichtung der Maschinenachse zeigt.
- Der Betrag des Richtungsvektors muss größer $1 \cdot 10^{-6}$ sein.

Syntax

`$NK_OFF_DIR [<n>, <k>] = <Wert>`

Bedeutung

\$NK_OFF_DIR:	Richtungsvektor (X; Y; Z)	
	Datentyp:	REAL
	Wertebereich:	Richtungsvektor: $1 \cdot 10^{-6} < \text{Vektor} \leq \text{max. REAL-Wert}$
	Defaultwert:	(0.0, 0.0, 0.0)
<n>:	Systemvariablen- bzw. Element-Index	
	Datentyp:	INT
	Wertebereich:	0, 1, 2, ... (\$MN_MM_MAXNUM_KIN_CHAIN_ELEM - 1)
<k>:	Koordinatenindex	
	Datentyp:	INT
	Wertebereich:	0: X-Koordinate (Abszisse) 1: Y-Koordinate (Ordinate) 2: Z-Koordinate (Applikate)
<Wert>:	Koordinatenwert	
	Datentyp:	REAL
	Wertebereich:	- max. REAL-Wert $\leq x \leq$ + max. REAL-Wert

Beispiel

Die Linearachse des 9. Elements bewegt sich entlang des Richtungsvektors. Der Richtungsvektor ist der Einheitsvektor (1; 0; 0), gedreht um Z mit $\gamma=90^\circ$ in der X/Y-Ebene und um X mit $\alpha=10^\circ$ in der Y/Z Ebene, bezogen auf das Weltkoordinatensystem. Daraus ergeben sich folgende Werte für die Komponenten (x, y, z) des Richtungsvektors:

- $x = \cos(\gamma) * \cos(\alpha) = \cos(90) * \cos(10) = 0,0$
- $y = \sin(\gamma) * \cos(\alpha) = \sin(90) * \cos(10) \approx 0,985$
- $z = \sin(\alpha) = \sin(10) \approx 0,174$

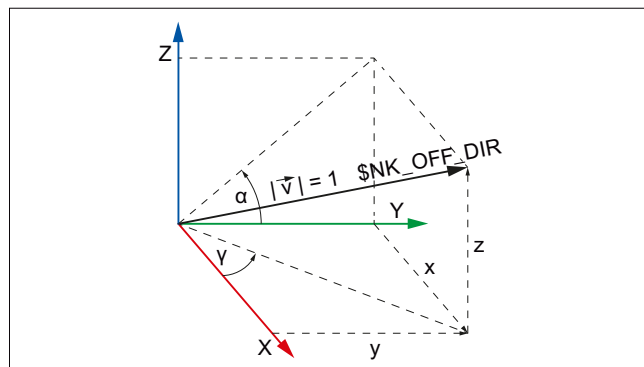


Bild 5-6 Richtungsvektor, allgemein

Programmcode	Kommentar
; 9. kinematisches Element	
N100 \$NK_OFF_DIR[8,0] = COS(90)*COS(10)	; 0 = X-Komponente
N110 \$NK_OFF_DIR[8,1] = SIN(90)*COS(10)	; 1 = Y-Komponente
N120 \$NK_OFF_DIR[8,2] = SIN(10)	; 2 = Z-Komponente

\$NK_AXIS

Funktion

In die Systemvariable ist der Name der Maschinenachse (MD10000 \$MN_AXCONF_MACHAX_NAME_TAB) einzutragen, die dem Element zugeordnet wird.

Das Ausgangskoordinatensystem des Elements ergibt sich aus dem Eingangskoordinatensystem, verschoben um die aktuelle Sollposition der Maschinenachse im MKS und dem in \$NK_A_OFF angegebenen Offset. In der Sollposition der Maschinenachse sind alle aktiven Nullpunktverschiebungen und Überlagerungen enthalten.

Die Maschinenachse muss, entsprechend dem in \$NK_TYPE eingetragenen Typ AXIS_LIN, eine Linearachse sein:

MD30300 \$MA_IS_ROT_AX = 0

Syntax

\$NK_AXIS [<n>] = <Name>

Bedeutung

\$NK_AXIS:	Maschinenachsname	
	Datentyp:	STRING
	Wertebereich:	Maschinenachsamen
	Defaultwert:	"" (Leerstring)
<n>:	Systemvariablen- bzw. Element-Index	
	Datentyp:	INT
	Wertebereich:	0, 1, 2, ... (\$MN_MM_MAXNUM_KIN_CHAIN_ELEM - 1)
<Wert>:	Maschinenachsname	
	Datentyp:	STRING
	Wertebereich:	Maschinenachsenamen

Beispiel

Dem 9. kinematischen Element ist als Linearachse die Maschinenachse mit dem Namen V1 zugeordnet

Programmcode	Kommentar
N100 \$NK_AXIS[8] = "V1"	; 9. kin. Element ; Achse = Maschinenachse V1

\$NK_A_OFF

Funktion

In die Systemvariable kann für die zugeordnete Maschinenachse (\$NK_AXIS) eine zusätzliche Nullpunktverschiebung eingetragen werden. Diese Nullpunktverschiebung ist nur innerhalb der kinematischen Kette wirksam.

Syntax

\$NK_A_OFF [<n>] = <Wert>

Bedeutung

\$NK_A_OFF:	Nullpunktverschiebung	
	Datentyp:	REAL
	Wertebereich:	- max. REAL-Wert ≤ x ≤ ± max. REAL-Wert
	Defaultwert:	0.0
<n>:	Systemvariablen- bzw. Element-Index	
	Datentyp:	INT
	Wertebereich:	0, 1, 2, ... (\$MN_MM_MAXNUM_KIN_CHAIN_ELEM - 1)
<Wert>:	Verschiebungswert	
	Datentyp:	REAL
	Wertebereich:	- max. REAL-Wert ≤ x ≤ ± max. REAL-Wert

Beispiel

Der Nullpunkt der Linearachse des 9. kinematischen Elements ist um 30.0 mm gegenüber der modellierten Kinematik verschoben.

Die Einheit, in der der Zahlenwert interpretiert wird, ist abhängig vom aktuellen Eingabesystem (inch / metrisch).

Programmcode	Kommentar
N100 \$NK_A_OFF[8] = 30.0	; 9. kin. Element ; Nullpunktversch. = 30.0 mm

5.2.3.7 Typabhängige Variablen bei \$NK_TYPE = "AXIS_ROT"**\$NK_OFF_DIR****Funktion**

In die Systemvariable ist der Richtungsvektor einzutragen, um den sich die dem Element zugeordnete Rundachse \$NK_AXIS dreht. Das Ausgangskordinatensystem berechnet sich somit aus dem Eingangskordinatensystem, gedreht um den aktuellen Positionswert der Rundachse und dem in \$NK_A_OFF angegebenen Offset, um den Richtungsvektor \$NK_OFF_DIR.

Randbedingungen:

- Der Richtungsvektor ist absolut, d.h. bezogen auf das **Weltkordinatensystem**, anzugeben.
- Der Richtungsvektor ist so anzugeben, dass, entsprechend der "Rechten-Hand-Regel", bei positiver Drehrichtung der Rundachse der Daumen in Richtung des Vektors zeigt.
- Der Betrag des Richtungsvektors muss größer $1 \cdot 10^{-6}$ sein.

Hinweis**Spindel**

Ist die zugeordnete Maschinenachse eine Spindel, wird ihre Position funktionspezifisch unterschiedlich berücksichtigt:

- Kollisionsvermeidung: unbestimmte Position
- Kinematische Transformation: abhängig von der Einstellung in \$NT_CNTRL, Bit 1-3
 - Bit x == 0 → unbestimmte Position
 - Bit x == 1 → aktueller Positionswert + \$NK_A_OFF,

Syntax

\$NK_OFF_DIR[<n>,<k>] = <Wert>

Bedeutung

\$NK_OFF_DIR:	Richtungsvektor (X; Y; Z)	
	Datentyp:	REAL
	Wertebereich:	Richtungsvektor: $1 \cdot 10^{-6} < \text{Vektor} \leq \text{max. REAL-Wert}$
	Defaultwert:	(0.0, 0.0, 0.0)
<n>:	Systemvariablen- bzw. Element-Index	
	Datentyp:	INT
	Wertebereich:	0, 1, 2, ... (\$MN_MM_MAXNUM_KIN_CHAIN_ELEM - 1)
<k>:	Koordinatenindex	
	Datentyp:	INT
	Wertebereich:	0: X-Koordinate (Abszisse) 1: Y-Koordinate (Ordinate) 2: Z-Koordinate (Applikate)
<Wert>:	Koordinatenwert	
	Datentyp:	REAL
	Wertebereich:	- max. REAL-Wert $\leq x \leq$ + max. REAL-Wert

Beispiel

Die Rundachse des 9. Elements dreht sich um den Richtungsvektor. Der Richtungsvektor ist der Einheitsvektor (1; 0; 0), gedreht um Z mit $\gamma=90^\circ$ in der X/Y-Ebene und um X mit $\alpha=10^\circ$ in der Y/Z Ebene, bezogen auf das Weltkoordinatensystem. Daraus ergeben sich folgende Werte für die Komponenten (x, y, z) des Richtungsvektors:

- $x = \cos(\gamma) \cdot \cos(\alpha) = \cos(90) \cdot \cos(10) = 0,0$
- $y = \sin(\gamma) \cdot \cos(\alpha) = \sin(90) \cdot \cos(10) \approx 0,985$
- $z = \sin(\alpha) = \sin(10) \approx 0,174$

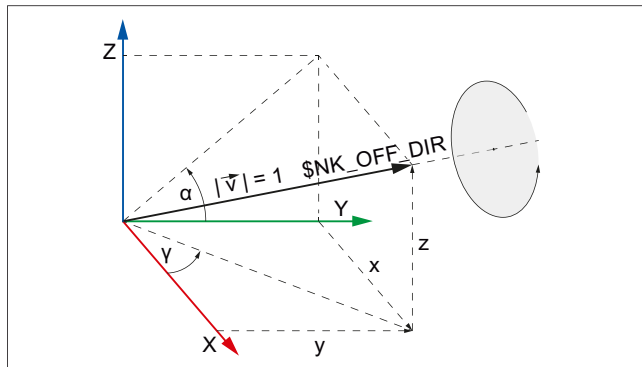


Bild 5-7 Richtungsvektor, allgemein

Programmcode	Kommentar
; 9. kinematisches Element	
N100 \$NK_OFF_DIR[8,0] = COS(90)*COS(10)	; 0 = X-Komponente
N110 \$NK_OFF_DIR[8,1] = SIN(90)*COS(10)	; 1 = Y-Komponente
N120 \$NK_OFF_DIR[8,2] = SIN(10)	; 2 = Z-Komponente

\$NK_AXIS

Funktion

In die Systemvariable ist der Name der Maschinenachse (MD10000 \$MN_AXCONF_MACHAX_NAME_TAB) einzutragen, die dem Element zugeordnet wird.

Das Ausgangskoordinatensystem des Elements ergibt sich aus dem Eingangskoordinatensystem, gedreht um die aktuelle Sollposition der Maschinenachse im MKS und dem in \$NK_A_OFF angegebenen Offset. In der Sollposition der Maschinenachse sind alle aktiven Nullpunktverschiebungen und Überlagerungen enthalten.

Der Typ der Maschinenachse muss dem in \$NK_TYPE eingetragenen Typ entsprechen:

Die Maschinenachse muss, entsprechend dem in \$NK_TYPE eingetragenen Typ AXIS_ROT, eine Rundachse sein:

MD30300 \$MA_IS_ROT_AX = 1

Syntax

\$NK_AXIS[<n>] = <Name>

Bedeutung

\$NK_AXIS:	Maschinenachsname	
	Datentyp:	STRING
	Wertebereich:	Maschinenachsnamen
	Defaultwert:	"" (Leerstring)
<n>:	Systemvariablen- bzw. Element-Index	
	Datentyp:	INT
	Wertebereich:	0, 1, 2, ... (\$MN_MM_MAXNUM_KIN_CHAIN_ELEM - 1)

<Wert>:	Maschinenachsname	
	Datentyp:	STRING
	Wertebereich:	Maschinenachsenamen

Beispiel

Dem 9. kinematischen Element ist als Rundachse die Maschinenachse mit dem Namen B1 zugeordnet.

Programmcode	Kommentar
N100 \$NK_AXIS[8] = "B1"	; 9. kin. Element
	; Achse = Maschinenachse B1

\$NK_A_OFF

Funktion

In die Systemvariable kann für die zugeordnete Maschinenachse (\$NK_AXIS) eine zusätzliche Nullpunktverschiebung eingetragen werden. Diese Nullpunktverschiebung ist nur innerhalb der kinematischen Kette wirksam.

Syntax

\$NK_A_OFF[<n>] = <Wert>

Bedeutung

\$NK_A_OFF:	Nullpunktverschiebung	
	Datentyp:	REAL
	Wertebereich:	- max. REAL-Wert ≤ x ≤ ± max. REAL-Wert
	Defaultwert:	0.0
<n>:	Systemvariablen- bzw. Element-Index	
	Datentyp:	INT
	Wertebereich:	0, 1, 2, ... (\$MN_MM_MAXNUM_KIN_CHAIN_ELEM - 1)
<Wert>:	Verschiebungswert	
	Datentyp:	REAL
	Wertebereich:	- max. REAL-Wert ≤ x ≤ ± max. REAL-Wert

Beispiel

Der Nullpunkt der Rundachse des 9. kinematischen Elements ist um 30.0° gegenüber der modellierten Kinematik verschoben.

Programmcode	Kommentar
N100 \$NK_A_OFF[8] = 30.0	; 9. kin. Element
	; Nullpunktversch. = 30.0°

5.2.3.8 Typabhängige Variablen bei \$NK_TYPE = "ROT_CONST"

\$NK_OFF_DIR

Funktion

In die Systemvariable ist der Richtungsvektor einzutragen, um den die konstante Drehung ausgeführt wird. Das Ausgangskordinatensystem berechnet sich somit aus dem Eingangskordinatensystem, gedreht um den in \$NK_A_OFF angegebenen Winkel um den Richtungsvektor \$NK_OFF_DIR.

Randbedingungen:

- Der Richtungsvektor ist absolut, d.h. bezogen auf das **Weltkordinatensystem**, anzugeben.
- Der Richtungsvektor ist so anzugeben, dass, entsprechend der "Rechten-Hand-Regel", bei positiver Drehrichtung der Rundachse der Daumen in Richtung des Vektors zeigt.
- Der Betrag des Richtungsvektors muss größer $1 \cdot 10^{-6}$ sein.

Syntax

\$NK_OFF_DIR[<n>,<k>] = <Wert>

Bedeutung

\$NK_OFF_DIR:	Richtungsvektor (X; Y; Z)	
	Datentyp:	REAL
	Wertebereich:	Richtungsvektor: $1 \cdot 10^{-6} < \text{Vektor} \leq \text{max. REAL-Wert}$
	Defaultwert:	(0.0, 0.0, 0.0)
<n>:	Systemvariablen- bzw. Element-Index	
	Datentyp:	INT
	Wertebereich:	0, 1, 2, ... (\$MN_MM_MAXNUM_KIN_CHAIN_ELEM - 1)
<k>:	Koordinatenindex	
	Datentyp:	INT
	Wertebereich:	0: X-Koordinate (Abszisse) 1: Y-Koordinate (Ordinate) 2: Z-Koordinate (Applikate)
<Wert>:	Koordinatenwert	
	Datentyp:	REAL
	Wertebereich:	$-\text{max. REAL-Wert} \leq x \leq +\text{max. REAL-Wert}$

Beispiel

Die Ausgangskordinatensystem des 9. Elements ergibt sich aus dem Eingangskordinatensystem, gedreht um den in \$NK_A_OFF angegebenen Winkel um den Richtungsvektor. Der Richtungsvektor ist der Einheitsvektor (1; 0; 0), gedreht um Z mit $\gamma=90^\circ$ in der X/Y-Ebene und um X mit $\alpha=10^\circ$ in der Y/Z Ebene, bezogen auf das

Weltkoordinatensystem. Daraus ergeben sich folgende Werte für die Komponenten (x, y, z) des Richtungsvektors:

- $x = \cos(\gamma) * \cos(\alpha) = \cos(90) * \cos(10) = 0,0$
- $y = \sin(\gamma) * \cos(\alpha) = \sin(90) * \cos(10) \approx 0,985$
- $z = \sin(\alpha) = \sin(10) \approx 0,174$

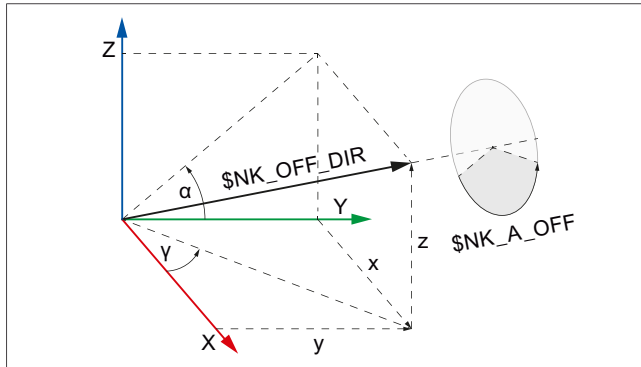


Bild 5-8 ROT_CONST

Programmcode	Kommentar
<code>; 9. kinematisches Element</code>	
<code>N100 \$NK_OFF_DIR[8,0] = COS(90)*COS(10)</code>	<code>; 0 = X-Komponente</code>
<code>N110 \$NK_OFF_DIR[8,1] = SIN(90)*COS(10)</code>	<code>; 1 = Y-Komponente</code>
<code>N120 \$NK_OFF_DIR[8,2] = SIN(10)</code>	<code>; 2 = Z-Komponente</code>

\$NK_A_OFF

Funktion

In die Systemvariable ist der Winkel einzutragen um den das Ausgangs- gegenüber dem Eingangskordinatensystem um den Richtungsvektor \$NK_OFF_DIR gedreht wird.

Syntax

`$NK_A_OFF[<n>] = <Wert>`

Bedeutung

\$NK_A_OFF:	Drehwinkel	
	Datentyp:	REAL
	Wertebereich:	- max. REAL-Wert ≤ x ≤ + max. REAL-Wert
	Defaultwert:	0.0
<n>:	Systemvariablen- bzw. Element-Index	
	Datentyp:	INT
	Wertebereich:	0, 1, 2, ... (\$MN_MM_MAXNUM_KIN_CHAIN_ELEM - 1)
<Wert>:	Winkel	
	Datentyp:	REAL
	Wertebereich:	- max. REAL-Wert ≤ x ≤ + max. REAL-Wert

Beispiel

Der Drehwinkel des 9. kinematischen Elements beträgt 30.0°.

Programmcode	Kommentar
N100 \$NK_A_OFF[8] = 30.0	; 9. kin. Element ; Drehwinkel. = 30.0°

5.2.3.9 Typabhängige Variablen bei \$NK_TYPE = "OFFSET"**\$NK_OFF_DIR****Funktion**

In die Systemvariable ist der Verschiebungsvektor einzutragen, um den das Ausgangskordinatensystem gegenüber dem Eingangskordinatensystem verschoben ist.

Der Verschiebungsvektor ist absolut, d.h. bezogen auf das **Weltkordinatensystem**, anzugeben.

Syntax

\$NK_OFF_DIR[<n>,<k>] = <Wert>

Bedeutung

\$NK_OFF_DIR:	Verschiebungsvektor (X; Y; Z)	
	Datentyp:	REAL
	Wertebereich:	(- max. REAL-Wert) ≤ x ≤ (+ max. REAL-Wert)
	Defaultwert:	(0.0, 0.0, 0.0)
<n>:	Systemvariablen- bzw. Element-Index	
	Datentyp:	INT
	Wertebereich:	0, 1, 2, ... (\$MN_MM_MAXNUM_KIN_CHAIN_ELEM - 1)
<k>:	Koordinatenindex	
	Datentyp:	INT
	Wertebereich:	0: X-Koordinate (Abszisse) 1: Y-Koordinate (Ordinate) 2: Z-Koordinate (Applikate)
<Wert>:	Koordinatenwert	
	Datentyp:	REAL
	Wertebereich:	(- max. REAL-Wert) ≤ x ≤ (+ max. REAL-Wert)

Beispiel

Das Ausgangskoordinatensystem des 9. Elements ergibt sich aus dem Eingangskoordinatensystem, verschoben um den Verschiebungsvektor (x, y, z) mit den folgenden, auf das Weltkoordinatensystem bezogenen Koordinaten:

- x = 10,0
- y = 20,0
- z = 30,0

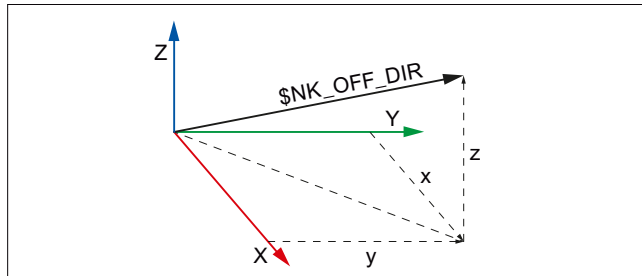


Bild 5-9 Verschiebungsvektor

Programmcode	Kommentar
; 9. kinematisches Element	
N100 \$NK_OFF_DIR[8,0] = 10.0	; 0 = X-Komponente
N110 \$NK_OFF_DIR[8,1] = 20.0	; 1 = Y-Komponente
N120 \$NK_OFF_DIR[8,2] = 30.0	; 2 = Z-Komponente

5.2.3.10 Typabhängige Variablen bei \$NK_TYPE = "SWITCH"

\$NK_SWITCH_INDEX

Funktion

Ein Schalter wird durch die Systemvariablen \$NK_SWITCH_INDEX und \$NK_SWITCH_POS (siehe nachfolgenden Absatz) gebildet.

In \$NK_SWITCH_INDEX ist der Index i einzutragen mit dem der Schalter über die Systemvariable \$NK_SWITCH[<i>] ein- und ausgeschaltet wird.

Syntax

\$NK_SWITCH_INDEX[<n>] = <i>

Bedeutung

\$NK_SWITCH_INDEX:	Index i über den der Schalter durch die Systemvariable \$NK_SWITCH[<i>] angesprochen wird	
	Datentyp:	INT
	Wertebereich:	-1, 0, 1, 2, ... (\$MN_MAXNUM_KIN_SWITCHES - 1) -1: Der Zustand des Schalters ist konstant EIN
	Defaultwert:	-1

<n>:	Systemvariablen- bzw. Element-Index	
	Datentyp:	INT
	Wertebereich:	0, 1, 2, ... (\$MN_MM_MAXNUM_KIN_CHAIN_ELEM - 1)
<i>:	Indexwert	
	Datentyp:	INT

Beispiel

Das 9. Element der kinematischen Kette ist ein Schalter, der durch die Systemvariable \$NK_SWITCH[2] geschaltet wird.

Programmcode	Kommentar
; 9. Element	
N100 \$NK_SWITCH_INDEX[8] = 2	; Index 2

\$NK_SWITCH_POS**Funktion**

Ein Schalter wird durch die Systemvariablen \$NK_SWITCH_INDEX (siehe vorhergehenden Absatz) und \$NK_SWITCH_POS gebildet.

In \$NK_SWITCH_POS ist der Wert p (Einschaltwert) einzutragen, bei dem der Schalter durch die Systemvariable \$NK_SWITCH[<i>] in Stellung EIN geschaltet wird.

Der Schalter hat abhängig von \$NK_SWITCH[<i>] folgende Zustände:

- **EIN:** Der in \$NK_SWITCH_POS angegebene Einschaltwert p des Schalters ist gleich dem aktuellen Wert von \$NK_SWITCH[<i>].
 $\$NK_SWITCH_POS[<n>] == \$NK_SWITCH[<i>]$
Das vorhergehende Element der kinematischen Kette ist mit dem Ausgang d.h. dem in \$NK_NEXT angegebenen nachfolgenden Element des Schalters verbunden.
- **AUS:** Die in \$NK_SWITCH_POS angegebene Einschaltwert des Schalters ist ungleich dem aktuellen Wert von \$NK_SWITCH[<i>].
 $\$NK_SWITCH_POS[<n>] \neq \$NK_SWITCH[<i>]$
Das vorhergehende Element der kinematischen Kette ist **nicht** mit dem Ausgang d.h. dem in \$NK_NEXT angegebenen nachfolgenden Element des Schalters verbunden.

Der Einschaltwert ist frei wählbar

Hinweis**Paralleles Element \$NK_PARALLEL**

Die Verbindung zu einem in \$NK_PARALLEL angegebenen parallelen Element wird durch den Schalter nicht beeinflusst. D.h. das vorhergehende Element ist immer mit dem parallel vom Schalter abzweigenden Element verbunden.

Syntax

$\$NK_SWITCH_POS[<n>] = <p>$

Bedeutung

\$NK_SWITCH_POS:	Einschaltwert	
	Datentyp:	INT
	Wertebereich:	0 ≤ x ≤ positiver max. INT-Wert
	Defaultwert:	0
<n>:	Systemvariablen- bzw. Element-Index	
	Datentyp:	INT
	Wertebereich:	0, 1, 2, ... (\$MN_MM_MAXNUM_KIN_CHAIN_ELEM - 1)
<p>:	Wert für die Schalterstellung EIN	
	Datentyp:	INT

Beispiel

Das 9. Element der kinematischen Kette ist ein Schalter, der eingeschaltet ist, wenn \$NK_SWITCH[3] == 1

Programmcode	Kommentar
; 9. Element	
N110 \$NK_SWITCH_POS[8] = 1	; Schalterstellung = 1

Siehe auch

\$NK_SWITCH (Seite 224)

\$MN_MAXNUM_KIN_SWITCHES (Seite 207)

\$MN_MM_MAXNUM_KIN_CHAIN_ELEM (Seite 207)

5.2.3.11 \$NK_SWITCH

Funktion

Die Schaltvariable besteht aus einem Feld von Schaltpositionen i. In diese sind die aktuellen Schaltstellungen p einzutragen.

Funktion

Zur Parametrierung (Seite 222) eines Schalters_i in einer kinematischen Kette muss der Schalter_i mit dem Index i der Schaltvariablen verbunden und ihm seine Schalterstellung p für den Zustand EIN zugeordnet werden.

\$NK_SWITCH_INDEX[<n>] = <i>

\$NK_SWITCH_POS[<n>] = <p>

Der Schalter_i kann dann über den Index i der Schaltvariablen ein- und ausgeschaltet werden:

Einschalten: \$NK_SWITCH[<i>] = <p>

Ausschalten: \$NK_SWITCH[<i>] ≠ <p>

Es können beliebig viele Schalter mit einem Index der Schaltvariablen verbunden sein.

Syntax

```
$NK_SWITCH[<i>] = <p>
```

Bedeutung

\$NK_SWITCH<i>:	Schaltvariable mit Schaltpositionen i	
	Datentyp:	INT
	Wertebereich:	-1 ≤ x ≤ positiver max. INT-Wert Hinweis -1: Grundzustand AUS
<i>:	Schalter-Index	
	Datentyp:	INT
	Wertebereich:	0, 1, 2, ... (MD18882 \$MN_MM_MAX_NUM_KIN_SWITCHES - 1)
<p>:	Schalterstellung	
	Datentyp:	INT
	Wertebereich:	-1 ≤ x ≤ positiver max. INT-Wert -1: Grundzustand AUS

Beispiel

Dem 9. kinematischen Elementes wird der Name "B-Achse" zugewiesen:

Programmcode	Kommentar
N100 \$NK_SWITCH[3] = 1	; Aktuelle Schalterstellung ; für Schalter[3] = 1

5.3 Programmierung

5.3.1 Löschen von Komponenten (DELOBJ)

Die Funktion `DELOBJ()` "löscht" Komponenten durch Zurücksetzen der zugeordneten Systemvariablen auf ihren Defaultwert:

- Elemente von kinematischen Ketten
- Schutzbereiche, Schutzbereichselemente und Kollisionspaare
- Transformationsdaten

Syntax

```
[<RetVal>=] DELOBJ (<CompType> [ , , , <NoAlarm> ] )
[<RetVal>=] DELOBJ (<CompType> , <Index1> [ , , <NoAlarm> ] )
[<RetVal>=] DELOBJ (<CompType> [ , <Index1> ] [ , <Index2> ] [ , <NoAlarm> ] )
```

Bedeutung

DELOBJ:	Löschen von Elementen von kinematischen Ketten, Schutzbereichen, Schutzbereichselementen, Kollisionspaaren und Transformationsdaten
<CompType>:	Typ der zu löschenden Komponente
	Datentyp: STRING
	Wert: "KIN_CHAIN_ELEM" Bedeutung: Systemvariablen aller kinematischen Elemente: \$NK_...
	Wert: "KIN_CHAIN_SWITCH" Bedeutung: Systemvariable \$NK_SWITCH[<i>]]
	Wert: "KIN_CHAIN_ALL" Bedeutung: Alle kinematischen Elemente und Schalter. Gleichbedeutend mit dem sukzessiven Aufruf von DELOBJ mit "KIN_CHAIN_ELEM" und "KIN_CHAIN_SWITCH"
	Wert: "PROT_AREA" Bedeutung: Systemvariablen der Schutzbereiche: <ul style="list-style-type: none"> • \$NP_PROT_NAME • \$NP_CHAIN_NAME • \$NP_CHAIN_ELEM • \$NP_1ST_PROT
	Wert: "PROT_AREA_ELEM" Bedeutung: Systemvariablen der Schutzbereichselemente von Maschinenschutzbereichen und/oder automatischen Werkzeugschutzbereichen: <ul style="list-style-type: none"> • \$NP_NAME • \$NP_NEXT • \$NP_NEXTP • \$NP_COLOR • \$NP_D_LEVEL • \$NP_USAGE • \$NP_TYPE • \$NP_FILENAME • \$NP_PARA • \$NP_OFF • \$NP_DIR • \$NP_ANG
	Wert: "PROT_AREA_COLL_PAIRS" Bedeutung: Systemvariablen der Kollisionspaare: <ul style="list-style-type: none"> • \$NP_COLL_PAIR • \$NP_SAFETY_DIST
	Wert: "PROT_AREA_ALL" Bedeutung: Alle Schutzbereiche, Schutzbereichselemente und Kollisionspaare (Systemvariablen \$NP_...) Gleichbedeutend mit dem sukzessiven Aufruf von DELOBJ mit "PROT_AREA", "PROT_AREA_ELEM" und "PROT_AREA_COLL_PAIRS"
	Wert: "TRAFO_DATA" Bedeutung: Systemvariablen aller Transformationen \$NT_...

<Index1>:	Index der ersten zu löschenden Komponente (optional)	
	Datentyp:	INT
	Defaultwert:	-1
	Wertebereich:	$-1 \leq x \leq$ (maximale Anzahl projektierter Komponenten -1)
	Wert	Bedeutung
	0, 1, 2,	Index der zu löschenden Komponente.
	-1	Alle Komponenten des angegebenen Typs werden gelöscht. <Index2> wird nicht ausgewertet.
<Index2>:	Index der letzten zu löschenden Komponenten (optional) Ist <Index2> nicht programmiert, werden nur die Systemvariablen der in <Index1> referenzierten Komponente gelöscht.	
	Datentyp:	INT
	Defaultwert:	Nur die Systemvariablen der in <Index1> referenzierten Komponente werden gelöscht.
	Wertebereich:	<Index1> < x ≤ (max. Anzahl projektierter Komponenten -1)
<NoAlarm>:	Alarmunterdrückung (optional)	
	Datentyp:	BOOL
	Defaultwert:	FALSE
	Wert	Bedeutung
	FALSE	Im Fehlerfall (<RetVal> < 0) wird die Programmabarbeitung angehalten und ein Alarm angezeigt.
	TRUE	Im Fehlerfall wird die Programmabarbeitung nicht angehalten und es wird kein Alarm angezeigt. Anwendungsfall: Anwenderspezifische Reaktion entsprechend Rückgabewert
<RetVal>:	Rückgabewert der Funktion	
	Datentyp:	INT
	Wertebereich:	0, -1, -2, ... -7
	Wert	Bedeutung
	0	Kein Fehler aufgetreten.
	-1	Aufruf der Funktion ohne Parameter. Mindestens der Parameter <CompType> muss angegeben werden.
	-2	<CompType> bezeichnet unbekannte Komponente
	-3	<Index1> ist kleiner als -1
	-4	<Index1> ist größer als die projektierte Anzahl Komponenten
	-5	<Index1> hat beim Löschen einer Komponentengruppe einen Wert ungleich -1
	-6	<Index2> ist kleiner als <Index1>
-7	<Index2> ist größer als die projektierte Anzahl Komponenten	

5.3.2 Indexermittlung per Namen (NAMETOINT)

In Systemvariablenfeldern vom Typ STRING sind anwenderspezifische Namen eingetragen. Anhand des Bezeichners der Systemvariablen und des Namens, ermittelt die Funktion `NAMETOINT()` den zum Namen gehörenden Indexwert, unter dem er im Systemvariablenfeld abgelegt ist.

Syntax

```
<RetVal> = NAMETOINT (<SysVar>, <Name> [, <NoAlarm>])
```

Bedeutung

NAMETOINT:	Ermittlung des Systemvariablenindex	
<SysVar>:	Name des Systemvariablenfeldes vom Typ STRING	
	Datentyp:	STRING
	Wertebereich:	Namen aller Systemvariablenfelder der NC vom Typ STRING
<Name>:	Zeichenkette bzw. Name, zu dem der Systemvariablenindex ermittelt werden soll.	
	Datentyp:	STRING
<NoAlarm>:	Alarmunterdrückung (optional)	
	Datentyp:	BOOL
	Defaultwert:	FALSE
	Wert	Bedeutung
	TRUE	Im Fehlerfall wird die Programmabarbeitung nicht angehalten und es wird kein Alarm angezeigt. Anwendungsfall: Anwenderspezifische Reaktion entsprechend Rückgabewert
FALSE	Im Fehlerfall (<RetVal> < 0) wird die Programmabarbeitung angehalten und ein Alarm angezeigt.	
<RetVal>:	Systemvariablenindex oder Fehlermeldung	
	Datentyp:	INT
	Wertebereich:	$-1 \leq x \leq (\text{max. Anzahl projektierte Komponenten} - 1)$
	Wert	Bedeutung
	≥ 0	Der gesuchte Name wurde unter dem angegebenen Systemvariablenindex gefunden.
-1	Der gesuchte Name wurde nicht gefunden bzw. es ist ein Fehler aufgetreten.	

Beispiel

Programmcode	Kommentar
DEF INT INDEX	
\$NP_PROT_NAME[27] = "Abdeckung"	
...	
INDEX = NAMETOINT("\$NP_PROT_NAME", "Abdeckung")	; INDEX == 27

5.4 Beispiel

5.4.1 Vorgaben

Allgemeines

Anhand einer 5-Achs-Maschine mit drei unterschiedlichen Werkzeugköpfen, die wechselweise zum Einsatz kommen, wird beispielhaft das prinzipielle Vorgehen zur Parametrierung der kinematischen Kette mit drei Schaltern über ein Teileprogramm gezeigt. Im Teileprogramm werden alle für die kinematische Kette relevanten Systemvariablen geschrieben:

- Kinematische Kette \$NK_...

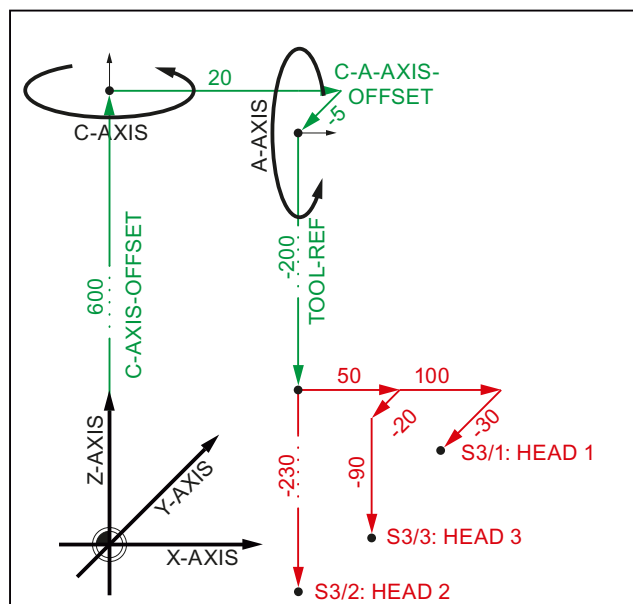
Options- und Maschinendaten

Für das Beispiel sind folgende Options- und Maschinendaten einzustellen:

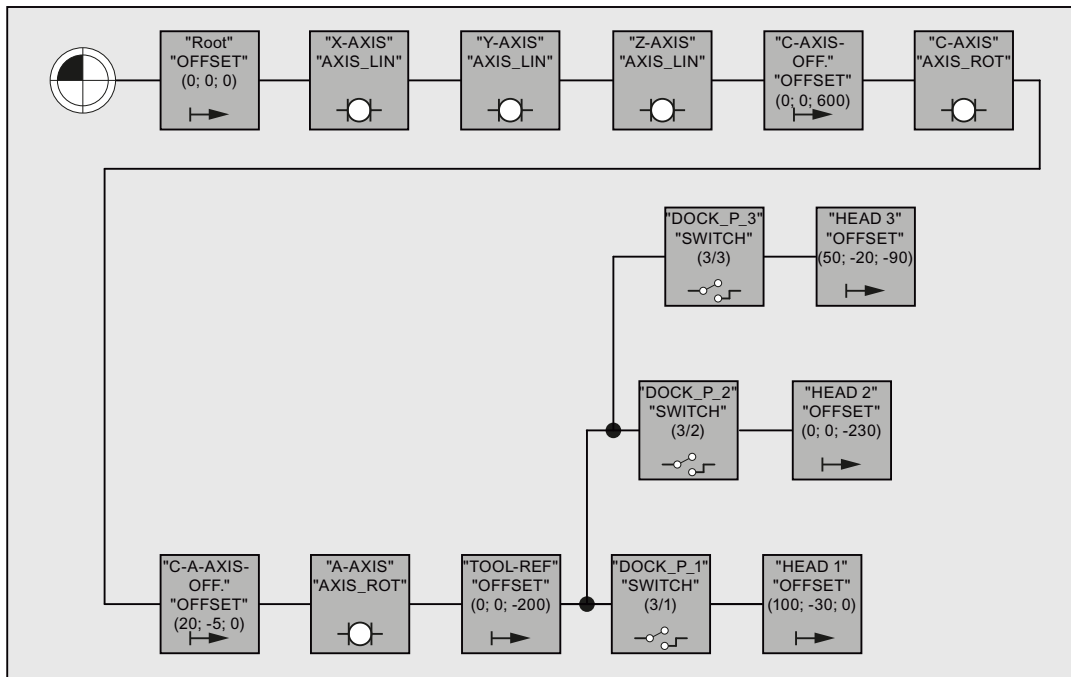
- MD19830 \$ON_COLLISION_MASK.Bit 0 = 1
- MD18880 \$MN_MM_MAXNUM_KIN_CHAIN_ELEM = 15

Kinematische Kette

Schematische Darstellung der Maschinenkinematik



Elemente der kinematischen Kette



Die kinematische Kette beginnt mit einem Element vom Typ "Offset". Diesem werden bei einer vollständigen Parametrierung der Kollisionsvermeidung alle statischen Schutzbereiche der Maschine zugeordnet.

Auf das Offset-Element folgen die kinematischen Elemente der linearen Maschinenachsen X, Y und Z. Sowie die Offset- und kinematischen Elemente der Rundachsen C und A.

Nach dem Offset-Element für den Referenzpunkt der Werkzeugköpfe folgen die drei Schalter zum Ein- und Ausschalten der Werkzeugköpfe. Alle drei Schalter beziehen sich durch $\$NK_INDEX = 3$ auf dieselbe Schaltervariable $\$NK_SWITCH[3]$. Durch die unterschiedlichen Schalterpositionen (1, 2, 3) wird nur die Teilkette des jeweils aktiven Werkzeugkopfs wirksam.

5.4.2 Teileprogramm des Maschinenmodells

Programmcode

```

;=====
; Definitionen
;=====
N10 DEF INT KIE_CNTR ; ZAEHLER FÜR ELEMENTE DER KIN. KETTEN
N20 DEF INT RETVAL
;
;=====
; Initialisierung der Kollisionsdaten
;=====
; Alle Parameter auf ihre Grundstellungswerte zuruecksetzen:
N30 RETVAL = DELOBJ("KIN_CHAIN_ELEM")
N40 KIE_CNTR = 0
;
;=====
; Kinematische Kette
;=====
; KE1: OFFSET: Root
; -----
N50 $NK_TYPE[KIE_CNTR] = "OFFSET"
N60 $NK_NAME[KIE_CNTR] = "ROOT"
N70 $NK_NEXT[KIE_CNTR] = "X-AXIS"
N80 KIE_CNTR = KIE_CNTR + 1
;
; -----
; Kinematisches Element: LINEARACHSE: X-Achse
; -----
N90 $NK_TYPE[KIE_CNTR] = "AXIS_LIN"
N100 $NK_NAME[KIE_CNTR] = "X-AXIS"
N110 $NK_NEXT[KIE_CNTR] = "Y-AXIS"
N120 $NK_AXIS[KIE_CNTR] = "X1"
;
N130 $NK_OFF_DIR[KIE_CNTR,0] = 1.0 ; X
N140 KIE_CNTR = KIE_CNTR + 1
;
; -----
; Kinematisches Element: LINEARACHSE: Y-Achse
; -----
N150 $NK_TYPE[KIE_CNTR] = "AXIS_LIN"
N160 $NK_NAME[KIE_CNTR] = "Y-AXIS"
N170 $NK_NEXT[KIE_CNTR] = "Z-AXIS"
N180 $NK_AXIS[KIE_CNTR] = "Y1"
;
N190 $NK_OFF_DIR[KIE_CNTR,1] = 1.0 ; Y
N200 KIE_CNTR = KIE_CNTR + 1
;
; -----
; Kinematisches Element: LINEARACHSE: Z-Achse
; -----
N210 $NK_TYPE[KIE_CNTR] = "AXIS_LIN"
N220 $NK_NAME[KIE_CNTR] = "Z-AXIS"
N230 $NK_NEXT[KIE_CNTR] = "C-AXIS-OFFSET"
N240 $NK_AXIS[KIE_CNTR] = "Z1"
;

```

Programmcode

```

N250 $NK_OFF_DIR[KIE_CNTR,2] = 1.0      ; Z
N260 KIE_CNTR = KIE_CNTR + 1
;
; -----
; Kinematisches Element: OFFSET: C-Achs
; -----
N270 $NK_TYPE[KIE_CNTR] = "OFFSET"
N280 $NK_NAME[KIE_CNTR] = "C-AXIS-OFFSET"
N290 $NK_NEXT[KIE_CNTR] = "C-AXIS"
;
N300 $NK_OFF_DIR[KIE_CNTR,2] = 600.0    ; Z-Richtung
N310 KIE_CNTR = KIE_CNTR + 1 ;
;
; -----
; Kinematisches Element: RUNDACHSE: C-Achse
; -----
N320 $NK_TYPE[KIE_CNTR] = "AXIS_ROT"
N330 $NK_NAME[KIE_CNTR] = "C-AXIS"
N340 $NK_NEXT[KIE_CNTR] = "C-A-OFFSET"
N350 $NK_AXIS[KIE_CNTR] = "C1"
;
N360 $NK_OFF_DIR[KIE_CNTR,2] = 1.0      ; Z-Richtung
N370 KIE_CNTR = KIE_CNTR + 1
;
; -----
; Kinematisches Element: OFFSET: C-A-Achs
; -----
N380 $NK_TYPE[KIE_CNTR] = "OFFSET"
N390 $NK_NAME[KIE_CNTR] = "C-A-OFFSET"
N400 $NK_NEXT[KIE_CNTR] = "A-AXIS"
;
N410 $NK_OFF_DIR[KIE_CNTR,0] = 20.0     ; X-Richtung
N420 $NK_OFF_DIR[KIE_CNTR,1] = -5.0    ; Y-Richtung
N430 KIE_CNTR = KIE_CNTR + 1
;
; -----
; Kinematisches Element: RUNDACHSE: A-Achse
; -----
N440 $NK_TYPE[KIE_CNTR] = "AXIS_ROT"
N450 $NK_NAME[KIE_CNTR] = "A-AXIS"
N460 $NK_NEXT[KIE_CNTR] = "TOOL-REF"
N470 $NK_AXIS[KIE_CNTR] = "A1"
;
N480 $NK_OFF_DIR[KIE_CNTR,0] = 1.0     ; X-Richtung
N490 KIE_CNTR = KIE_CNTR + 1
;
; -----
; Kinematisches Element: OFFSET: Werkzeug-Referenz
; -----
N500 $NK_TYPE[KIE_CNTR] = "OFFSET"
N510 $NK_NAME[KIE_CNTR] = "TOOL-REF"
N520 $NK_NEXT[KIE_CNTR] = "DOCKING_POINT 1"
;
N530 $NK_OFF_DIR[KIE_CNTR,2] = -200.0   ; Z-Richtung
N540 KIE_CNTR = KIE_CNTR + 1
;

```

Programmcode

```

; -----
; Kinematisches Element: Schalter 3/1
; -----
N550 $NK_TYPE[KIE_CNTR] = "SWITCH"
N560 $NK_NAME[KIE_CNTR] = "DOCKING_POINT 1"
N570 $NK_NEXT[KIE_CNTR] = "HEAD 1"
N580 $NK_PARALLE[KIE_CNTR] = "DOCKING_POINT 2"
;
N590 $NK_SWITCH_INDEX[KIE_CNTR] = 3      ; Index 3
N600 $NK_SWITCH_POS[KIE_CNTR] = 1       ; Schalterstellung 1
N610 KIE_CNTR = KIE_CNTR + 1
;
; -----
; Kinematisches Element: Schalter 3/2
; -----
N620 $NK_TYPE[KIE_CNTR] = "SWITCH"
N630 $NK_NAME[KIE_CNTR] = "DOCKING_POINT 2"
N640 $NK_NEXT[KIE_CNTR] = "HEAD 2"
N650 $NK_PARALLE[KIE_CNTR] = "DOCKING_POINT 3"
;
N660 $NK_SWITCH_INDEX[KIE_CNTR] = 3      ; Index 3
N670 $NK_SWITCH_POS[KIE_CNTR] = 2       ; Schalterstellung 2
N680 KIE_CNTR = KIE_CNTR + 1
;
; -----
; Kinematisches Element: Schalter 3/3
; -----
N690 $NK_TYPE[KIE_CNTR] = "SWITCH"
N700 $NK_NAME[KIE_CNTR] = "DOCKING_POINT 3"
N710 $NK_NEXT[KIE_CNTR] = "HEAD 3"
N720 $NK_PARALLE[KIE_CNTR] = ""
N730 $NK_SWITCH_INDEX[KIE_CNTR] = 3      ; Index 3
N740 $NK_SWITCH_POS[KIE_CNTR] = 3       ; Schalterstellung 3
N750 KIE_CNTR = KIE_CNTR + 1
;
; -----
; Kinematisches Element: OFFSET: HEAD 1
; -----
N760 $NK_TYPE[KIE_CNTR] = "OFFSET"
N770 $NK_NAME[KIE_CNTR] = "HEAD 1"
N780 $NK_NEXT[KIE_CNTR] = ""
;
N790 $NK_OFF_DIR[KIE_CNTR,0] = 100.      ; X-Richtung
N800 $NK_OFF_DIR[KIE_CNTR,1] = -30.     ; Y-Richtung
N810 KIE_CNTR = KIE_CNTR + 1
;
; -----
; Kinematisches Element: OFFSET: HEAD 2
; -----
N820 $NK_TYPE[KIE_CNTR] = "OFFSET"
N830 $NK_NAME[KIE_CNTR] = "HEAD 2"
N840 $NK_NEXT[KIE_CNTR] = ""
;
N850 $NK_OFF_DIR[KIE_CNTR,2] = -230.     ; Z-Richtung
N860 KIE_CNTR = KIE_CNTR + 1
;

```

Programmcode

```

; -----
; Kinematisches Element: OFFSET: HEAD 3
; -----
N870 $NK_TYPE[KIE_CNTR] = "OFFSET"
N880 $NK_NAME[KIE_CNTR] = "HEAD 3"
N890 $NK_NEXT[KIE_CNTR] = ""
;
N900 $NK_OFF_DIR[KIE_CNTR,0] = 50.      ; X-Richtung
N910 $NK_OFF_DIR[KIE_CNTR,1] = -20.    ; Y-Richtung
N920 $NK_OFF_DIR[KIE_CNTR,2] = -90.    ; Z-Richtung
N930 KIE_CNTR = KIE_CNTR + 1
;===== ENDE =====

```

5.5 Datenlisten

5.5.1 Maschinendaten

5.5.1.1 NC-spezifische Maschinendaten

Nummer	Bezeichner: \$MN_	Beschreibung
16800	ROOT_KIN_ELEM_NAME	Name des ersten Elements der aktiven kinematischen Kette
18880	MM_MAXNUM_KIN_CHAIN_ELEM	Maximale Anzahl Elemente für kinematische Ketten
18882	MM_MAX_NUM_KIN_SWITCHES	Maximale Anzahl Schalter für kinematische Ketten

5.5.2 Systemvariablen

Bezeichner	Beschreibung
\$NK_NAME	Name des kinematischen Elements
\$NK_NEXT	Name des nächsten Elements in der kinematischen Kette
\$NK_PARALLEL	Name des ersten Elements einer vor dem Element parallel abzweigenden kinematischen Kette
\$NK_TYPE	Typ des kinematischen Elements
\$NK_OFF_DIR	Abhängig von \$NK_TYPE: Verschiebungs- oder Richtungsvektor
\$NK_AXIS	Name der zugeordneten Maschinenachse bzw. Objektname
\$NK_A_OFF	Nullpunktverschiebung im Element bei Linear - bzw. Rundachsen
\$NK_SWITCH_INDEX	Index i, über den der Schalter durch die Systemvariable \$NK_SWITCH[<i>] angesprochen wird
\$NK_SWITCH_POS	Schalterstellung EIN
\$NK_SWITCH	Schaltvariable

K8: Geometrische Maschinenmodellierung

6.1 Funktionsbeschreibung

6.1.1 Merkmale

Im vorliegenden Kapitel wird beschrieben, wie für NC-Funktionen wie z. B. die "Kollisionsvermeidung" die Geometrie von Maschinenteilen über Schutzbereiche abgebildet und in der Steuerung über Systemvariable parametrisiert wird.

Die Systemvariablen werden in der NC remanent gespeichert und können über SINUMERIK Operate mittels Inbetriebnahme-archiv als "NC-Daten" archiviert bzw. eingelesen werden.

Durch Zuordnung eines Schutzbereichs zu einem Element der im vorhergehenden Kapitel beschriebenen kinematischen Kette (Seite 201) wird die Lage und Bewegung des Maschinenteils innerhalb des Maschinenraums eindeutig beschrieben.

Hinweis

Grafischer Editor

Die Maschinenmodellierung kann, alternativ zum Schreiben der Systemvariablen in einem Teileprogramm, über die Bedienoberfläche SINUMERIK Operate erfolgen:

Bedienbereich: "Inbetriebnahme" > "NC" > "Maschinenmodell"

Änderungen am Maschinenmodell

Direkt an den Systemvariablen vorgenommene Änderungen am Maschinenmodell werden erst nach einer expliziten Anforderung zum Neuberechnen des Maschinenmodells durch Aufruf der Funktion PROTA() (Seite 305) bzw. PROTS() (Seite 306) an der Bedienoberfläche sichtbar.

Über die Bedienoberfläche vorgenommene Änderungen am Maschinenmodell werden sofort in die Systemvariablen der NC übernommen. Die Änderungen werden aber erst nach einer expliziten Anforderung zum Neuberechnen des Maschinenmodells durch Aufruf der Funktion PROTA() (Seite 305) bzw. PROTS() (Seite 306) aktiv.

Schutzbereich

Das zentrale Element der geometrischen Maschinenmodellierung sind die Schutzbereiche. Durch einen Schutzbereich werden die geometrischen Abmessungen eines Maschinenteils, sein Bezug zur kinematischen Kette, sowie weitere allgemeine Eigenschaften beschrieben.

Ein Schutzbereich hat folgende Parameter:

- Name des Schutzbereichs
- Name des kinematischen Elements, dem der Schutzbereich zugeordnet ist
- Typ des Schutzbereichs

- Name des ersten Schutzbereichselements
- Farbe und Transparenz des Schutzbereichs
- Detaillierungsgrad des Schutzbereichs
- Nummer des NC/PLC-Nahtstellenbits des Schutzbereichs
- Initialisierungsstatus des Schutzbereichs
- Adresse der Geometriedaten des zu schützenden Maschinenelements
(nur relevant bei automatischen Schutzbereichen)

Jeder Parameter wird durch eine Systemvariable abgebildet. Die einzelnen Parameter bzw. Systemvariablen sind im Kapitel "Systemvariablen: Schutzbereiche (Seite 245)" ausführlich beschrieben.

Kinematische Kette

Um die Lage und Bewegung eines Maschinenteils abzubilden, wird der entsprechende Schutzbereich einem Element der kinematischen Kette (Seite 201) zugeordnet. Die Geometriedaten des Schutzbereichs beziehen sich dann auf das lokale Koordinatensystem dieses kinematischen Elements.

Typ des Schutzbereichs

Es gibt folgende Typen von Schutzbereichen:

- Maschinenschutzbereiche (Typ: "MACHINE")
Maschinenschutzbereiche dienen der allgemeinen Maschinenmodellierung. Über sie werden feststehende und bewegliche Maschinenteile abgebildet, deren Geometrie einmalig bei der Inbetriebnahme definiert wird und sich im Betrieb der Maschine nicht mehr verändert.
- Automatische Werkzeugschutzbereiche (Typ: "TOOL")
Automatische Werkzeugschutzbereiche dienen der Abbildung von Werkzeugen. Die Geometrie des Werkzeugs wird dabei nicht direkt angegeben, sondern bei Aktivieren des Werkzeugs von der Steuerung automatisch erzeugt.
Siehe Kapitel "Automatische Werkzeugschutzbereiche (Seite 238)".

Schutzbereichselement

Mit einem Schutzbereichselement wird das verwendete geometrische Element in seinen geometrischen und allgemeinen Eigenschaften beschrieben.

Ein Schutzbereichselement hat folgende Parameter:

- Name des Schutzbereichselements
- Name des nachfolgenden Schutzbereichselements
- Name des nachfolgenden, zu \$NP_NEXT parallelen Schutzbereichselements
- Farbe und Transparenz des Schutzbereichselements
- Detaillierungsgrad des Schutzbereichselements
- Verwendungsart des Schutzbereichselements
- Typ des Schutzbereichselements

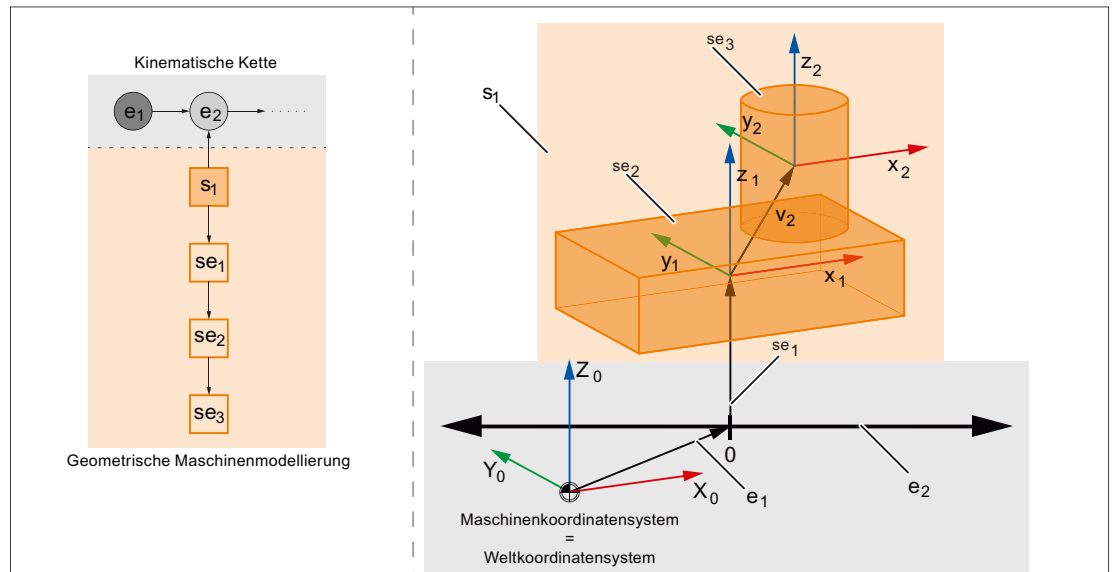
- Dateiname der STL-Datei, welche die Geometriedaten des Schutzbereichselements enthält
(nur relevant bei Typ "FILE")
- Geometrische Parameter des Schutzbereichskörpers
(nur relevant bei Typ "BOX", "SPHERE" oder "CYLINDER")
- Verschiebungsvektor des lokalen Koordinatensystems des Schutzbereichselements
- Richtungsvektor für die Drehung des lokalen Koordinatensystems des Schutzbereichselements
- Winkel für die Drehung des lokalen Koordinatensystems des Schutzbereichselements

Jeder Parameter wird durch eine Systemvariable abgebildet. Die einzelnen Parameter bzw. Systemvariablen sind ausführlich beschrieben in:

- Kapitel "Systemvariablen: Schutzbereichselemente für Maschinenschutzbereiche (Seite 257)"
- Kapitel "Systemvariablen: Schutzbereichselemente für automatische Werkzeugenschutzbereiche (Seite 279)"

Schutzbereich, Schutzbereichselemente und kinematische Kette

Das nachfolgende Bild zeigt anhand eines beispielhaften Schutzbereichs mit zwei Schutzbereichselementen den Zusammenhang eines Schutzbereichs, seiner Schutzbereichselemente und der Zuordnung zu einem Element der kinematischen Kette.



- e_1 Kinematisches Element 1, Typ "OFFSET", konstante Verschiebung
- e_2 Kinematisches Element 2, Typ "AXIS_LIN", Maschinenachse AX1
- s_1 Schutzbereich
- se_1 Schutzbereichselement 1, Typ "FRAME", Verschiebung
- se_2 Schutzbereichselement 2, Typ "BOX"
- se_3 Schutzbereichselement 3, Typ "CYLINDER"

Bild 6-1 Schutzbereich, Schutzbereichselemente und kinematische Kette

6.1.2 Automatische Werkzeugschutzbereiche

Im Gegensatz zu Maschinenschutzbereichen, deren Geometrie einmal während der Maschinenmodellierung definiert wird und sich dann nicht mehr verändert, kann sich die Geometrie eines Werkzeugschutzbereichs mit jedem Werkzeugwechsel ändern. Daher wird bei der Erstellung des Maschinenmodells die Geometrie eines automatischen Werkzeugschutzbereichs nicht direkt beschrieben, sondern die Adresse (Magazinnummer, Magazinplatz, etc.) angegeben, unter der die Werkzeugdaten abgelegt sind. Von der Steuerung werden dann automatisch folgende Aktionen ausgeführt:

1. Die Werkzeugmodellierung (siehe unten Absatz "Werkzeugmodellierung") erzeugt eine STL-Datei .
2. Ein Schutzbereichselement vom Typ "FILE" wird erzeugt und die STL-Datei zugeordnet.
3. Dem Schutzbereich (Typ "TOOL") wird das Schutzbereichselement zugeordnet.

Werkzeugdefinition unabhängig von der Einbaulage des Werkzeugs

Im Normalfall bleibt bei der Definition eines Werkzeugschutzbereichs der Parameter "\$NP_1ST_PROT (Seite 249)" leer. Der Name des Schutzbereichselements wird erst bei Aktivierung des Werkzeugs von der Steuerung eingetragen (siehe oben).

Damit eine Werkzeugdefinition unabhängig von der Einbaulage des Werkzeugs erfolgen kann, kann über den Parameter "\$NP_1ST_PROT" ein Schutzbereichselement vom Typ "FRAME (Seite 265)" (Transformationselement) zugeordnet werden. Über dieses zusätzliche Element können die Transformationen zum Ausrichten des Werkzeugs vorgenommen werden. Bei Aktivierung eines Werkzeugs wird von der Steuerung der Name des internen Schutzbereichselements in den Parameter "\$NP_NEXT (Seite 259)" des Transformationselements eingetragen.

Folgende Regeln sind einzuhalten:

- Das Transformationselement darf nur vom Typ "FRAME" sein.
- Pro Werkzeugschutzbereich darf nur ein Transformationselement verwendet werden.
- Der Parameter "\$NP_NEXTP" des Transformationselements wird nicht ausgewertet.

Werkzeugbezugspunkt

Die Lage des Werkzeugbezugspunkts im Maschinenmodell wird durch das kinematische Element, dem der Werkzeugschutzbereich zugeordnet ist, festgelegt. Zusätzlich kann der Werkzeugbezugspunkt innerhalb des Werkzeugschutzbereichs durch ein optionales Transformationselement verschoben werden.

Kinematische Transformationen

Bei der Definition einer kinematischen Transformation darf der Werkzeugbezugspunkt nur über die kinematische Kette festgelegt werden. Verschiebungen durch das Transformationselement des Werkzeugschutzbereichs werden nicht berücksichtigt.

ACHTUNG
Festlegung des Werkzeugbezugspunkts bei kinematischen Transformationen
Verschiebungen des Werkzeugbezugspunktes durch das Transformationselement des Werkzeugschutzbereichs werden von kinematischen Transformationen nicht berücksichtigt.

Werkzeugmodellierung

Das Modell eines Werkzeugs wird von der Steuerung heuristisch aus den Werkzeugdaten erstellt. Die dazu verwendeten Werkzeugdaten (L1, L2, L3, R), sind dabei immer die resultierenden Gesamtabmessungen der einzelnen Komponenten, z.B. Länge plus Verschleiß, wie sie auch zur Werkzeugkorrektur in der Programmbearbeitung eingehen.

Programmierbare Werkzeugkorrekturen

Programmierbare Werkzeugkorrekturen wie z.B. `OFFEN` (Aufmaß zur programmierten Kontur) werden nicht berücksichtigt, da sie sich in jedem Satz, auch ohne Werkzeugwechsel, ändern können.

Werkzeugtypabhängige Modellbildung

Bei der Modellbildung wird zwischen folgenden Werkzeugtypen unterschieden:

- Fräswerkzeuge und alle anderen Werkzeuge, die weder Dreh- noch Schleifwerkzeuge sind
 - Modellierung
Das Werkzeug wird durch einen Zylinder mit der Höhe L1 und dem Radius R modelliert. Bei negativer Länge L1 wird für die Zylinderhöhe der Betrag von L1 verwendet. Das Vorzeichen von L1 wird bei der Positionierung des Zylinders im Maschinenmodell berücksichtigt. Die Zylinderachse liegt parallel zu L1.
Werkzeugtyp 110 (Kugelkopffräser) und 111 (Stirnfräser) werden mittels Halbkugel bzw. Kugelsegmenten modelliert.
Bei negativem Radius wird der Betrag des Radius verwendet. Bei einem Wert für den Radius kleiner 1/3 mm, wird ein Radius von 1/3 mm verwendet.
 - Positionierung
Die Positionierung des Zylinders im Maschinenmodell erfolgt anhand der Werkzeuglängskomponenten L2 und L3.
Bei Fräswerkzeugen (Werkzeugtyp 100 ... 199) und Bohrwerkzeugen (Werkzeugtyp 200 ... 299) an Drehmaschinen erfolgt die Positionierung des Zylinders anhand der Schneidenlage.
Voraussetzung: Schneidenlage == 5 ... 8
- Schleifwerkzeuge
 - Modellierung
Schleifwerkzeuge (Schleifscheibe, Werkzeugtyp 400 ... 499) werden durch einen Zylinder mit der Werkzeuglänge als Radius und dem doppelten Werkzeugradius als Höhe modelliert.
 - Positionierung
Die Positionierung des Zylinders im Maschinenmodell erfolgt anhand der Werkzeuglängen L1, L2 und L3.
- Drehwerkzeuge
Bei Drehwerkzeuge werden im Maschinenmodell nur die Schneidplatten berücksichtigt, nicht aber deren Verbindung zum Werkzeugbezugspunkt.
Bei der Modellierung einer Schneidplatte werden folgende Daten berücksichtigt:
 - Werkzeugtyp
 - Schneidenlage
 - Schneidenradius
 - Freiwinkel
 - Halterwinkel
 - Plattenlänge
 - Plattenbreite
 - Plattendicke (Annahme: Plattendicke = 10% Plattenlänge)

Werkzeugmodell

Ein Werkzeug wird standardmäßig mit einer Genauigkeit von einem Drittel der Kollisionstoleranz (Seite 295) modelliert. Die Geometriedaten des modellierten Werkzeugs werden in einer internen Datei im STL-Format abgelegt:

- Verzeichnis: `_N_PROT3D_DIR/_N_TOOL_DIR`
- Bezeichnung: Namen des zugehörigen Schutzbereichs mit Präfix `_N_` und Endung `_STL`

Das Koordinatensystem der Geometriedaten hat sein Ursprung immer in dem Punkt, von dem aus die Werkzeuglängenkorrekturen zur Werkzeugspitze hin zeigen.

Systemvariable

Alle Parameter eines automatischen Werkzeugschutzbereichs können über Systemvariable (Seite 257) gelesen werden.

6.2 Inbetriebnahme

6.2.1 Allgemein

6.2.1.1 Übersicht

Die Inbetriebnahme der Funktion "Kollisionsvermeidung" erfolgt mittels:

- Maschinendaten
 - Vorgaben zum Mengengerüst von Schutzbereichen, Schutzbereichselementen, NC/PLC-Nahstellensignalen, Dreiecken zur Geometriemodellierung
 - Erzeugungsmodus des Maschinenmodells
 - Erzeugungsmodus für automatische Werkzeugschutzbereiche
- Systemvariablen
 - Parametrierung der Schutzbereiche
 - Parametrierung der Schutzbereichselemente eines Schutzbereichs

6.2.1.2 Aufbau der Systemvariablen

Die Systemvariablen sind nach folgendem Schema aufgebaut:

- `$NP_<Name>[<Index_1>]`
- `$NP_<Name>[<Index_1>, <Index_2>]`

Allgemein

Die Systemvariablen zur Beschreibung von Schutzbereichen bzw. Schutzbereichselemente haben folgende Eigenschaften:

- Präfix: **\$NP_**, (N für NC, P für Protection).
- Sie sind über NC-Programme les- und schreibbar.
- Sie können über Archive gesichert und wieder in die NC eingelesen werden.

Datentyp

STRING

Alle Systemvariablen vom Datentyp STRING haben folgende Eigenschaften:

- Maximale String-Länge: 31 Zeichen
- Es wird keine Unterscheidung zwischen Groß- und Kleinschreibung gemacht
Beispiel: "Achse1" identisch mit "ACHSE1"
- Leer- und Sonderzeichen sind zulässig
Beispiel: "Achse1" nicht identisch mit " Achse 1"
- Namen, die mit **zwei** Unterstrichen "__" **beginnen**, sind für Systemzwecke reserviert und dürfen **nicht** für anwenderdefinierte Namen verwendet werden.

Hinweis

Führendes Leerzeichen

Da Leerzeichen gültige und der Unterscheidung dienende Zeichen sind, dürfen Namen, die mit einem **Leerzeichen**, gefolgt von **zwei** Unterstrichen "__" **beginnen**, prinzipiell für anwenderdefinierte Namen verwendet werden. Aufgrund der Verwechslungsgefahr mit Systemnamen wird dieses Vorgehen **nicht** empfohlen.

Index_1

Systemvariablen für Schutzbereiche

Über Index_1 werden die einzelnen Schutzbereiche adressiert. Index 0 → 1. Schutzbereich, Index 1 → 2. Schutzbereich, ... m → (m+1) Schutzbereich, mit m = (\$MN_MM_MAXNUM_3D_PROT_AREAS - 1)

Alle Systemvariablen eines Schutzbereichs haben den gleichen Index.

Systemvariablen für Schutzbereichselemente

Über Index_1 werden die einzelnen Schutzbereichselemente adressiert. Index 0 → 1. Schutzbereichselement, Index 1 → 2. Schutzbereichselement, ... n → (n+1)tes Schutzbereichselement, mit n = (\$MN_MM_MAXNUM_3D_PROT_AREA_ELEM - 1)

Alle Systemvariablen eines Schutzbereichselementes haben den gleichen Index.

Index_2

Index_2 hat, abhängig von der jeweiligen Systemvariablen, unterschiedliche Bedeutungen.

Siehe auch

Löschen von Komponenten (DELOBJ) (Seite 225)

6.2.1.3 Farbtafel

Die nachfolgende Farbtafel bietet ein Überblick über die RGB-Farbwerte und die dazugehörige Farbe. Ein RGB-Farbwert besteht aus 3 Bytes. Ein Byte pro Farbe:

3. Byte	2. Byte	1. Byte
Farbwert für Rot	Farbwert für Grün	Farbwert für Blau
0 - 255 _D bzw. 0 - FF _H	0 - 255 _D bzw. 0 - FF _H	0 - 255 _D bzw. 0 - FF _H



6.2.2 Maschinendaten

6.2.2.1 Maximale Anzahl von Schutzbereichen

Mit dem Maschinendatum wird die maximale Anzahl über alle Typen von parametrierbaren Schutzbereichen (Seite 248) festgelegt.

MD18890 \$MN_MM_MAXNUM_3D_PROT_AREAS = <Anzahl>

6.2.2.2 Maximale Anzahl von Schutzbereichselementen für Maschinenschutzbereiche

Mit dem Maschinendatum wird die maximale Anzahl parametrierbarer Schutzbereichselemente für Maschinenschutzbereiche (\$NP_PROT_TYPE == "MACHINE" (Seite 248)) festgelegt.

MD18892 \$MN_MM_MAXNUM_3D_PROT_AREA_ELEM = <Anzahl>

6.2.2.3 Maximale Anzahl von Schutzbereichselementen für automatische Werkzeugschutzbereiche

Mit dem Maschinendatum wird die maximale Anzahl von Schutzbereichselementen für automatische Werkzeugschutzbereiche festgelegt. Da pro automatischem Werkzeugschutzbereich von der Steuerung immer genau ein Schutzbereichselement erzeugt wird, wird über den hier parametrisierte Wert gleichzeitig auch die maximal möglichen Anzahl von parametrierbaren automatischen Werkzeugschutzbereichen (\$NP_PROT_TYPE == "TOOL" (Seite 248)) festgelegt.

MD18893 \$MN_MM_MAXNUM_3D_T_PROT_ELEM = <Anzahl>

6.2.2.4 Maximale Anzahl von NC/PLC-Nahtstellensignalen zur Voraktivierung von Schutzbereichen

Mit dem Maschinendatum wird der Steuerung die Anzahl der tatsächlich verwendeten NC/PLC-Nahtstellensignale der Schnittstelle DB10, DBX234.0 - DBX241.7 (Kollisionsvermeidung: Schutzbereich aktivieren) mitgeteilt. Die Anzahl der verwendeten Nahtstellensignale erhöht den Speicherplatzbedarf pro Teileprogrammsatz. Die Zählung der Anzahl verwendeter NC/PLC-Nahtstellensignale beginnt bei DB10, DBX234.0

MD18897 \$MN_MM_MAXNUM_3D_INTERFACE_IN = <Anzahl>

Literatur:

Eine ausführliche Beschreibung der Nahtstellensignale findet sich im Listenhandbuch NC-Variablen und Nahtstellensignale.

6.2.2.5 Maximale Anzahl von Dreiecken für Maschinenschutzbereiche

Mit dem Maschinendatum wird die maximal von der Steuerung zur Verfügung zu stellende Anzahl von Dreiecken für Schutzbereichskörper (\$NP_TYPE == "FILE" (Seite 265)) von Maschinenschutzbereichen (\$NP_PROT_TYPE == "MACHINE" (Seite 248)) festgelegt.

MD18895 \$MN_MM_MAXNUM_3D_FACETS = <Anzahl>

6.2.2.6 Maximale Anzahl von Dreiecken für automatisch Werkzeugschutzbereiche

Mit dem Maschinendatum wird die maximal von der Steuerung zur Verfügung zu stellende Anzahl von Dreiecken für Schutzbereichskörper von automatischen Werkzeugschutzbereichen festgelegt.

MD18894 \$MN_MM_MAXNUM_3D_FACETS_INTERN = <Anzahl>

Die Schutzbereichskörper werden von der Steuerung automatisch anhand der Geometriedaten des zum Erzeugungszeitpunkts aktiven Werkzeugs modelliert. Der Dreiecksbedarf ist dabei umso größer:

- je höher die geometrische Komplexität des Werkzeugs ist.
- je kleiner die parametrisierte Kollisionstoleranz ist.

Hinweis

Schutzbereichskörper und Kollisionstoleranz

Die Schutzbereichskörper von automatischen Werkzeugschutzbereichen werden von der Steuerung standardmäßig mit einer Genauigkeit von 1/3 der Kollisionstoleranz (Seite 295) erzeugt.

6.2.2.7 Erzeugungsmodus für automatische Werkzeugschutzbereiche

Mit dem Maschinendatum wird festgelegt, wie die Steuerung Schutzbereichskörper von automatischen Werkzeugschutzbereichen erzeugt.

MD18899 \$MN_PROT_AREA_TOOL_MASK = <Mode>

<Mode>		
Bit	Wert	Bedeutung
0	0	Heuristische Modellbildung nicht anwenden
	1	Heuristische Modellbildung mittels Werkzeugdaten

6.2.3 Systemvariablen: Schutzbereiche

6.2.3.1 Übersicht

Mit folgenden Systemvariablen wird ein Schutzbereich parametrisiert:

Name	Bedeutung
\$NP_PROT_NAME	Name des Schutzbereichs
\$NP_CHAIN_ELEM	Name des kinematischen Elements, dem der Schutzbereich zugeordnet ist
\$NP_PROT_TYPE	Typ des Schutzbereichs
\$NP_1ST_PROT	Name des ersten Schutzbereichselements
\$NP_PROT_COLOR	Farbe und Transparenz des Schutzbereichs.

Name	Bedeutung
\$NP_PROT_D_LEVEL	Detaillierungsgrad des Schutzbereichs
\$NP_BIT_NO	Nummer des NC/PLC–Nahtstellenbits des Schutzbereichs
\$NP_INIT_STAT	Initialisierungsstatus des Schutzbereichs
\$NP_INDEX	Adresse der Geometriedaten des zu schützenden Maschinenelements (nur relevant bei automatischen Schutzbereichen)

Die Systemvariablen sind in den nachfolgenden Kapiteln ausführlich beschrieben.

Hinweis

Definierten Ausgangszustand herstellen

Es wird empfohlen, vor Parametrierung der Schutzbereiche einen definierten Ausgangszustand zu erzeugen. Dazu sind die Systemvariablen der Schutzbereiche mit der Funktion DELOBJ() (Seite 225) auf ihren Default-Wert zu setzen.

Ändern von Systemvariablenwerten

Wird der Wert einer der oben aufgeführten Systemvariablen geändert, wird die Änderung auf der Bedienoberfläche, z.B. SINUMERIK Operate, sofort sichtbar. Das Maschinenmodell der NC wird aber erst nach einer expliziten Anforderung zum Neuberechnen des Maschinenmodells durch Aufruf der Funktion PROTA() (Seite 305) bzw. PROTS() (Seite 306) aktualisiert.

6.2.3.2 \$NP_PROT_NAME

Funktion

In die Systemvariable ist der NC-weit eindeutige Name des Schutzbereichs einzutragen. Über diesen Namen wird der Schutzbereich, z.B. von einem Schutzbereichselement, referenziert. Der Name wird auch im grafischen Editor von SINUMERIK Operate angezeigt.

Syntax

\$NP_PROT_NAME [<m>] = "<Name>"

Bedeutung

\$NP_PROT_NAME:	Name des Schutzbereichs	
	Datentyp:	STRING
	Defaultwert:	"" (Leerstring)
<m>:	Systemvariablen- bzw. Schutzbereichsindex	
	Datentyp:	INT
	Wertebereich:	0, 1, 2, ... (\$MN_MM_3D_MAXNUM_PROT_AREAS - 1)
<Name>:	Name des Schutzbereichs	
	Datentyp:	STRING

Beispiel

Dem 6. Schutzbereich wird der Name "Spindel" zugewiesen:

Programmcode	Kommentar
N100 \$NP_PROT_NAME[5] = "Spindel"	; 6. Schutzbereich, ; Name = "Spindel"

6.2.3.3 \$NP_CHAIN_ELEM**Funktion**

In die Systemvariable ist der Name des kinematischen Elements (Seite 209) einzutragen, mit dem der Schutzbereich verbunden wird.

Hinweis**Bezugskordinatensystem**

Die Geometriedaten des Schutzbereichs, ausgehend vom ersten Schutzbereichselement (\$NP_1ST_PROT (Seite 249)), beziehen sich auf das lokale Koordinatensystem des kinematischen Elements, mit dem der Schutzbereich verbunden wird.

Syntax

```
$NP_CHAIN_ELEM[<m>] = "<Name>"
```

Bedeutung

\$NP_CHAIN_ELEM:	Name des kinematischen Elements, mit dem der Schutzbereich verbunden wird	
	Datentyp:	STRING
	Defaultwert:	"" (Leerstring)
<m>:	Systemvariablen- bzw. Schutzbereichsindex	
	Datentyp:	INT
	Wertebereich:	0, 1, 2, ... (\$MN_MM_3D_MAXNUM_PROT_AREAS - 1)
<Name>:	Name des kinematischen Elements	
	Datentyp:	STRING
	Wertebereich:	In \$NK_NAME (Seite 209) parametrisierte Namen

Beispiel

Der 6. Schutzbereich wird mit dem kinematischen Element mit dem Namen "Z-Achse" verbunden:

Programmcode	Kommentar
N100 \$NP_CHAIN_ELEM[5] = "Z-Achse"	; 6. Schutzbereich, ; Name des kin. Elements: "Z-Achse"

6.2.3.4 \$NP_PROT_TYPE

Funktion

In die Systemvariable ist der Typ des Schutzbereichs einzutragen:

- Maschinen-Schutzbereich: "MACHINE"
Der Schutzbereichskörper wird durch ein oder mehrere Schutzbereichselemente definiert. \$NP_1ST_PROT (Seite 249) verweist auf das erste Schutzbereichselement.
- Automatischer Werkzeug-Schutzbereich: "TOOL"
Die Steuerung berechnet die Abmessungen des Schutzbereichskörpers aus den Werkzeugdaten. \$NP_INDEX (Seite 255) verweist auf das Werkzeug.
- Spannmittel-Schutzbereich: "FIXTURE"
- Werkstück-Schutzbereich: "WORKPIECE"

Syntax

\$NP_PROT_TYPE [<m>] = "<Type>"

Bedeutung

\$NP_PROT_TYPE:	Typ des Schutzbereichs	
	Datentyp:	STRING
	Wertebereich:	"MACHINE", "TOOL", "FIXTURE", "WORKPIECE"
	Defaultwert:	"" (Leerstring)
<m>:	Systemvariablen- bzw. Schutzbereichsindex	
	Datentyp:	INT
	Wertebereich:	0, 1, 2, ... (\$MN_MM_3D_MAXNUM_PROT_AREAS - 1)
<Type>:	Typ	
	Datentyp:	STRING

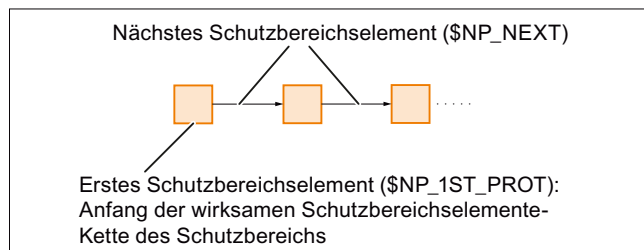
Beispiel

Der 6. Schutzbereich ist ein Maschinenschutzbereich:

Programmcode	Kommentar
N100 \$NP_PROT_TYPE[5] = "MACHINE"	; 6. Schutzbereich, ; Typ = "MACHINE"

6.2.3.5 \$NP_1ST_PROT**Funktion**

In die Systemvariable ist der Name des ersten Schutzbereichselements (Seite 258) des Schutzbereichs einzutragen.

**Syntax**

```
$NP_1ST_PROT [<m>] = "<Name>"
```

Bedeutung

\$NP_1ST_PROT:	Name des ersten Schutzbereichselements des Schutzbereichs	
	Datentyp:	STRING
	Wertebereich:	In \$NP_NAME (Seite 258) parametrisierte Namen
	Defaultwert:	"" (Leerstring)
<m>:	Systemvariablen- bzw. Schutzbereichsindex	
	Datentyp:	INT
	Wertebereich:	0, 1, 2, ... (\$MN_MM_3D_MAXNUM_PROT_AREAS - 1)
<Name>:	Schutzbereichsname	
	Datentyp:	STRING

Automatische Werkzeugschutzbereiche, \$NP_PROT_TYPE == "TOOL"

Bei automatischen Werkzeugschutzbereichen sind für \$NP_1ST_PROT nur folgende Werte zulässig:

- "" (Leerstring)
- Name eines Schutzbereichselements vom Typ "FRAME"

Verhalten bei Wert == "" (Leerstring)

Bei der Aktivierung des zugehörigen Werkzeugs, wird für das Werkzeug von der Steuerung ein Schutzbereichselement mit einem eindeutigen internen Namen und einem aus den Geometriedaten des Werkzeugs generierten Schutzbereichskörper erzeugt. Der Name wird der Systemvariablen \$NP_1ST_PROT zugewiesen.

- Schutzbereich "TOOL" : \$NP_1ST_PROT = "<interner Name>"

Die Koordinaten des Werkzeugschutzbereichs beziehen sich auf das lokale Koordinatensystem des kinematischen Elements dem er zugeordnet ist.

Verhalten bei Wert == Name eines Schutzbereichselements vom Typ "FRAME"

Bei der Aktivierung des zugehörigen Werkzeugs, wird für das Werkzeug von der Steuerung ein Schutzbereichselement mit einem eindeutigen internen Namen und einem aus den Geometriedaten des Werkzeugs generierten Schutzbereichskörper erzeugt. Der Name wird der Systemvariablen \$NP_NEXT des Schutzbereichselements vom Typ "FRAME" zugewiesen, auf das \$NP_1ST_PROT verweist.

- Schutzbereich "TOOL" : \$NP_1ST_PROT = "WKZ_Frame" →
 - Schutzbereichselement "WKZ_Frame" : \$NP_NEXT = "<interner Name>"

Die Koordinaten des Werkzeugschutzbereichs beziehen sich auf das lokale Koordinatensystem des Schutzbereichselement vom Typ "FRAME".

Anwendungsmöglichkeit: Werkzeugdefinition unabhängig von der Einbaulage an der Maschine.

Beispiel

Das 1. Schutzbereichselement, aus dem sich der 6. Schutzbereich zusammensetzt, hat den Namen "Spindelkasten":

Programmcode	Kommentar
N100 \$NP_1ST_PROT[5] = "Spindelkasten"	; 6. Schutzbereich, ; 1. Schutzbereichselement = "Spindelkasten"

6.2.3.6 \$NP_PROT_COLOR

Funktion

In die Systemvariable ist der Schutzbereichs-spezifische Wert für Alpha/Transparenz- und Farbe (ARGB) einzutragen. Dieser Wert wird für die Darstellung des Schutzbereichs bzw. der Schutzbereichselemente auf der Bedienoberfläche verwendet. Ist für ein Schutzbereichselement in \$NP_COLOR (Seite 261) ein eigener Wert eingetragen, wird dieser für die Darstellung des Schutzbereichselements verwendet.

Aufbau

Alpha/Transparenz- und Farbwert werden als Doppelwort im Hex-Format angegeben:

AARRGGBB_H

- 1. - 3. Byte: RGB-Farbwert. Siehe Kapitel "Farbtafel (Seite 243)".
- 4. Byte: Alpha-Kanal- bzw. Transparenzwert

	Byte	Bedeutung	Wertebereich
BB	1	Blau	0 - 255 _D bzw. 0 - FF _H
GG	2	Grün	
RR	3	Rot	
AA	4	Alpha-Kanal bzw. Transparenz ¹⁾	
1) 0 = transparent bzw. nicht sichtbar, 255 _D = FF _H = nicht transparent bzw. massiv			

Syntax

\$NP_PROT_COLOR[<m>] = <Wert>

Bedeutung

\$NP_PROT_COLOR:	Alpha/Transparenz- und Farbwert des Schutzbereichs	
	Datentyp:	DWORD
	Wertebereich:	00000000 _H - FFFFFFFF _H
	Defaultwert:	00000000 _H (Schwarz, nicht sichtbar)
<m>:	Systemvariablen- bzw. Schutzbereichsindex	
	Datentyp:	INT
	Wertebereich:	0, 1, 2, ... (\$MN_MM_3D_MAXNUM_PROT_AREAS - 1)
<Wert>:	Transparenz- und Farbwert	
	Datentyp:	DWORD

Beispiel

Der 6. Schutzbereich soll an der Bedienoberfläche halbtransparent und in einem grün-blauen Farbton dargestellt werden:

- AA = 7F_H = 127_D $\hat{=}$ 50% Transparenz
- RR (Rot) = 00 $\hat{=}$ kein roter Farbanteil
- GG (Grün) = FF_H = 255_D $\hat{=}$ 100% grüner Farbanteil
- BB (Blau) = 33_H = 51_D $\hat{=}$ 20% blauer Farbanteil

Programmcode	Kommentar
N100 \$NP_PROT_COLOR[5] = 'H7F00FF33'	; 6. Schutzbereich, ; Alpha/Transparenz- und Farbwert = H7F00FF33

6.2.3.7 \$NP_PROT_D_LEVEL

Funktion

Über die Systemvariable wird festgelegt, ab welchem Detaillierungsgrad der Schutzbereich bzw. die Schutzbereichselemente auf der Bedienoberfläche angezeigt werden. Ist für ein Schutzbereichselement in \$NP_D_LEVEL (Seite 263) ein eigener Wert eingetragen, wird dieser bei der Darstellung des Schutzbereichselements verwendet.

Detaillierungsgrad

- Niedrigster Detaillierungsgrad: 0
- Höchster Detaillierungsgrad: 3

Ist am HMI der Detaillierungsgrad x für die Anzeige angewählt, werden alle Schutzbereiche und Schutzbereichselemente angezeigt, für deren Detaillierungsgrad D gilt: $D \leq x$

Syntax

`$NP_PROT_D_LEVEL[<m>] = <Wert>`

Bedeutung

\$NP_PROT_D_LEVEL:	Detaillierungsgrad des Schutzbereichs	
	Datentyp:	INT
	Wertebereich:	$0 \leq D \leq 3$
	Defaultwert:	0
<m>:	Systemvariablen- bzw. Schutzbereichsindex	
	Datentyp:	INT
	Wertebereich:	0, 1, 2, ... (\$MN_MM_3D_MAXNUM_PROT_AREAS - 1)
<Wert>:	Detaillierungsgrad	
	Datentyp:	INT

Beispiel

Der 6. Schutzbereich soll ab Detaillierungsgrad 3 dargestellt werden:

Programmcode	Kommentar
N100 \$NP_PROT_D_LEVEL[5] = 3	; 6. Schutzbereich, ; Detaillierungsgrad = 3

6.2.3.8 \$NP_BIT_NO

Funktion

In die Systemvariable \$NP_BIT_NO ist die Bit-Nummer (0, 1, 2, ... 63) des NC/PLC-Nahtstellensignals einzutragen, mit dem der Schutzbereich verbunden ist. Soll der Schutzbereich mit keinem NC/PLC-Nahtstellensignal verbunden sein, ist der Wert -1 einzutragen.

NC/PLC-Nahtstelle

Über NC/PLC-Nahtstellensignale kann vom PLC-Anwenderprogramm die Aktivierung / Deaktivierung des Schutzbereichs angefordert werden bzw. erfolgt die Rückmeldung des aktuellen Status an das PLC-Anwenderprogramm:

- Anforderung: DB10, DBX234.0 - DBX241.7
- Rückmeldung: DB10, DBX226.0 - DBX233.7

Voraussetzung

Damit das zugeordnete NC/PLC-Nahtstellensignal des Schutzbereichs beachtet wird, muss der Status des Schutzbereich "voraktiviert" bzw. "PLC-gesteuert" sein:

\$NP_INIT_STAT (Seite 254) == "P" (Voraktiviert bzw. PLC-gesteuert)

Syntax

\$NP_BIT_NO[<m>] = <Bit-Nummer>

Bedeutung

\$NP_BIT_NO:	Bit-Nummer des Nahtstellensignals zum Aktivieren / Deaktivieren des Schutzbereichs	
	Datentyp:	INT
	Wertebereich:	-1, 0, 1, 2, ... (\$MN_MM_MAXNUM_3D_INTERFACE_IN - 1)
	Defaultwert:	-1 (kein Nahtstellensignal ausgewählt)
<m>:	Systemvariablen- bzw. Schutzbereichsindex	
	Datentyp:	INT
	Wertebereich:	0, 1, 2, ... (\$MN_MM_3D_MAXNUM_PROT_AREAS - 1)
<Bit-Nummer>:	Bit-Nummer (0, 1, 2, ... 63) der 64-Bit breiten Schnittstelle	
	Datentyp:	INT

Beispiel

Der 6. Schutzbereich ist dem 18. Bit der Schnittstelle (DB10.DBX236.1) zugeordnet:

Programmcode	Kommentar
N100 \$NP_BIT_NO[5] = 17	; 6. Schutzbereich, ; DB10.DBX236.1

Zuordnung: Bit-Nummer zu Nahtstellensignal

Bit →	DB10, (PLC → NC)	DB10, (NC → PLC)	Bit →	DB10, (PLC → NC)	DB10, (NC → PLC)	Bit →	DB10, (PLC → NC)	DB10, (NC → PLC)	Bit →	DB10, (PLC → NC)	DB10, (NC → PLC)
0	DBX234.0	DBX226.0	8	DBX235.0	DBX227.0	16	DBX236.0	DBX228.0	24	DBX237.0	DBX229.0
1	DBX234.1	DBX226.1	9	DBX235.1	DBX227.1	17	DBX236.1	DBX228.1	25	DBX237.1	DBX229.1
2	DBX234.2	DBX226.2	10	DBX235.2	DBX227.2	18	DBX236.2	DBX228.2	26	DBX237.2	DBX229.2
3	DBX234.3	DBX226.3	11	DBX235.3	DBX227.3	19	DBX236.3	DBX228.3	27	DBX237.3	DBX229.3
4	DBX234.4	DBX226.4	12	DBX235.4	DBX227.4	20	DBX236.4	DBX228.4	28	DBX237.4	DBX229.4
5	DBX234.5	DBX226.5	13	DBX235.5	DBX227.5	21	DBX236.5	DBX228.5	29	DBX237.5	DBX229.5
6	DBX234.6	DBX226.6	14	DBX235.6	DBX227.6	22	DBX236.6	DBX228.6	30	DBX237.6	DBX229.6
7	DBX234.7	DBX226.7	15	DBX235.7	DBX227.7	23	DBX236.7	DBX228.7	31	DBX237.7	DBX229.7

Bit →	DB10, (PLC → NC)	DB10, (NC → PLC)	Bit →	DB10, (PLC → NC)	DB10, (NC → PLC)	Bit →	DB10, (PLC → NC)	DB10, (NC → PLC)	Bit →	DB10, (PLC → NC)	DB10, (NC → PLC)
32	DBX238.0	DBX230.0	40	DBX239.0	DBX231.0	48	DBX240.0	DBX232.0	56	DBX241.0	DBX233.0
33	DBX238.1	DBX230.1	41	DBX239.1	DBX231.1	49	DBX240.1	DBX232.1	57	DBX241.1	DBX233.1
34	DBX238.2	DBX230.2	42	DBX239.2	DBX231.2	50	DBX240.2	DBX232.2	58	DBX241.2	DBX233.2
35	DBX238.3	DBX230.3	43	DBX239.3	DBX231.3	51	DBX240.3	DBX232.3	59	DBX241.3	DBX233.3
36	DBX238.4	DBX230.4	44	DBX239.4	DBX231.4	52	DBX240.4	DBX232.4	60	DBX241.4	DBX233.4
37	DBX238.5	DBX230.5	45	DBX239.5	DBX231.5	53	DBX240.5	DBX232.5	61	DBX241.5	DBX233.5
38	DBX238.6	DBX230.6	46	DBX239.6	DBX231.6	54	DBX240.6	DBX232.6	62	DBX241.6	DBX233.6
39	DBX238.7	DBX230.7	47	DBX239.7	DBX231.7	55	DBX240.7	DBX232.7	63	DBX241.7	DBX233.7

Literatur

Eine ausführliche Beschreibung der Nahtstellensignale findet sich im Listenhandbuch NC-Variablen und Nahtstellensignale.

6.2.3.9 \$NP_INIT_STAT

Funktion

In die Systemvariable ist der Initialisierungsstatus des Schutzbereichs einzutragen.

In folgenden Situationen wird der Status eines Schutzbereichs auf den parametrisierten Initialisierungsstatus gesetzt:

- Im Hochlauf der Steuerung
- Beim Aufruf der Funktion PROTA (Seite 305), nachdem der Schutzbereich im laufenden Betrieb durch Schreiben der Schutzbereich-spezifischen Systemvariablen neu angelegt wurde

- Beim Aufruf der Funktion PROTA (Seite 305) mit Parameter "R"
- Beim Aufruf der Funktion PROTS (Seite 306) mit Parameter "R"

Syntax

```
$NP_INIT_STAT[<m>] = "<Status>"
```

Bedeutung

\$NP_INIT_STAT:	Initialisierungsstatus des Schutzbereichs	
	Datentyp:	STRING
	Wertebereich:	"A", "a", "I", "i", "P", "p"
	Wert	Schutzraumstatus
	"A"oder "a"	Aktiviert
	"I"oder "i"	Inaktiv
	"P"oder "p"	Voraktiviert bzw. PLC-gesteuert ¹⁾
Defaultwert:	"I" (inaktiv)	
<m>:	Systemvariablen- bzw. Schutzbereichsindex	
	Datentyp:	INT
	Wertebereich:	0, 1, 2, ... (\$MN_MM_MAXNUM_3D_PROT_AREAS - 1)
<Status>:	Initialisierungsstatus	
	Datentyp:	STRING
1) Die Aktivierung / Deaktivierung erfolgt über: DB10.DBX234.0 - DBX241.7		

Beispiel

Der Initialisierungsstatus des 6. Schutzbereichs wird auf "P" (voraktiviert bzw. PLC-gesteuert) gesetzt:

Programmcode	Kommentar
N100 \$NP_INIT_STAT[5] = "P"	; 6. Schutzbereich, ; Initialisierungsstatus = "P"

Der aktuelle Status ist abhängig vom Zustand des in \$NP_BIT_NO (Seite 253) parametrisierten Nahtstellensignals.

6.2.3.10 \$NP_INDEX

Funktion

Für automatische Schutzbereiche (\$NP_PROT_TYPE (Seite 248)) ist in die Systemvariable die Adresse einzutragen, unter der die Geometriedaten des zu schützenden Maschinenteils, Werkzeugs, etc., abgelegt sind. Aus den Geometriedaten werden von der Steuerung automatisch die geometrischen Abmessungen des Schutzbereichs erzeugt.

Beispiel

Bei einem automatischen Werkzeugschutzbereich (\$NP_PROT_TYPE == "TOOL") werden die geometrischen Abmessungen des Schutzbereichs anhand der Werkzeugdaten erzeugt.

Syntax

\$NP_INDEX[<m>,<i>] = <Wert>

Bedeutung

\$NP_INDEX:	Adresse der Geometriedaten für den automatischen Schutzbereiche	
	Datentyp:	INT[3]
<m>:	Systemvariablen- bzw. Schutzbereichsindex	
	Datentyp:	INT
	Wertebereich:	0, 1, 2, ... (\$MN_MM_MAXNUM_3D_PROT_AREAS - 1)
<i>:	Index Die Bedeutung der Systemvariablen \$NP_INDEX[<m>,<i>], mit i = 0, 1, 2, ... ist abhängig vom Typ (\$NP_PROT_TYPE) des automatischen Schutzbereichs. Siehe typspezifische Tabellen.	
	Datentyp:	INT
	Wertebereich:	0, 1, 2
<Wert>:	Adresse	
	Datentyp:	INT

Typ: Automatischer Werkzeugschutzbereich (\$NP_PROT_TYPE == "TOOL")

<i>	<Wert>	
	Bei aktiver Werkzeugverwaltung	Ohne Werkzeugverwaltung
0	Revolvermagazin: Werkzeugplatznummer	Spindelnummer
	Kein Revolvermagazin: Spindelnummer	
1	Magazinnummer	---
2	TOA-Bereich ¹⁾	
1) Der TOA-Bereich "1" kann sowohl mit 0 als auch mit 1 adressiert werden.		

Beispiel

Der 6. Schutzbereich ist ein automatischer Werkzeugschutzbereich (\$NP_PROT_TYPE == "TOOL"). Die geometrischen Abmessungen des Schutzbereichs sollen aus den Geometriedaten des Werkzeugs auf folgendem Werkzeugplatz erzeugt werden:

- Werkzeugplatznummer: 1
- Magazinnummer: 9998 (Spindel 1)
- TOA-Bereich: 1

Die Werkzeugverwaltung ist aktiv.

Programmcode	Kommentar
; Die Abmessungen des 6. Schutzbereichs basieren auf den Werkzeugdaten	
; des Werkzeugs, das sich an folgender Stelle befindet:	
N100 \$NP_INDEX[5,0] = 1	; Werkzeugplatznummer = 1
N110 \$NP_INDEX[5,1] = 9998	; Magazinnummer = 9998 (Spindel 1)
N120 \$NP_INDEX[5,2] = 1	; TOA-Bereich = 1

Siehe auch

\$NP_PROT_TYPE (Seite 248)

6.2.4 Systemvariablen: Schutzbereichselemente für Maschinenschutzbereiche

6.2.4.1 Übersicht

Mit folgenden Systemvariablen wird ein Schutzbereichselement eines Maschinenschutzbereichs parametrisiert:

Name	Bedeutung
\$NP_NAME	Name des Schutzbereichselements
\$NP_NEXT	Name des nachfolgenden Schutzbereichselements
\$NP_NEXTP	Name des nachfolgenden, zu \$NP_NEXT parallelen Schutzbereichselements
\$NP_COLOR	Farbe und Transparenz des Schutzbereichselements.
\$NP_D_LEVEL	Detaillierungsgrad des Schutzbereichselements
\$NP_USAGE	Verwendungsart des Schutzbereichselements
\$NP_TYPE	Typ des Schutzbereichselements
\$NP_FILENAME	Dateiname der STL-Datei, welche die Geometriedaten des Schutzbereichselements enthält (nur relevant bei \$NP_TYPE == "FILE")
\$NP_PARA	Geometrische Parameter des Schutzbereichskörpers (nur relevant bei \$NP_TYPE == "BOX" oder "SPHERE" oder "CYLINDER")
\$NP_OFF	Verschiebungsvektor des lokalen Koordinatensystems des Schutzbereichselements
\$NP_DIR	Richtungsvektor für die Drehung des lokalen Koordinatensystems des Schutzbereichselements
\$NP_ANG	Winkel für die Drehung des lokalen Koordinatensystems des Schutzbereichselements

Die Systemvariablen sind in den nachfolgenden Kapiteln ausführlich beschrieben.

Hinweis

Definierten Ausgangszustand herstellen

Es wird empfohlen, vor Parametrierung der Schutzbereichselemente einen definierten Ausgangszustand zu erzeugen. Dazu sind die Systemvariablen der Schutzbereichselemente mit der Funktion DELOBJ() (Seite 225) auf ihren Default-Wert zu setzen.

Ändern von Systemvariablenwerten

Wird der Wert einer der oben aufgeführten Systemvariablen geändert, wird die Änderung auf der Bedienoberfläche, z.B. SINUMERIK Operate, sofort sichtbar. Das Maschinenmodell der NC wird aber erst nach einer expliziten Anforderung zum Neuberechnen des Maschinenmodells durch Aufruf der Funktion PROTA() (Seite 305) bzw. PROTS() (Seite 306) aktualisiert.

6.2.4.2 \$NP_NAME

Funktion

In die Systemvariable ist der NC-weit eindeutige Name des Schutzbereichselements einzutragen. Über diesen Namen wird das Schutzbereichselement referenziert. Der Name wird auch im grafischen Editor von SINUMERIK Operate angezeigt.

Syntax

`$NP_NAME [<n>] = "<Name>"`

Bedeutung

\$NP_NAME:	Name des Schutzbereichselements	
	Datentyp:	STRING
	Defaultwert:	"" (Leerstring)
<n>:	Systemvariablen- bzw. Schutzbereichselementindex	
	Datentyp:	INT
	Wertebereich:	0, 1, 2, ... (\$MN_MM_MAXNUM_3D_PROT_AREA_ELEM - 1)
<Name>:	Name des Schutzbereichselements	
	Datentyp:	STRING

Beispiel

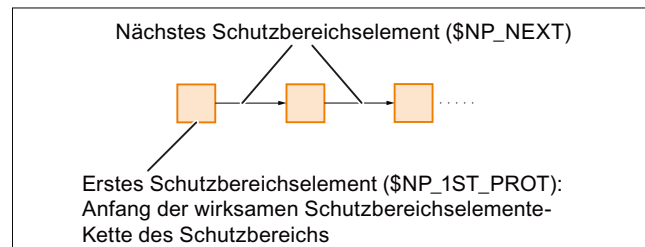
Dem 19. Schutzbereichselement wird der Name "Spindelkasten" zugewiesen:

Programmcode	Kommentar
N100 \$NP_NAME[18] = "Spindelkasten"	; 19. Schutzbereichselements, ; Name = "Spindelkasten"

6.2.4.3 \$NP_NEXT

Funktion

Ist ein Schutzbereich aus mehreren Schutzbereichselementen aufgebaut, müssen diese miteinander verkettet werden. Dazu ist in jedem Schutzbereichselement in die Systemvariable \$NP_NEXT der Name des nachfolgenden Schutzbereichselements einzutragen.



Ist kein nachfolgendes Schutzbereichselement vorhanden, ist als Name ein Leerstring "" einzutragen.

Verschiebung und Rotation

Eine Verschiebung und/oder Rotation im aktuellen Schutzbereichselement (\$NP_OFF (Seite 275), \$NP_DIR (Seite 276) und \$NP_ANG (Seite 278)) wirkt auf das nachfolgende in \$NP_NEXT angegebene Schutzbereichselement. D.h. die Festlegung der räumlichen Lage und Orientierung des nachfolgenden Schutzbereichselements erfolgt relativ zum aktuellen Schutzbereichselement.

Syntax

```
$NP_NEXT[<n>] = "<Name>"
```

Bedeutung

\$NP_NEXT:	Name des nachfolgenden Schutzbereichselements	
	Datentyp:	STRING
	Wertebereich:	Alle in \$NP_NAME (Seite 258) enthaltenen Namen
	Defaultwert:	"" (Leerstring)

<n>:	Systemvariablen- bzw. Schutzbereichselementindex	
	Datentyp:	INT
	Wertebereich:	0, 1, 2, ... (\$MN_MM_MAXNUM_3D_PROT_AREA_ELEM - 1)
<Name>:	Schutzbereichsname	
	Datentyp:	STRING

Beispiel

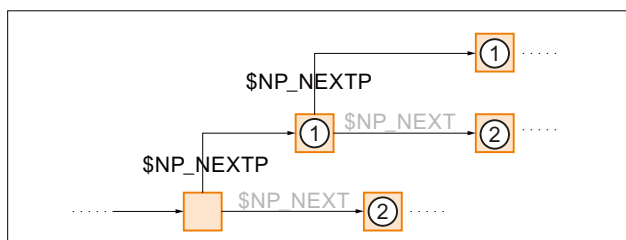
Am 19. Schutzbereichselement ist das nachfolgende Schutzbereichselement mit dem Namen "Kühlmitteldüse 1" angebracht:

Programmcode	Kommentar
N100 \$NP_NAME[18] = "Kühlmitteldüse 1"	; 19 Schutzbereichselement, ; Name des Nachfolgers: "Kühlmitteldüse 1"

6.2.4.4 \$NP_NEXTP

Funktion

Die Schutzbereichselemente-Kette kann über die Systemvariable \$NP_NEXTP verzweigt werden. Dazu sind in einem Schutzbereichselement in den Systemvariablen \$NP_NEXT und \$NP_NEXTP nachfolgende Schutzbereichselemente anzugeben. Diese Schutzbereichselemente liegen dann parallel zueinander in zwei voneinander unabhängigen Teilketten.



- ① Nachfolgendes paralleles Schutzbereichselement
- ② Nachfolgendes Schutzbereichselement der gleichen Teilkette

Bild 6-2 Schutzbereichselemente in parallelen Teilketten

Anwendungsbeispiel

Über unabhängige Teilketten können z.B. verschiedene Maschinenteile eines Schutzbereichs für Visualisierung bzw. Kollisionsvermeidung unterschiedlich modelliert werden. Typischerweise wird dazu für das Schutzbereichselement, auf das mit \$NP_NEXT verwiesen wird, als Verwendung in \$NP_USAGE (Seite 264) "C" (Kollisionsüberwachung) und im Schutzbereichselement, auf das in \$NP_NEXTP verwiesen wird, der Wert "V" (Visualisierung) angegeben.

Verschiebung und Rotation

Eine Verschiebung und/oder Rotation im aktuellen Schutzbereichselement (\$NP_OFF (Seite 275), \$NP_DIR (Seite 276) und \$NP_ANG (Seite 278)) wirkt auf das nachfolgende in \$NP_NEXTP angegebene Schutzbereichselement. D.h. die Festlegung der räumlichen Lage und Orientierung des nachfolgenden Schutzbereichselements erfolgt relativ zum aktuellen Schutzbereichselement.

Syntax

```
$NP_NEXTP[<n>] = "<Name>"
```

Bedeutung

\$NP_NEXTP:	Name des verzweigenden Schutzbereichselements	
	Datentyp:	STRING
	Wertebereich:	Alle in \$NP_NAME (Seite 258) enthaltenen Namen
	Defaultwert:	"" (Leerstring)
<n>:	Systemvariablen- bzw. Schutzbereichselementindex	
	Datentyp:	INT
	Wertebereich:	0, 1, 2, ... (\$MN_MM_MAXNUM_3D_PROT_AREA_ELEM - 1)
<Name>:	Schutzbereichsname	
	Datentyp:	STRING

Beispiel

Am 19. Schutzbereichselement ist das nachfolgende parallele Schutzbereichselement mit dem Namen "Kühlmitteldüse 2" angebracht:

Programmcode	Kommentar
N100 \$NP_NEXTP[18] = "Kühlmitteldüse 2"	; 19. Schutzbereichselement, ; Name des parallelen Nachfolgers: "Kühlmitteldüse 2"

6.2.4.5 \$NP_COLOR

Funktion

In die Systemvariable ist der Schutzbereichselement-spezifische Wert für Alpha/Transparenz- und Farbe (ARGB) einzutragen. Dieser Wert wird für die Darstellung des Schutzbereichselements auf der Bedienoberfläche verwendet. Wird für ein Schutzbereichselement kein eigener Wert parametrisiert, wirkt der Schutzbereichs-spezifische Wert aus \$NP_PROT_COLOR (Seite 250).

6.2 Inbetriebnahme

Aufbau

Alpha/Transparenz- und Farbwert werden als Doppelwort im Hex-Format angegeben:

AARRGGBB_H

- 1. - 3. Byte: RGB-Farbwert. Siehe Kapitel "Farbtafel (Seite 243)".
- 4. Byte: Alpha-Kanal- bzw. Transparenzwert

	Byte	Bedeutung	Wertebereich
BB	1	Blau	0 - 255 _D bzw. 0 - FF _H
GG	2	Grün	
RR	3	Rot	
AA	4	Alpha-Kanal bzw. Transparenz ¹⁾	

1) 0 = transparent bzw. nicht sichtbar, 255_D = FF_H = nicht transparent bzw. massiv

Syntax

\$NP_COLOR[<n>] = <Name>

Bedeutung

\$NP_COLOR:	Alpha/Transparenz- und Farbwert des Schutzbereichselements	
	Datentyp:	DWORD
	Wertebereich:	00000000 _H - FFFFFFFF _H
	Defaultwert:	00000000 _H (Schwarz, nicht sichtbar)
<m>:	Systemvariablen- bzw. Schutzbereichselementindex	
	Datentyp:	INT
	Wertebereich:	0, 1, 2, ... (\$MN_MM_MAXNUM_3D_PROT_AREA_ELEM - 1)
<Wert>:	Transparenz- und Farbwert	
	Datentyp:	DWORD

Beispiel

Der 19. Schutzbereich soll an der Bedienoberfläche halbttransparent und in einem grün-blauen Farbton dargestellt werden:

- AA = 7F_H = 127_D ≙ 50% Transparenz
- RR (Rot) = 00 ≙ kein roter Farbanteil
- GG (Grün) = FF_H = 255_D ≙ 100% grüner Farbanteil
- BB (Blau) = 33_H = 51_D ≙ 20% blauer Farbanteil

Programmcode	Kommentar
N100 \$NP_COLOR[18] = 'H7F00FF33'	; 19. Schutzbereich, ; Alpha/Transparenz- und Farbwert = 'H7F00FF33'

6.2.4.6 \$NP_D_LEVEL

Funktion

Über die Systemvariable wird festgelegt, ab welchem Detaillierungsgrad das Schutzbereichselement auf der Bedienoberfläche angezeigt wird. Wird für ein Schutzbereichselement kein vom Defaultwert verschiedener Wert parametrisiert, wirkt der Schutzbereichs-spezifische Wert aus \$NP_PROT_D_LEVEL (Seite 252).

Detaillierungsgrad

- Niedrigster Detaillierungsgrad: 0
- Höchster Detaillierungsgrad: 3

Ist zur Visualisierung des Maschinenmodells der Detaillierungsgrad x aktiv, werden alle Schutzbereiche und Schutzbereichselemente angezeigt, für deren Detaillierungsgrad D gilt: $D \leq x$

Syntax

`$NP_D_LEVEL[<n>] = <Wert>`

Bedeutung

\$NP_PROT_D_LEVEL:	Detaillierungsgrad des Schutzbereichselements	
	Datentyp:	INT
	Wertebereich:	$0 \leq D \leq 3$
	Defaultwert:	0
<m>:	Systemvariablen- bzw. Schutzbereichselementindex	
	Datentyp:	INT
	Wertebereich:	0, 1, 2, ... (\$MN_MM_MAXNUM_3D_PROT_AREA_ELEM - 1)
<Wert>:	Detaillierungsgrad	
	Datentyp:	INT

Beispiel

Der 19. Schutzbereich soll immer dargestellt werden \Rightarrow Detaillierungsgrad 0:

Programmcode	Kommentar
N100 \$NP_PROT_D_LEVEL[18] = 0	; 19. Schutzbereich, ; Detailierungsgrad = 0

6.2.4.7 \$NP_USAGE

Funktion

In die Systemvariable ist die Verwendungsart des Schutzbereichselements einzutragen. Die Verwendungsart legt fest, wie das Schutzbereichselement von der Kollisionsvermeidung zu berücksichtigen ist:

- Nur Visualisierung, keine Kollisionsberechnung
- Nur Kollisionsberechnung, keine Visualisierung
- Visualisierung und Kollisionsberechnung

Verwendungsart	Bedeutung
Visualisierung	Das Schutzbereichselement wird im Maschinenmodell auf der Bedienoberfläche von SINUMERIK Operate angezeigt
Kollisionsberechnung	Das Schutzbereichselement wird bei der Kollisionsberechnung mit berücksichtigt

Syntax

\$NP_USAGE [<n>] = "<Wert>"

Bedeutung

\$NP_USAGE:	Verwendungsart des Schutzbereichselements	
	Datentyp:	CHAR
	Wertebereich:	"V", "v", "C", "c", "A", "a"
	Wert	Bedeutung
	"V" oder "v"	Nur Visualisierung, keine Kollisionsberechnung
	"C" oder "c"	Nur Kollisionsberechnung, keine Visualisierung
	"A" oder "a"	Visualisierung und Kollisionsberechnung
	Defaultwert:	"A"
<n>:	Systemvariablen- bzw. Schutzbereichselementindex	
	Datentyp:	INT
	Wertebereich:	0, 1, 2, ... (\$MN_MM_MAXNUM_3D_PROT_AREA_ELEM - 1)
<Wert>:	Verwendungsart	
	Datentyp:	CHAR

Beispiel

Das 19. Schutzbereichselement soll an der Bedienoberfläche angezeigt und bei der Kollisionsberechnung berücksichtigt werden:

Programmcode	Kommentar
N100 \$NP_USAGE[18] = "A"	; 19. Schutzbereich, ; Verwendungsart = "A"

6.2.4.8 \$NP_TYPE

Funktion

In die Systemvariable ist der Typ des Schutzbereichselements einzutragen.

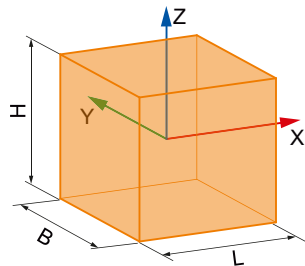
Typ: "FRAME"

Ein Schutzbereichselement vom Typ "FRAME" enthält keinen Körper, sondern definiert eine Koordinatentransformation des lokalen Koordinatensystems. Die Koordinatentransformation wirkt für alle nachfolgenden (\$NP_NEXT (Seite 259)) und/oder parallelen (\$NP_NEXTP (Seite 260)) Schutzbereichselemente. Die Werte der Koordinatentransformation werden eingestellt über:

- Verschiebung: \$NP_OFF (Seite 275)
- Richtungsvektor der Drehung: \$NP_DIR (Seite 276)
- Drehwinkel: \$NP_ANG (Seite 278)

Für den Typ "FRAME" sind in \$NP_PARA (Seite 274) keine Parameter anzugeben.

Typ: "BOX"



- L Länge in X-Richtung
- B Breite in Y-Richtung
- H Höhe in Z-Richtung

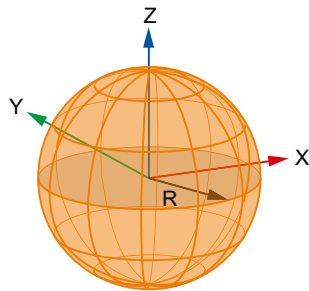
Ein Schutzbereichselement vom Typ "BOX" definiert im lokalen Koordinatensystem des Schutzbereichselements einen achsenparallelen Quader. Der Mittelpunkt des Quaders liegt im

Ursprung des lokalen Koordinatensystems. Gleichzeitig mit der Definition des Körpers kann über folgende Systemvariablen das lokale Koordinatensystem transformiert werden:

- Verschiebung: \$NP_OFF (Seite 275)
- Richtungsvektor der Drehung: \$NP_DIR (Seite 276)
- Drehwinkel: \$NP_ANG (Seite 278)

Die Parameter Länge, Breite und Höhe sind einzutragen in \$NP_PARA (Seite 274)

Typ: "SPHERE"



R Radius der Kugel

Ein Schutzbereichselement vom Typ "SPHERE" definiert im lokalen Koordinatensystem des Schutzbereichselements eine Kugel. Der Mittelpunkt der Kugel liegt im Ursprung des lokalen Koordinatensystems. Gleichzeitig mit der Definition des Körpers kann über folgende Systemvariablen das lokale Koordinatensystem transformiert werden:

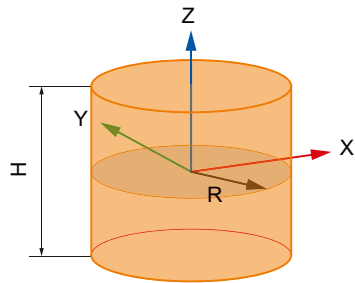
- Verschiebung: \$NP_OFF (Seite 275)
- Richtungsvektor der Drehung: \$NP_DIR (Seite 276)
- Drehwinkel: \$NP_ANG (Seite 278)

Der Parameter Radius ist einzutragen in \$NP_PARA (Seite 274)

Hinweis

Drehung

Da der Kugelmittelpunkt und der Startpunkt des Richtungsvektors im Koordinatenursprung des lokalen Koordinatensystems des Schutzbereichselements liegen, hat eine Drehung mittels Richtungsvektor \$NP_DIR (Seite 276) und Drehwinkel \$NP_ANG (Seite 278) keine Auswirkung auf die Lage der Kugel.

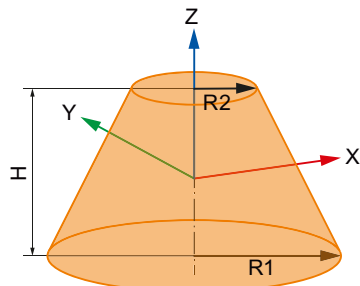
Typ: "CYLINDER"

- H Höhe in Z-Richtung
R Radius in der X/Y-Ebene

Ein Schutzbereichselement vom Typ "CYLINDER" definiert im lokalen Koordinatensystem des Schutzbereichselements einen Zylinder. Der Mittelpunkt des Zylinders liegt im Ursprung des lokalen Koordinatensystems. Gleichzeitig mit der Definition des Körpers kann über folgende Systemvariablen das lokale Koordinatensystem transformiert werden:

- Verschiebung: \$NP_OFF (Seite 275)
- Richtungsvektor der Drehung: \$NP_DIR (Seite 276)
- Drehwinkel: \$NP_ANG (Seite 278)

Die Parameter Höhe und Radius sind einzutragen in \$NP_PARA (Seite 274)

Typ: "CONE"

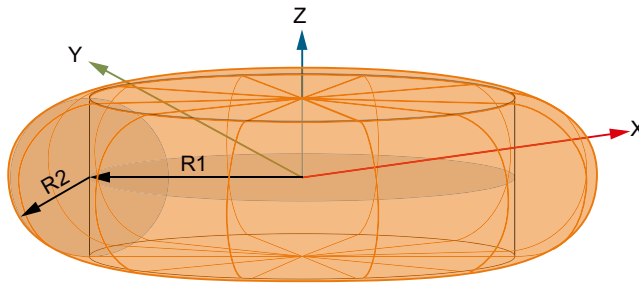
- H Höhe in Z-Richtung
R1 Radius 1 in der X/Y-Ebene
R2 Radius 2 in der X/Y-Ebene

Ein Schutzbereichselement vom Typ "CONE" definiert im lokalen Koordinatensystem des Schutzbereichselements einen Kegel. Der Mittelpunkt des Kegels (halbe Kegelhöhe auf der Symmetrieachse des Kegels) liegt im Ursprung des lokalen Koordinatensystems. Gleichzeitig mit der Definition des Körpers kann über folgende Systemvariablen das lokale Koordinatensystem transformiert werden:

- Verschiebung: \$NP_OFF (Seite 275)
- Richtungsvektor der Drehung: \$NP_DIR (Seite 276)
- Drehwinkel: \$NP_ANG (Seite 278)

Die Parameter Höhe, Radius 1 und Radius 2 sind einzutragen in \$NP_PARA (Seite 274)

Typ: "TORUS"



- R1 Major Radius (= Abstand des Kreismittelpunkts vom Torusmittelpunkt in der X/Y-Ebene)
- R2 Minor Radius (= Kreisradius)

Ein Schutzbereichselement vom Typ "TORUS" definiert im lokalen Koordinatensystem des Schutzbereichselements einen gefüllten Torus. Die Form eines Torus lässt sich am einfachsten mithilfe eines Kreises beschreiben, der um eine in der Kreisebene liegenden Achse rotiert wird. Im Unterschied zum normalen Torus ist bei einem gefüllten Torus das Loch in der Mitte ausgefüllt. Der Mittelpunkt des gefüllten Torus liegt im Ursprung des lokalen Koordinatensystems. Gleichzeitig mit der Definition des Körpers kann über folgende Systemvariablen das lokale Koordinatensystem transformiert werden:

- Verschiebung: \$NP_OFF (Seite 275)
- Richtungsvektor der Drehung: \$NP_DIR (Seite 276)
- Drehwinkel: \$NP_ANG (Seite 278)

Die Parameter Radius 1 und Radius 2 sind einzutragen in \$NP_PARA (Seite 274)

Typ: "FILE"

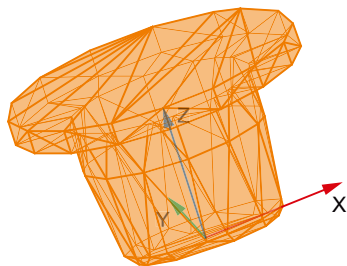


Bild 6-3 Beispielkörper im STL-Format

Ein Schutzbereichselement vom Typ "FILE" definiert im lokalen Koordinatensystem des Schutzbereichselements einen Körper, dessen Geometriedaten im STL-Format (Dreiecksflächen) in der angegebenen Datei enthalten sind. Der Nullpunkt des Körpers liegt im Ursprung des lokalen Koordinatensystems. Gleichzeitig mit der Definition des Körpers kann über folgende Systemvariablen das lokale Koordinatensystem transformiert werden:

- Verschiebung: \$NP_OFF (Seite 275)
- Richtungsvektor der Drehung: \$NP_DIR (Seite 276)
- Drehwinkel: \$NP_ANG (Seite 278)

Der Parameter ist einzutragen in \$NP_FILENAME (Seite 269):

Syntax

```
$NP_TYPE [<n>] = "<Type>"
```

Bedeutung

\$NP_TYPE:	Typ des Schutzbereichselements	
	Datentyp:	STRING
	Wertebereich:	"FRAME", "BOX", "SPHERE", "CYLINDER", "CONE", "TORUS", "FILE"
	Defaultwert:	"" (Leerstring)
<n>:	Systemvariablen- bzw. Schutzbereichselementindex	
	Datentyp:	INT
	Wertebereich:	0, 1, 2, ... (\$MN_MM_MAXNUM_3D_PROT_AREA_ELEM - 1)
<Type>:	Typbezeichnung	
	Datentyp:	STRING

Beispiel

Das 19. Schutzbereichselement ist ein Quader:

Programmcode	Kommentar
N100 \$NP_TYPE[18] = "BOX"	; 19. Schutzbereichselement, ; Typ = "Quader"

6.2.4.9 \$NP_FILENAME

Funktion

Für Schutzbereichselemente vom Typ "FILE" (\$NP_TYPE (Seite 265)) ist in die Systemvariable der Dateiname der Datei mit der Beschreibung der Geometriedaten des Schutzbereichselements einzutragen.

Folgende Dateitypen sind aktuell möglich:

- STL-Dateien
- NPP-Dateien

STL-Dateien

Eine STL-Datei (Dateiendung .STL) muss die Beschreibung der Geometriedaten eines 3-D Körpers mittels Dreiecken im STL-Format (**S**tandard **T**essellation **L**anguage) enthalten.

Suchpfad

Die in der Systemvariablen eingetragene Datei wird in folgender Reihenfolge in den auf der CF-Card vordefinierten Verzeichnissen gesucht:

1. /oem/sinumerik/nck/prot_data/machine/3d_data/mm
2. /oem/sinumerik/nck/prot_data/machine/3d_data/inch

Interpretation der Längenangaben

Die in der STL-Datei enthaltenen Längenangaben werden, abhängig vom Ablageverzeichnis, in mm oder inch interpretiert:

- <Pfad>/mm: Interpretation der Längenangaben in Millimeter
- <Pfad>/inch: Interpretation der Längenangaben in Zoll

Hinweis

Maximale Länge des Dateinamens

Die Länge des Dateinamens darf einschließlich Punkt und Dateiendung 49 Zeichen betragen. Bei mehr als 49 Zeichen wird bei der Erstellung eines Archivs ein Alarm angezeigt.

NPP-Dateien

Eine NPP-Datei (Dateiendung .NPP) muss die Beschreibung der Geometriedaten von einem oder mehreren Schutzbereichselementen mittels NPP-Systemvariablen (**NC Protection Area Primitives**) enthalten. Über eine NPP-Datei dürfen als Schutzbereichselemente alle in der NC verfügbaren geometrischen Primitive (siehe \$NP_TYPE (Seite 265)) definiert werden.

Suchpfad

Die in der Systemvariablen eingetragene Datei wird in folgender Reihenfolge in den auf der CF-Card vordefinierten Verzeichnissen gesucht:

1. /oem/sinumerik/nck/prot_data/machine/3d_data/mm
2. /oem/sinumerik/nck/prot_data/machine/3d_data/inch

Interpretation der Längenangaben

Die in der NPP-Datei enthaltenen Längenangaben werden, abhängig vom Ablageverzeichnis, in mm oder inch interpretiert:

- <Pfad>/mm: Interpretation der Längenangaben in Millimeter
- <Pfad>/inch: Interpretation der Längenangaben in Zoll

Eigenschaften der NPP-Dateien

- Eine NPP-Datei muss mit den folgenden Kommentarzeilen beginnen:
;COLLISION AVOIDANCE DATA
;LOC_NP_ROOT_NAME="<Root_Name>"
- Als <Root_Name> muss der Name des ersten in der NPP-Datei enthaltenen Schutzbereichselements, dort angegeben unter \$NP_NAME, eingetragen werden.

- Kommentarzeilen sind Zeilen, die mit dem Zeichen ; beginnen
- NPP-Dateien dürfen Leerzeilen enthalten

Eigenschaften der NPP-Systemvariablen

Die in der NPP-Datei enthaltenen NPP-Systemvariablen haben folgende Eigenschaften:

- Gleiche Namen, Bedeutung und Syntax wie die entsprechenden in den NC-Programmen verwendeten Systemvariablen
- Durch die NPP-Systemvariablen werden die Systemvariablen der NC **nicht** überschrieben.
- Die Indizes der NPP-Systemvariablen müssen **innerhalb** einer NPP-Datei **eindeutig** sein.
- Die Indizes der NPP-Systemvariablen können in **verschiedenen** NPP-Dateien **gleich** sein.
- Die Indizes und die den NPP-Systemvariablen zugewiesenen Werte müssen Konstante sein.

Randbedingungen

- Die Werte für \$NP_COLOR (Seite 261), \$NP_D_LEVEL (Seite 263), \$NP_USAGE (Seite 264) werden für die in der NPP-Datei definierten Schutzbereichselemente von dem Schutzbereichselement geerbt, von dem sie eingebunden werden. Alle Schutzbereichselemente einer NPP-Datei haben somit die gleichen Werte für diese Eigenschaften.
- Für die Positionierung der Schutzbereichselemente einer NPP-Datei gelten die gleichen Bedingungen wie für die Positionierung der Schutzbereichselemente mit den Systemvariablen der NC (\$NP_TYPE (Seite 265): "BOX", "SPHERE" und "CYLINDER").
- In einer NPP-Datei darf keine weitere STL- oder NPP-Datei eingebunden werden.

Hinweis

Maximale Länge des Dateinamens

Die Länge des Dateinamens darf einschließlich Punkt und Dateierweiterung 49 Zeichen betragen. Bei mehr als 49 Zeichen wird bei der Erstellung eines Archivs ein Alarm angezeigt.

Syntax

```
$NP_FILENAME [<n>] = "<Name>"
```

Bedeutung

\$NP_FILENAME:	Name der STL- oder NPP-Datei	
	Datentyp:	STRING
	Defaultwert:	"" (Leerstring)
<n>:	Systemvariablen- bzw. Schutzbereichselementindex	
	Datentyp:	INT
	Wertebereich:	0, 1, 2, ... (\$MN_MM_MAXNUM_3D_PROT_AREA_ELEM - 1)

<Name>:	Name der STL- oder NPP-Datei	
	Datentyp:	STRING

Beispiele

Verwendung einer STL-Datei

Die Geometriedaten für das 19. Schutzbereichselementes sind in der Datei KUEHLDUESE_1.STL hinterlegt:

Programmcode	Kommentar
N100 \$NP_FILENAME[18] = "KUEHLDUESE_1.STL"	; 19. Schutzbereichselement, ; Filename = "KUEHLDUESE_1.STL"

Verwendung einer NPP-Datei

In den Systemvariablen der NC für das 19. Schutzbereichselement wird die NPP-Datei "Kopf_A.NPP" geladen. Diese enthält die folgenden drei Schutzbereichselemente "Quader-1", "Kugel-1" und "Zylinder-1".

Programmcode	Kommentar
\$NP_NAME[18] = "Kopf"	; 19. Schutzbereichselement
\$NP_NEXT[18] = ""	
\$NP_NEXTP[18] = ""	
\$NP_TYPE[18] = "FILE"	
\$NP_OFF[18,0] = 80.0	
\$NP_OFF[18,1] = 100.0	
\$NP_OFF[18,2] = -50.0	
\$NP_DIR[18,0] = 0.0	
\$NP_DIR[18,1] = 0.0	
\$NP_DIR[18,2] = 0.0	
\$NP_ANG[18] = 0.0	
\$NP_COLOR[18] = 0	; 1)
\$NP_D_LEVEL[18] = 0	; 1)
\$NP_USAGE[18] = "A"	; 1)
\$NP_FILENAME[18] = "Kopf_A.npp"	
; 1) für alle geladenen Schutzbereichselemente aus Datei Kopf_A.NPP wirksam	

Inhalt der Datei "Kopf_A.NPP"

Programmcode	Kommentar
;COLLISION AVOIDANCE DATA	; 1. Kopfzeile
;LOC_NP_ROOT_NAME = "Box-1"	; 2. Kopfzeile
\$NP_NAME[0] = "Quader-1"	; 1. Schutzbereichselement
\$NP_NEXT[0] = "Kugel-1"	
\$NP_NEXTP[0] = "Zylinder-1"	
\$NP_TYPE[0] = "BOX"	

Programmcode	Kommentar
\$NP_PARA[0,0] = 340	
\$NP_PARA[0,1] = 340	
\$NP_PARA[0,2] = 340	
\$NP_OFF[0,0] = 42	
\$NP_OFF[0,1] = 73	
\$NP_OFF[0,2] = -100	
\$NP_DIR[0,0] = 0	
\$NP_DIR[0,1] = 0	
\$NP_DIR[0,2] = 1	
\$NP_ANG[0] = 30	
\$NP_NAME[1] = "Kugel-1"	; 2. Schutzbereichselement
\$NP_NEXT[1] = ""	
\$NP_NEXTP[1] = ""	
\$NP_TYPE[1] = "SPHERE"	
\$NP_PARA[1,0] = 20	
\$NP_PARA[1,1] = 0	
\$NP_PARA[1,2] = 0	
\$NP_OFF[1,0] = 170	
\$NP_OFF[1,1] = 170	
\$NP_OFF[1,2] = 170	
\$NP_DIR[1,0] = 0	
\$NP_DIR[1,1] = 0	
\$NP_DIR[1,2] = 0	
\$NP_ANG[1] = 0	
\$NP_NAME[2] = "Zylinder-1"	; 3. Schutzbereichselement
\$NP_NEXT[2] = ""	
\$NP_NEXTP[2] = ""	
\$NP_TYPE[2] = "CYLINDER"	
\$NP_PARA[2,0] = 20	
\$NP_PARA[2,1] = 20	
\$NP_PARA[2,2] = 0	
\$NP_OFF[2,0] = 170	
\$NP_OFF[2,1] = 170	
\$NP_OFF[2,2] = 170	
\$NP_DIR[2,0] = 1	
\$NP_DIR[2,1] = 2	
\$NP_DIR[2,2] = 3	
\$NP_ANG[2] = 73	

6.2.4.10 \$NP_PARA

Funktion

In die Systemvariable sind die Abmessungen des Schutzbereichskörpers entsprechend des Typs des Schutzbereichselements (\$NP_TYPE (Seite 265)) einzutragen.

Koordinatensystem

Das lokale Koordinatensystem, in dem die Lage des Schutzbereichskörpers angegeben wird, wird durch die Systemvariablen \$NP_OFF (Seite 275), \$NP_DIR (Seite 276), \$NP_ANG (Seite 278) festgelegt.

Syntax

\$NP_PARA[<n>,<i>] = <Wert>

Bedeutung

\$NP_PARA:	Parameterwerte entsprechend des Typs des Schutzbereichselements					
	Datentyp:	REAL				
	Defaultwert:	0.0				
<n>:	Systemvariablen- bzw. Schutzbereichselementindex					
	Datentyp:	INT				
	Wertebereich:	0, 1, 2, ... (\$MN_MM_MAXNUM_3D_PROT_AREA_ELEM - 1)				
<i>:	Parameterindex					
	Datentyp:	INT				
	Wertebereich:	0, 1, 2				
	Parameterindex	Typ des Schutzbereichselements				
		BOX	SPHERE	CYLINDER	CONE	TORUS
	0	Länge in X	Radius	Höhe in Z	Höhe in Z	Radius 1 ¹⁾
	1	Breite in Y	---	Radius ¹⁾	Radius 1 ¹⁾	Radius 2
	2	Höhe in Z	---	---	Radius 2 ¹⁾²⁾	---
<Wert>:	Parameterwert					
	Datentyp:	REAL				
	Wertebereich:	0.0 < x ≤ max. REAL-Wert				

1) Radius in der X/Y-Ebene

2) Parameterwert 0.0 ist zulässig.

---: nicht ausgewerteter Parameter

Beispiel

Das 19. Schutzbereichselement ist ein Quader mit den Abmessungen:

- Länge: 50.0 in X-Richtung
- Breite: 100.0 in Y-Richtung
- Höhe: 75.5 in Z-Richtung

Programmcode	Kommentar
; 19. Schutzbereichselement	
N100 \$NP_TYPE[18] = "BOX"	; Typ = "BOX"
N120 \$NP_PARA[18,0] = 50.0	; Länge in X = 50.0
N130 \$NP_PARA[18,1] = 100.0	; Breite in Y = 100.0
N140 \$NP_PARA[18,2] = 75.5	; Höhe in Z = 75.5

6.2.4.11 \$NP_OFF

Funktion

In die Systemvariable ist der Verschiebungsvektor einzutragen, um den das lokale Koordinatensystem des Schutzbereichselements zum Koordinatensystem des vorhergehenden Schutzbereichselements verschoben ist.

Syntax

\$NP_OFF[<n>,<i>] = <Wert>

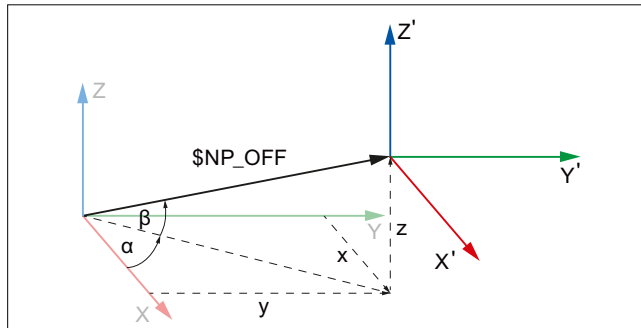
Bedeutung

\$NP_OFF:	Verschiebungsvektor	
	Datentyp:	REAL
	Wertebereich:	- max. REAL-Wert ≤ x ≤ + max. REAL-Wert
	Defaultwert:	(0.0, 0.0, 0.0)
<m>:	Systemvariablen- bzw. Schutzbereichselementindex	
	Datentyp:	INT
	Wertebereich:	0, 1, 2, ... (\$MN_MM_MAXNUM_3D_PROT_AREA_ELEM - 1)
<i>:	Koordinatenindex	
	Datentyp:	INT
	Wertebereich:	0: X-Koordinate (Abszisse) 1: Y-Koordinate (Ordinate) 2: Z-Koordinate (Applikate)
<Wert>:	Koordinatenwert	
	Datentyp:	REAL
	Wertebereich:	- max. REAL-Wert ≤ x ≤ + max. REAL-Wert

Beispiel

Das lokale Koordinatensystem des 19. Schutzbereichselements ist gegenüber dem Koordinatensystem des vorhergehenden Schutzbereichselements um folgenden Vektor verschoben:

- X-Richtung: 25.0
- Y-Richtung: 50.0
- Z-Richtung: 37.25



X, Y, Z Koordinatensystem des vorhergehenden Schutzbereichselements

X', Y', Z' Koordinatensystem des aktuellen Schutzbereichselements

Programmcode	Kommentar
; 19. Schutzbereichselement, Verschiebungsvektor	
N100 \$NP_OFF[18,0] = 25.0	X = 25.0
N110 \$NP_OFF[18,1] = 50.0	Y = 50.0
N120 \$NP_OFF[18,2] = 37.25	Z = 37.25

6.2.4.12 \$NP_DIR

Funktion

In die Systemvariable ist der Richtungsvektor einzutragen, um den das lokale Koordinatensystem des Schutzbereichselements zum Koordinatensystem des vorhergehenden Schutzbereichselements gedreht ist. Der Drehwinkel ist in \$NP_ANG (Seite 278) einzutragen.

Randbedingungen

- Der Betrag des Richtungsvektors muss größer sein als: $1 \cdot 10^{-6}$
- Eine in \$NP_OFF (Seite 275) parametrisierte Nullpunktverschiebung wird vor der Drehung ausgeführt.

Syntax

\$NP_DIR[<n>,<i>] = <Wert>

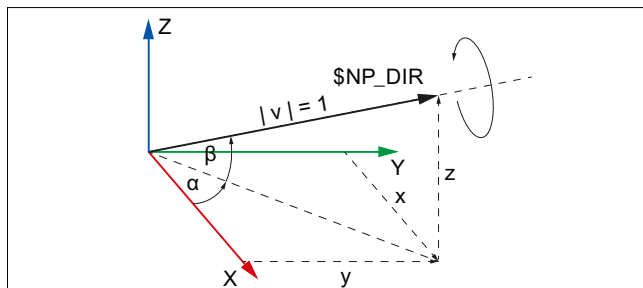
Bedeutung

\$NP_DIR:	Richtungsvektor	
	Datentyp:	REAL
	Wertebereich:	- max. REAL-Wert ≤ x ≤ ± max. REAL-Wert
	Defaultwert:	(0.0, 0.0, 0.0)
<n>:	Systemvariablen- bzw. Schutzbereichselementindex	
	Datentyp:	INT
	Wertebereich:	0, 1, 2, ... (\$MN_MM_MAXNUM_3D_PROT_AREA_ELEM - 1)
<i>:	Koordinatenindex	
	Datentyp:	INT
	Wertebereich:	0 → X; 1 → Y; 2 → Z
<Wert>:	Koordinatenwert	
	Datentyp:	REAL
	Wertebereich:	- max. REAL-Wert ≤ x ≤ ± max. REAL-Wert

Beispiel

Das lokale Koordinatensystem des 19. Schutzbereichselements ist gegenüber dem Koordinatensystem des vorhergehenden Schutzbereichselements um den Richtungsvektor gedreht. Der Richtungsvektors ist der Einheitsvektor (1; 0; 0), gedreht um $\alpha=90^\circ$ in der X/Y-Ebene und $\beta=10^\circ$ in der Y/Z Ebene, bezogen auf das Weltkoordinatensystem. Daraus ergeben sich folgende Werte für die einzelnen Komponenten des Richtungsvektors:

- X-Komponente = $\cos(\alpha) * \cos(\beta) = \cos(90) * \cos(10) = 0,0$
- Y-Komponente = $\sin(\alpha) * \cos(\beta) = \sin(90) * \cos(10) \approx 0,985$
- Z-Komponente = $\sin(\beta) = \sin(10) \approx 0,174$



Programmcode	Kommentar
; 19. Schutzbereichselement, Richtungsvektors	
N100 \$NP_DIR[18,0] = COS(90) * COS(10)	; 0 = X-Komponente
N110 \$NP_DIR[18,1] = SIN(90) * COS(10)	; 1 = Y-Komponente
N120 \$NP_DIR[18,2] = SIN(10)	; 2 = Z-Komponente

6.2.4.13 \$NP_ANG

Funktion

In die Systemvariable ist der Winkel einzutragen um den das lokale Koordinatensystem des Schutzbereichselements zum Koordinatensystem des vorhergehenden Schutzbereichselements um den Richtungsvektor (\$NP_DIR (Seite 276)) gedreht ist.

Syntax

\$NP_ANG [<n>] = <Wert>

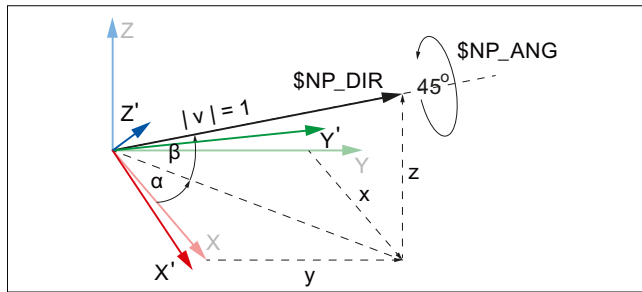
Bedeutung

\$NP_ANG:	Drehwinkel	
	Datentyp:	REAL
	Wertebereich:	$-360^\circ < x \leq 360^\circ$
	Defaultwert:	0.0
<n>:	Systemvariablen- bzw. Schutzbereichselementindex	
	Datentyp:	INT
	Wertebereich:	0, 1, 2, ... (\$MN_MM_MAXNUM_3D_PROT_AREA_ELEM - 1)
<Wert>:	Winkel	
	Datentyp:	REAL

Beispiel

Das lokale Koordinatensystem des 19. Schutzbereichselements ist gegenüber dem Koordinatensystem des vorhergehenden Schutzbereichselements um den Winkel um $\delta=45.0^\circ$ um den Richtungsvektor gedreht. Der Richtungsvektors ist der Einheitsvektor (1; 0; 0), gedreht um $\alpha=90^\circ$ in der X/Y-Ebene und $\beta=10^\circ$ in der Y/Z Ebene, bezogen auf das Weltkoordinatensystem. Daraus ergeben sich folgende Werte für die einzelnen Komponenten des Richtungsvektors:

- X-Komponente = $\cos(\alpha) * \cos(\beta) = \cos(90) * \cos(10) = 0,0$
- Y-Komponente = $\sin(\alpha) * \cos(\beta) = \sin(90) * \cos(10) \approx 0,985$
- Z-Komponente = $\sin(\beta) = \sin(10) \approx 0,174$
- Winkel $\delta = 45.0^\circ$



Programmcode	Kommentar
; 19. Schutzbereichselement, Richtungsvektors und Drehwinkel	
N100 \$NP_DIR[18,0] = COS(90)*COS(10)	; 0 = X-Komponente
N110 \$NP_DIR[18,1] = SIN(90)*COS(10)	; 1 = Y-Komponente
N120 \$NP_DIR[18,2] = SIN(10)	; 2 = Z-Komponente
N130 \$NP_ANG[18] = 45.0	; Drehwinkel $\delta = 45^\circ$

6.2.5 Systemvariablen: Schutzbereichselemente für automatische Werkzeugschutzbereiche

Das Schutzbereichselement eines automatischen Werkzeugschutzbereichs wird durch die nachfolgenden Systemvariablen beschrieben. Die Werte der Systemvariablen werden von der Steuerung automatisch aus den Geometriedaten des zugeordneten Werkzeugs erzeugt und können nur gelesen werden.

Name ¹⁾	Bedeutung	Analog zu ²⁾
\$NP_T_NAME[<n>]	Name des Schutzbereichselements	\$NP_NAME (Seite 258)
\$NP_T_TYPE[<n>]	Typ des Schutzbereichselements	\$NP_TYPE (Seite 265)
\$NP_T_FILENAME[<n>]	Dateiname der STL-Datei, welche die Geometriedaten des Schutzbereichselements enthält (nur relevant bei \$NP_T_TYPE == "FILE")	\$NP_FILENAME (Seite 269)
\$NP_T_PARA[<n>,<i>]	Geometrische Parameter des Schutzbereichskörpers (nur relevant bei \$NP_T_TYPE == "BOX" oder "SPHERE" oder "CYLINDER")	\$NP_PARA (Seite 274)
\$NP_T_OFF[<n>,<i>]	Verschiebungsvektor des lokalen Koordinatensystems des Schutzbereichselements	\$NP_OFF (Seite 275)
\$NP_T_DIR[<n>,<i>]	Richtungsvektor für die Drehung des lokalen Koordinatensystems des Schutzbereichselements	\$NP_DIR (Seite 276)
\$NP_T_ANG[<n>]	Winkel für die Drehung des lokalen Koordinatensystems des Schutzbereichselements	\$NP_ANG (Seite 278)
1) n = 0, 1, ... (\$MN_MM_MAXNUM_3D_T_PROT_ELEM - 1)		
2) Die Systemvariablen der automatischen Werkzeugschutzbereiche entsprechen denen der Maschinenschutzbereiche.		

6.2.6 Randbedingungen

Schutzbereichskörper bei Spindeln

Bei einer Spindel, die sich nicht im lagegeregelten Betrieb befinden, werden die mit ihr verbundenen Schutzbereichskörper ausschließlich statisch modelliert. Daher müssen bei der Modellierung von Schutzbereichskörpern, die mit einer Spindel als kinematischem Element verbunden ist, folgende Randbedingungen eingehalten werden:

- Der Schutzbereichskörper muss **rotationssymmetrisch** sein.
- Die Symmetrieachse des Schutzbereichskörpers muss auf der Drehachse der Spindel liegen (**kollinear**)

Dies ist für alle Arten von Schutzbereichskörpern zu beachten:

- Der Schutzbereichskörper ist ein einfacher geometrischer Grundkörper (Kugel, Zylinder).
- Der Schutzbereichskörper ist aus mehreren geometrischen Grundkörpern zusammengesetzt.
- Der Schutzbereichskörper ist aus Dreiecken aufgebaut (STL-Datei).
- Der Schutzbereichskörper wird für einen automatischen Werkzeugschutzbereich aus den Geometriedaten des Werkzeugs erzeugt.

Hinweis

Automatischen Werkzeugschutzbereiche

Es wird empfohlen, bei automatischen Werkzeugschutzbereichen im Zusammenhang mit Spindeln nur rotationssymmetrische Werkzeuge zu verwenden.

Nachfolgende Schutzbereiche

Die Rotationssymmetrie und Kollinearität des Schutzbereichskörpers bezüglich der Drehachse der Spindel, müssen auch bei allen Schutzbereichen eingehalten werden, die mit Elemente der kinematischen Kette verbunden sind (\$NP_NEXT, \$NP_NEXTP), die auf die Spindel folgenden.

Schematisches Beispiel einer derartigen kinematischen Kette: ... → (Rundachse/Spindel) → (Offset) → (Linearachse) → (Offset) → ...

Rundachsen

Die oben genannten Randbedingungen sind auch im Zusammenhang mit Rundachsen zu beachten, wenn diese auch als Spindeln betrieben werden.

Werkzeugbezugspunkt und kinematische Transformation

Prinzipiell wird durch die Zuordnung eines automatischen Werkzeugschutzbereichs zu einem Element der kinematischen Kette, die Lage des zugehörigen Werkzeugbezugspunkts festgelegt. Die Lage des Werkzeugbezugspunkts kann aber durch Offsets innerhalb des automatischen Werkzeugschutzbereichs über die Systemvariablen \$NP_T_OFF, \$NP_T_DIR und \$NP_T_ANG (Seite 279) verändert werden. Eine derartige Lageänderung des Werkzeugbezugspunkts wird von der kinematische Transformationen nicht erfasst. Es wird deshalb im Zusammenhang mit kinematischen Transformationen dringend empfohlen, für die geometrische Maschinenmodellierung keine Offsets zu verwenden, die den Werkzeugbezugspunkt verschieben.

6.3 Datenlisten

6.3.1 Maschinendaten

6.3.1.1 NC-spezifische Maschinendaten

Nummer	Bezeichner: \$MN_	Beschreibung
MD18890	\$MN_MM_MAXNUM_3D_PROT_AREAS	Maximale Anzahl von Schutzbereichen
MD18892	\$MN_MM_MAXNUM_3D_PROT_AREA_ELEM	Maximale Anzahl von Schutzbereichselementen
MD18893	\$MN_MM_MAXNUM_3D_T_PROT_ELEM	Maximale Anzahl von Werkzeugschutzbereichselementen
MD18897	\$MN_MM_MAXNUM_3D_INTERFACE_IN	Maximale Anzahl von NC/PLC-Nahtstellensignalen zur Voraktivierung von Schutzbereichen
MD18895	\$MN_MM_MAXNUM_3D_FACETS	Maximale Anzahl von Dreiecken für Schutzbereiche
MD18894	\$MN_MM_MAXNUM_3D_FACETS_INTERN	Maximale Anzahl von Dreiecken für automatisch Werkzeugschutzbereiche
MD18899	\$MN_PROT_AREA_TOOL_MASK	Erzeugungsmodus für automatische Werkzeugschutzbereiche

6.3.2 Systemvariablen

Bezeichner	Beschreibung
\$NP_PROT_NAME	Name des Schutzbereichs
\$NP_CHAIN_ELEM	Name des kinematischen Elements, mit dem der Schutzbereich verbunden wird
\$NP_PROT_TYPE	Typ des Schutzbereichs
\$NP_1ST_PROT	Name des ersten Schutzbereichselements des Schutzbereichs
\$NP_PROT_COLOR	Transparenz- und Farbwert des Schutzbereichs
\$NP_PROT_D_LEVEL	Detaillierungsgrad des Schutzbereichs

Bezeichner	Beschreibung
\$NP_BIT_NO	Bit-Nummer des Nahtstellensignals zum Aktivieren / Deaktivieren des Schutzbereichs
\$NP_INIT_STAT	Initialisierungsstatus des Schutzbereichs
\$NP_INDEX	Feld zur Adressierung der wirksamen Geometriedaten
\$NP_NAME	Name des Schutzbereichselements
\$NP_NEXT	Name des nachfolgenden Schutzbereichselements
\$NP_NEXTP	Name des verzweigenden Schutzbereichselements
\$NP_COLOR	Transparenz- und Farbwert des Schutzbereichselements
\$NP_D_LEVEL	Detaillierungsgrad des Schutzbereichselements
\$NP_USAGE	Verwendungsart des Schutzbereichselements
\$NP_TYPE	Typ des Schutzbereichselements
\$NP_FILENAME	Name der STL-Datei mit den Geometriedaten des Körpers des Schutzbereichselements
\$NP_PARA	Parameterwerte entsprechend des Typs des Schutzbereichselements
\$NP_OFF	Verschiebungsvektor
\$NP_DIR	Richtungsvektor
\$NP_ANG	Drehwinkel

K9: Kollisionsvermeidung, intern

7.1 Funktionsbeschreibung

7.1.1 Optionen

Die Funktion "Kollisionsvermeidung" ist eine lizenzpflichtige Option. Folgende Ausprägungen stehen zur Verfügung:

- **Kollisionsvermeidung ECO** (Maschine): 6FC5800-0AS03-0YB0
Eigenschaften:
 - Schutz: Maschine - Maschine
 - HMI Visualisierung
 - Nur für einkanalige Steuerungskonfigurationen
 - Schutzbereichselemente: geometrische Primitive
- **Kollisionsvermeidung** (Maschine, Arbeitsraum): 6FC5800-0AS02-0YB0
Eigenschaften:
 - wie Kollisionsvermeidung ECO
 - Schutzbereichselemente: Dateien im STL- und NPP-Format

7.1.2 Merkmale

Die Funktion "Kollisionsvermeidung" dient zur Verhinderung von Kollisionen von Maschinenteilen und Werkzeugschneiden während des Verfahrens von Maschinenachsen. Dazu berechnet die Funktion zyklisch den Abstand der die zu schützenden Körper umhüllenden Schutzbereiche. Nähern sich zwei Schutzbereiche bis auf einen projektierbaren Sicherheitsabstand aneinander an, wird ein Alarm angezeigt und das NC-Programm vor dem entsprechenden Verfahrssatz angehalten (Betriebsart AUTOMATIK, MDA) bzw. die Verfahrbewegung gestoppt (Betriebsart JOG).

Ablauf

Das Einrichten der Kollisionsvermeidung erfolgt in folgenden Schritten:

1. Freigabe der Funktion "Kollisionsvermeidung" durch Setzen der entsprechenden Option (Seite 283).
2. Einstellen der Maschinendaten zur grundlegenden Parametrierung der Funktionen:
 - Kinematische Kette, Kapitel "Maschinendaten (Seite 207)"
 - Geometrische Maschinenmodellierung, Kapitel "Maschinendaten (Seite 244)"
 - Kollisionsvermeidung, Kapitel "Maschinendaten (Seite 295)"

7.1 Funktionsbeschreibung

3. Beschreiben der kinematischen Struktur der Maschine durch kinematische Elemente. Siehe Kapitel "K7: Kinematische Kette (Seite 201)".
4. Beschreiben der Schutzbereiche und Schutzbereichselemente als umhüllende Geometrie der zu schützenden Maschinenteile, Werkzeuge und Werkstücke. Zuordnen der Schutzbereich zu Elementen der kinematischen Kette. Siehe Kapitel "K8: Geometrische Maschinenmodellierung (Seite 235)".
5. Definieren von Kollisionspaaren, d.h. von jeweils zwei Schutzbereichen, die gegenseitig auf Kollision überwacht werden soll. Siehe Kapitel "\$NP_COLL_PAIR (Seite 299)".
6. Neuberechnung des kinematischen und geometrischen Modells auslösen Siehe Kapitel "Neuberechnung des Maschinenmodells der Kollisionsvermeidung anfordern (PROTA) (Seite 305)".
7. Aktivieren der zu überwachenden Schutzbereiche Siehe Kapitel "Schutzbereichszustand setzen (PROTS) (Seite 306)".
8. Optional: Anwendung der erweiternden Funktionen und Systemvariablen

Grenzen der Kollisionsvermeidung

Die Funktion kann keinen vollständigen Schutz vor einer Kollision beim Verfahren von Maschinenteilen, Werkzeugen oder Werkstücken garantieren. Der Kollisionsschutz kann zum einen nur so gut sein, wie das parametrisierte kinematische und geometrische Modell der Maschine und der Schutzbereiche. Zum anderen können nicht modellierte Körper naturgemäß überhaupt nicht überwacht werden. Daher liegt es nach wie vor auch mit aktiviertem Kollisionsschutz in der Verantwortung des Maschinenbedieners, dass eine Verfahrbewegung kollisionsfrei ausgeführt werden kann.

Zustände von Schutzbereichen

Ob ein Schutzbereich in der Kollisionsbetrachtung berücksichtigt wird, ist abhängig vom Zustand des Schutzbereichs:

Zustand	Bedeutung
Aktiv	Der Schutzbereich wird in der Kollisionsbetrachtung berücksichtigt.
Inaktiv	Der Schutzbereich wird in der Kollisionsbetrachtung nicht berücksichtigt.
Voraktiviert	Der Schutzbereich wird in der Kollisionsbetrachtung berücksichtigt. Es wird aber nur dann einen Kollisionsalarm ausgelöst, wenn er zusätzlich durch sein Schutzbereichspezifisches NC/PLC-Nahtstellensignal aktiviert wurde.

Zustand nach Steuerungshochlauf

Nach dem Steuerungshochlauf befinden sich alle Schutzbereiche im Zustand entsprechend ihrer jeweiligen Einstellung in \$NP_INIT_STAT. Siehe Kapitel "\$NP_INIT_STAT (Seite 254)".

Zustandsänderung

Der Zustand eines Schutzbereichs kann geändert werden durch:

- Prozedur PROTS() (Seite 306)
- Änderung des Initialisierungszustandes in \$NP_INIT_STAT und anschließender Neuberechnung des Maschinenmodells durch die Prozedur PROTA() (Seite 305).

Voraussetzungen

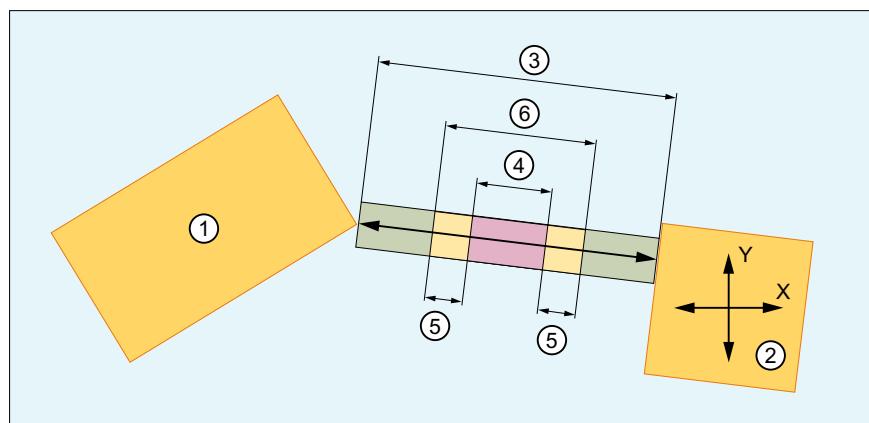
Damit die Schutzbereiche eines Kollisionspaares überwacht werden können, müssen folgende Voraussetzungen erfüllt sein:

- Achsen bzw. Spindeln: Referenziert/Synchronisiert
Die Lagemesssystem der Achsen bzw. Spindel, die einen Schutzbereich bewegen, müssen referenziert bzw. synchronisiert sein. Ist dies nicht der Fall, befindet sich der entsprechende Schutzbereich im Zustand "Inaktiv".
- Externe Bewegungen
Bei nicht von der NC ausgeführten Verfahrbewegungen, z.B. PLC-Achse oder manuell bewegte Achse, müssen die aktuellen Achspositionen der NC bekannt sein.

7.1.3 Reaktion der Steuerung bei Kollisionsgefahr

Die Kollisionsvermeidung beachtet bei der Kollisionserkennung folgende parametrierbaren Grenzwerte:

- Kollisionstoleranz
- Sicherheitsabstand



- ① Schutzbereich 1 (unbeweglich)
- ② Schutzbereich 2 (beweglich in X- und Y-Richtung)
- ③ Aktueller Abstand
- ④ Sicherheitsabstand
- ⑤ Kollisionstoleranz / 2
- ⑥ Kollisionsabstand = Sicherheitsabstand + Kollisionstoleranz

Bild 7-1 Aktueller Abstand, Kollisionstoleranz und Sicherheitsabstand

Kollisionstoleranz und Sicherheitsabstand

Sicherheitsabstand

Der Sicherheitsabstand definiert einen Abstand, bis zu dem sich zwei aktive und auf Kollision überwachte Schutzbereiche maximal annähern dürfen. Die Kollisionsvermeidung stellt sicher, dass dieser Abstand nicht unterschritten und die Kollision angezeigt wird.

Der Sicherheitsabstand kann kollisionspaarspezifisch über eine Systemvariable (Seite 300) eingestellt werden.

Für alle Kollisionspaare, für die über die Systemvariable kein spezifischer Sicherheitsabstand eingestellt wird, gilt der allgemeine über MD10622 \$MN_COLLISION_SAFETY_DIST (Seite 296) eingestellte Wert.

Kollisionstoleranz

Die Kollisionstoleranz definiert einen NC-weit gültigen zum Sicherheitsabstand zusätzlichen additiven Abstand. Zwei aktive und auf Kollision überwachte Schutzbereiche dürfen sich somit einander bis auf den Kollisionsabstand (Sicherheitsabstand + Kollisionstoleranz) annähern. Im Idealfall wird durch die Kollisionsvermeidung die Verfahrbewegung der Schutzbereiche exakt im Kollisionsabstand gestoppt und die Kollision angezeigt. Es ist jedoch zulässig, dass die Kollisionstoleranz nicht exakt eingehalten wird, oder bei kurzzeitiger Unterschreitung nicht auf Kollision erkannt und die Verfahrbewegung nicht gestoppt wird.

Die Kollisionstoleranz wird über MD10619 \$MN_COLLISION_TOLERANCE (Seite 295) für alle Kollisionspaare gleich eingestellt.

Hinweis

Unterschied zwischen Kollisionstoleranz und Sicherheitsabstand

Ein Unterschreiten der Kollisionstoleranz kann auftreten und ist zulässig. Der Sicherheitsabstand wird immer eingehalten.

Reaktionen in Betriebsart AUTOMATIK

Kollisionserkennung im Vorlauf

Im Automatikbetrieb werden bereits im Vorlauf die Verfahrätze des aktiven Programms auf Kollision geprüft. Wird dabei bereits eine Kollision erkannt, erfolgen folgende Reaktionen:

- Stopp der Verfahrbewegungen im Kanal
- NC/PLC-Nahtstellensignal: DB21,DBX377.0 = 1 (Kollisionsvermeidung: Stopp)
- Anzeige von Alarm 26260 mit der Satznummer des betreffenden Verfahrsatzes
- Abbruch der Programmbearbeitung

Kritische Annäherung

Auch im Automatikbetrieb können überlagernde oder nebenläufige Bewegungen auftreten, die nicht vorab berücksichtigt werden können. Daher wird bei einer kritischen Annäherung von

Schutzbereichen die Verfahrgeschwindigkeit verringert oder die Verfahrbewegung ganz gestoppt:

- Achsspezifisches NC/PLC-Nahtstellensignal bei Verringerung der Verfahrgeschwindigkeit: DB31,DBX77.0 == 1 (Kollisionsvermeidung: Geschwindigkeitsreduzierung)
- Kanalspezifisches NC/PLC-Nahtstellensignal bei Stopp der Verfahrbewegung: DB21,DBX377.0 == 1 (Kollisionsvermeidung: Stopp)

Voraktivierte Schutzbereiche

Wird bei der Satzaufbereitung im **Vorlauf** festgestellt, dass in einem Verfahrstz zwei Schutzbereiche, von denen mindestens einer nur **voraktiviert** ist, kollidieren würden, wenn sie aktiv wären, führt das noch nicht zu den oben unter "Kollisionserkennung im Vorlauf" beschriebenen Reaktionen. Die Reaktionen erfolgen erst, wenn beide Schutzbereiche aktiviert werden.

Wird der Satz zum Aktivierungszeitpunkt bereits im Hauptlauf verfahren, wird aufgrund der Kollisionsberechnung im Vorlauf auf Kollision erkannt und die oben genannten Reaktionen ausgelöst. Die Erkennung auf Kollision erfolgt unabhängig davon, ob zum Aktivierungszeitpunkt die Schutzbereiche tatsächlich kollidieren.

Reaktionen in Betriebsart JOG

Nähern sich zwei Schutzbereiche beim Verfahren in der Betriebsart JOG einander an, wird die Verfahrgeschwindigkeit kontinuierlich bis zum Stillstand bei Erreichen des Kollisionsabstandes abgebremst. Mit Erreichen des Kollisionsabstandes wird der Alarm 26280 angezeigt.

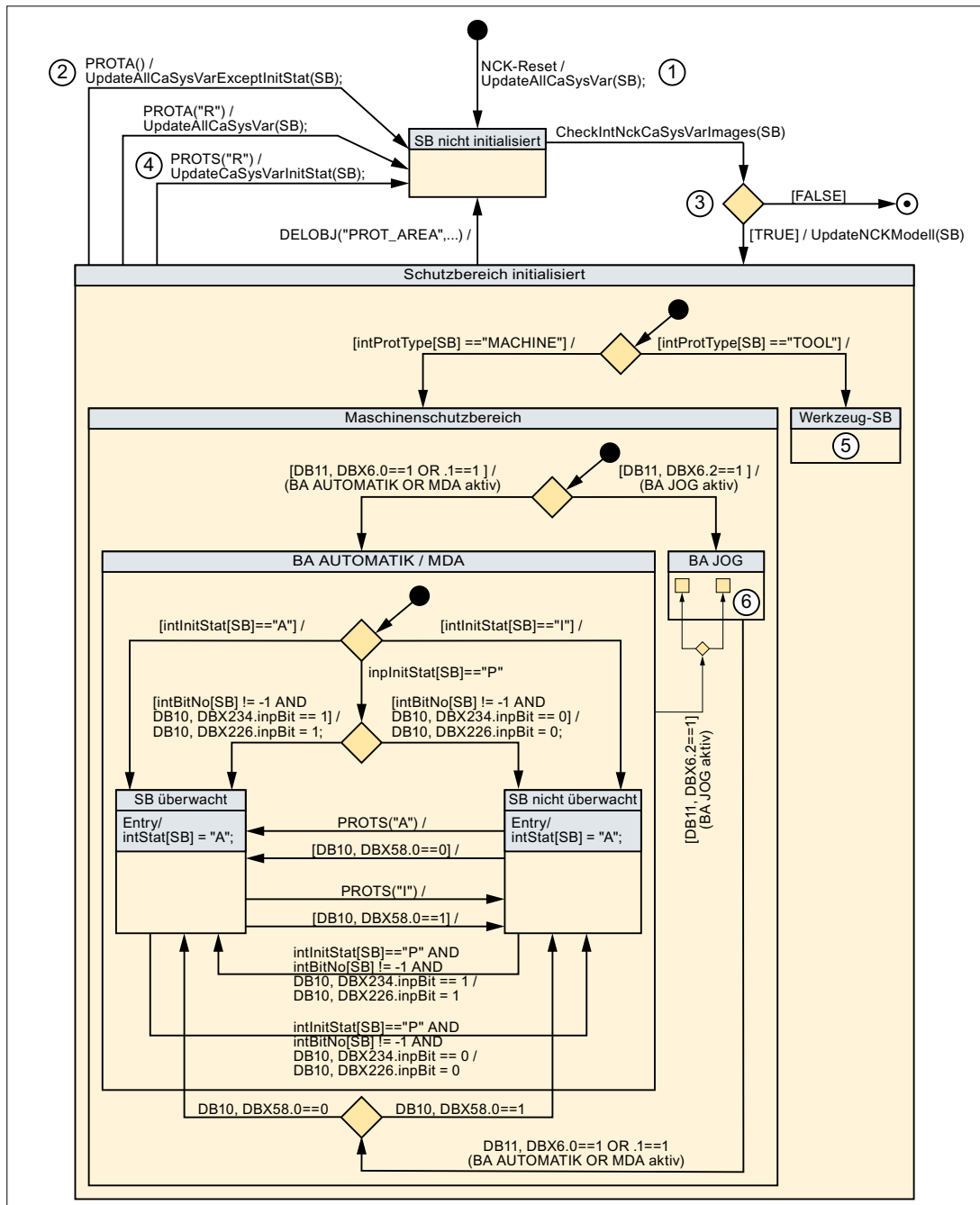
Wird die Kollisionsstelle in Gegenrichtung wieder verlassen, ist, abhängig vom Abstand der Schutzbereiche, wieder eine kontinuierlich höhere Verfahrgeschwindigkeit möglich.

Eine Verfahrbewegung wird mit Erreichen des Kollisionsabstandes immer **abgebrochen**. Ein Fortsetzen der Verfahrbewegung erfordert, unabhängig von der Verfahrrichtung, immer eine erneute Fahrenforderung (z. B. Betätigung einer Verfahrtaste).

Reaktionen in Betriebsart MDA

Nähern sich zwei Schutzbereiche beim Verfahren in der Betriebsart MDA einander an, wird die Verfahrgeschwindigkeit kontinuierlich bis zum Stillstand bei Erreichen des Kollisionsabstandes abgebremst. Mit Erreichen des Kollisionsabstandes wird der Alarm 26260 angezeigt.

7.1.4 Zustandsdiagramm: Schutzbereich



SB Schutzbereich

BA Betriebsart

① Funktion UpdateAllCaSysVar(SB)

Alle Systemvariablen der Kollisionsvermeidung werden in NC-interne Variablen eingelesen:

int... = \$N...

- ② Funktion UpdateAllCaSysVarExeptInitStat(SB)
Wie Funktion UpdateAllCaSysVar(SB), aber die Systemvariable \$NP_INIT_STAT wird nicht eingelesen. NC-intern bleibt dadurch der letzte Wert des Initialisierungsstatus intInitStat erhalten.
- ③ Funktion CheckIntNckCaSysVarImages(SB)
Die aus den Systemvariablen eingelesenen NC-internen Variablen werden auf Konsistenz überprüft.
Rückgabewert bei erkanntem Fehler: FALSE; bei Fehlerfreiheit: TRUE.
- ④ Funktion UpdateCaSysVarInitStat(SB)
Es wird nur die Systemvariable \$NP_INIT_STAT in die NC-internen Variable intInitStat eingelesen.
- ⑤ Der interne Aufbau des Zustands "Werkzeug-SB" ist gleich dem des Zustands "Maschinenschutzbereich".
- ⑥ Der interne Aufbau des Zustands "BA JOG" ist gleich dem des Zustands "BA AUTO / MDA".

7.1.5 Werkzeuge

Modellierung

Schutzbereiche für Werkzeuge können von der Kollisionsvermeidung automatisch modelliert und nach einem Werkzeugwechsel, mit Einschränkungen, automatisch aktualisiert werden. Dazu müssen folgende Voraussetzungen erfüllt sein:

- Der Schutzbereich für das Werkzeug ist als automatischer Werkzeugschutzbereich (Type: TOOL) modelliert. Siehe Systemvariable \$NP_PROT_TYPE (Seite 248)
- Das Werkzeug wird durch die Werkzeugverwaltung der Steuerung verwaltet.
- Die in der Werkzeugverwaltung hinterlegten Werkzeugdaten stimmen mit den tatsächlichen geometrischen Abmessungen des Werkzeugs überein.
- Die Kollisionsvermeidung erkennt den Abschluss des Werkzeugwechsels. Im Normalfall durch Programmierung der entsprechenden Werkzeugkorrekturnummer D_x , mit $x = 0, 1, 2, 3, \dots$
- Die Kollisionsvermeidung erkennt, bei welchem automatischen Werkzeugschutzbereich das Werkzeug gewechselt wurde.
- Bezieht sich die Werkzeugbahn auf ein normiertes Werkzeug, wird der Werkzeugradius des aktuellen Werkzeugs als positive bzw. negative Abweichungen bezogen auf das normierte Werkzeug angegeben. Die Kollisionsvermeidung rechnet in diesem Fall mit folgenden Werten:
 - positiver Wert: Werkzeugradius = angegebener Wert, mindestens aber der Wert der parametrisierten Kollisionstoleranz
 - negativer Wert: Werkzeugradius = Wert der parametrisierten Kollisionstoleranz

Änderung des Maschinenmodells

Wird in der Maschine ein Werkzeug, das sich in einem im aktiven Maschinenmodell der Kollisionsvermeidung modellierten Magazin oder Werkzeugaufnahme befindet, geändert, muss das Maschinenmodell aktualisiert werden. Das ist der Fall, wenn z.B. eine der folgenden Aktionen ausgeführt wird:

- In einem im Maschinenmodell modellierten Werkzeugmagazin wird ein Werkzeug beladen / entladen.
Beispiel: Werkzeugwechsel in einem Revolvermagazin.
Aktualisierung: Die Aktualisierung des Maschinenmodells muss vom Anwender nach Abschluss des Werkzeugwechsels mittels `PROTA` explizit angefordert werden.
- Das Werkzeug, das sich in einem Werkzeughalter befindet, wird gewechselt.
Beispiel: Werkzeugwechsel im Werkzeughalter der Hauptspindel.
Aktualisierung: Die Aktualisierung des Maschinenmodells erfolgt automatisch nach dem Werkzeugwechsel (Standard: `M6`) mit Ausgabe der programmierten Werkzeugkorrekturnummer `Dx` im Hauptlauf.

Werkzeugänderungen

Die Kollisionsvermeidung aktualisiert das aktive Maschinenmodell nach einer Werkzeugänderung ohne explizite Anforderung durch `PROTA` nur, wenn dabei eine Werkzeugkorrekturanwahl (Ausgabe der programmierten Werkzeugkorrekturnummer `Dx` an die NC/PLC-Nahtstelle) erfolgt.

An der Steuerung können aber auch Werkzeugänderungen durchgeführt werden, die nicht mit einer Werkzeugkorrekturanwahl verbunden sind. Bei diesen erfolgt keine Aktualisierung des aktiven Maschinenmodells. Derartige Werkzeugänderungen sind beispielsweise:

- Beladen / Entladen eines Werkzeugs über die Bedienoberfläche.
SINUMERIK Operate: Bedienbereich "Parameter" > "Werkzeugliste" > vertikaler Softkey: "Beladen" oder "Entladen"
- Durchführen des Werkzeugwechsels über das PLC-Anwenderprogramm
- Direktes Beschreiben des Werkzeugwischenspeichers über Systemvariable
`$TC_MPP6[9998, <Platz>]`
- `SETMS (<Spindelnummer>)`: Wechsel der Masterspindel im Kanal
- `TMMVTL`: PI-Dienst "Magazinplatz zum Beladen bereitstellen, Werkzeug entladen"
Literatur Funktionshandbuch Grundfunktionen; Kapitel: "P3: PLC-Grundprogramm für SINUMERIK 840D sl" > "Bausteinbeschreibungen" > "PI-Dienste" > "PI-Dienst: TMMVTL"
- `MVTOOL`: Befehl zum Bewegen eines Werkzeugs
Literatur Funktionshandbuch Werkzeugverwaltung; Kapitel "Programmierung" > "NC-Sprachbefehle" > "MVTOOL - Sprachbefehl zum Bewegen eines Werkzeugs"

Wird eine derartige Werkzeugänderung durchgeführt, muss die Aktualisierung des Maschinenmodells vom Maschinenhersteller über das PLC-Anwenderprogramm angefordert werden. Beispielhafte Möglichkeiten dazu sind:

- Ist der Kanal im Zustand "Reset", wird ein erneuter Kanal-Reset angefordert. Bei entsprechender Einstellung des Reset-Verhaltens (MD20110 \$MC_RESET_MODE_MASK) erfolgt dann eine erneute Ausgabe der aktuellen Werkzeugkorrekturnummer D_x .
- Starten eines ASUP oder Hersteller-Zyklus, der die Ausgabe der Werkzeugkorrekturnummer D_x und die Anforderung zum Aktualisieren des Maschinenmodells *PROTA* enthält.

Werkzeugverschleiß

Minimale Werkzeugänderungen müssen im Maschinenmodell nicht berücksichtigt werden, da sie gewöhnlich sehr viel kleiner als der Kollisionsabstand sind.

Entstehen für die Kollisionsvermeidung relevante Werkzeugänderung, z.B. Durchmesseränderungen bei Schleifwerkzeugen, müssen diese durch explizite Anforderung zum Aktualisieren des Maschinenmodells (*PROTA*) berücksichtigt werden.

Keine Änderung des Maschinenmodells

Das aktive Maschinenmodell der Kollisionsvermeidung ändert sich **nicht**, wenn in der Maschine ein vollständig modelliertes Maschinenteil mit Werkzeugen, z.B. ein Werkzeugmagazin, bewegt wird.

Beispiel: Revolvermagazin einer Drehmaschine

Im Maschinenmodell der Kollisionsvermeidung ist das Revolvermagazin einer Drehmaschine vollständig modelliert:

- Die Geometrie des Magazins und der sich darin befindlichen Werkzeuge
- Die Bewegungen des Magazins durch die Maschinenachsen

Eine Drehung des Revolvermagazins stellt dann keine Änderung des Maschinenmodells dar:

- Da keine Werkzeuge innerhalb des Maschinenmodells verändert werden, bleiben die Geometrien aller Schutzbereiche unverändert.
- Da die Bewegungen der Schutzbereiche durch die Maschinenachsen, von der Kollisionsvermeidung über die kinematische Kette vollständig erfasst werden.

Randbedingungen

Mehrere Spindeln im Kanal

Bei Konfigurationen mit mehreren Spindeln im Kanal geht die Kollisionsvermeidung davon aus, dass ein Werkzeugwechsel in der Masterspindel des Kanals (S1) erfolgt. Durch die Kollisionsvermeidung wird daher nach erfolgtem Werkzeugwechsel ausschließlich der automatische Werkzeugschutzbereich der Masterspindel aktualisiert.

Nicht unterstützte Werkzeugkonfigurationen

Werkzeugkonfigurationen gemäß ISO-Mode 4 und 5 (H-Nummern), sowie "Flache D-Nummern" werden von der Kollisionsvermeidung nicht unterstützt.

7.1.6 Randbedingungen

Kanalzuordnung

Alle für die Kollisionsvermeidung relevanten Komponenten der Maschine müssen dem **ersten Kanal** der NC zugeordnet sein:

- Alle **Achsen** und **Spindeln** der kinematischen Kette.
Siehe Kapitel "K7: Kinematische Kette (Seite 201)"
- Alle **Werkzeuge** von automatischen Werkzeugschutzbereichen der geometrischen Maschinenmodellierung.
Siehe Kapitel "K8: Geometrische Maschinenmodellierung (Seite 235)"

Berücksichtigung des Schleppabstandes

Die Kollisionsvermeidung verwendet für die Abstandsberechnung der Schutzbereiche die Sollpositionen der beteiligten Maschinenachsen. Die Istpositionen der Maschinenachsen weichen aber um den jeweiligen Schleppabstand von der Sollposition ab. Daher ergibt sich auch für die Schutzbereiche eine Abweichung der Soll- von der Istposition. Diese Abweichung muss vom Anwender durch Projektierung eines ausreichend großen Sicherheitsabstandes oder Vergrößerung des Schutzbereiches berücksichtigt werden.

Kompensationen

Die verschiedenen Kompensationsfunktionen der NC, z.B. Temperatur-, Spindelsteigungsfehler- und Durchhangkompensation, sorgen dafür, dass die im Werkstückkoordinatensystem programmierten Positionen im Maschinenkoordinatensystem tatsächlich eingenommen werden. Die von den Kompensationen vorgenommenen Positionskorrekturen werden von der Kollisionsvermeidung mit eingerechnet.



WARNUNG

Kollisionsgefahr

Werden Kompensationen zweckentfremdet eingesetzt, z.B. um Funktionen wie Achskopplungen im Maschinenkoordinatensystem zu realisieren, kann die mit Sollpositionen arbeitende Kollisionsvermeidung nicht mehr zuverlässig durchgeführt werden. Es besteht Kollisionsgefahr.

Istwertverschiebung im Maschinenkoordinatensystem PRESETON

Bei aktiver Kollisionsvermeidung und Anwendung einer Istwertverschiebung im Maschinenkoordinatensystem PRESETON liegt es in der alleinigen Verantwortung des Anwenders, das geometrische Modell der Kollisionsvermeidung konsistent zu halten.

 WARNUNG
Kollisionsgefahr
Wird durch PRESETON eine Istwertverschiebung im Maschinenkoordinatensystem vorgenommen und das geometrische Modell der Kollisionsvermeidung nicht entsprechend angepasst, kann die mit Sollpositionen arbeitende Kollisionsvermeidung nicht mehr zuverlässig durchgeführt werden. Es besteht Kollisionsgefahr.

Satzsuchlauf

Bei folgenden Satzsuchlaufarten werden von der Kollisionsvermeidung **keine** Kollisionsberechnungen durchgeführt:

- Typ 1: Satzsuchlauf ohne Berechnung
- Typ 2: Satzsuchlauf mit Berechnung an Kontur
- Typ 4: Satzsuchlauf mit Berechnung an Satzendpunkt

Bei folgender Satzsuchlaufart werden von der Kollisionsvermeidung Kollisionsberechnungen (im Vorlauf) durchgeführt bei:

- Typ 5: Satzsuchlauf mit Berechnung im Modus "Programmtest" (SERUPRO)

Betriebsarten AUTOMATIK: Unvollständige Schutzbereichsangabe bei Kollision

Ist eine großen Anzahl von Schutzbereichen projiziert, kann es in Ausnahmefällen zu folgendem Verhalten kommen:

- Mehrere Schutzbereiche haben sich gleichzeitig bis auf Kollisionstoleranz angenähert
- Im angezeigten Alarm 26260 "Kollision **zweier** Schutzbereiche" werden nur zwei Schutzbereiche genannt.
- Erst nach einem Wechsel in die Betriebsart JOG wird beim manuellen Verfahren von Achsen die Kollision der anderen Schutzbereiche angezeigt.

7.2 Inbetriebnahme

7.2.1 Allgemein

7.2.1.1 Übersicht

Die Inbetriebnahme der Funktion "Kollisionsvermeidung" erfolgt mittels:

- Maschinendaten
 - Vorgabe des Mengengerüsts
 - Festlegung allgemeiner Eigenschaften der Kollisionspaare
- Systemvariablen
 - Parametrierung der Kollisionspaare und deren Eigenschaften

7.2.1.2 Aufbau der Systemvariablen

Die Systemvariablen sind nach folgendem Schema aufgebaut:

\$NP_<Name>[<Index_1>,<Index_2>]

Hinweis

Index_2 ist nicht bei allen Systemvariablen vorhanden.

Allgemein

Die Systemvariablen zur Beschreibung von Schutzbereichen haben folgende Eigenschaften:

- Präfix: **\$NP_**, (N für NC, P für Protection).
- Sie sind über NC-Programme les- und schreibbar.
- Sie können über Archive gesichert und wieder in die NC eingelesen werden.

Datentyp

STRING

Alle Systemvariablen vom Datentyp STRING haben folgende Eigenschaften:

- Maximale String-Länge: 31 Zeichen
- Es wird keine Unterscheidung zwischen Groß- und Kleinschreibung gemacht
Beispiel: "Achse1" identisch mit "ACHSE1"

- Leer- und Sonderzeichen sind zulässig
Beispiel: "Achse1" nicht identisch mit " Achse 1"
- Namen, die mit **zwei** Unterstrichen "__" **beginnen**, sind für Systemzwecke reserviert und dürfen **nicht** für anwenderdefinierte Namen verwendet werden.

Hinweis**Führendes Leerzeichen**

Da Leerzeichen gültige und der Unterscheidung dienende Zeichen sind, dürfen Namen, die mit einem **Leerzeichen**, gefolgt von **zwei** Unterstrichen "__" **beginnen**, prinzipiell für anwenderdefinierte Namen verwendet werden. Aufgrund der Verwechslungsgefahr mit Systemnamen wird dieses Vorgehen **nicht** empfohlen.

Index_1

Über Index_1 werden die einzelnen Schutzbereiche adressiert. Index 0 → 1. Schutzbereich, Index 1 → 2. Schutzbereich, ... n → (n+1) Schutzbereich, mit n = (\$MN_MM_MAXNUM_3D_PROT_AREAS - 1)

Alle Systemvariablen eines Schutzbereichs haben den gleichen Index.

Index_2

Bei Systemvariablen, die ein Kollisionspaar definieren, werden über Index_2 die Schutzbereiche des Kollisionspaars adressiert.

- 0 → 1. Schutzbereich
- 1 → 2. Schutzbereich

Siehe auch

Löschen von Komponenten (DELOBJ) (Seite 225)

7.2.2 Maschinendaten**7.2.2.1 Kollisionstoleranz**

Mit dem Maschinendatum wird die Kollisionstoleranz (Genauigkeit der Kollisionsprüfung) für alle auf Kollision überwachten Schutzbereiche der NC eingestellt. Wird der Abstand zweier Schutzbereiche kleiner dem Kollisionsabstand, d.h. der Summe aus Sicherheitsabstand (Seite 296) und Kollisionstoleranz, liegt eine Kollision vor.

MD10619 \$MN_COLLISION_TOLERANCE = <Kollisionstoleranz>

Genauigkeit automatisch erzeugter Schutzbereiche

Die Kollisionstoleranz bestimmt auch die Genauigkeit der Schutzbereichskörper von automatisch erzeugten Schutzbereichen, z.B. automatische Werkzeugschutzbereiche. Die

7.2 Inbetriebnahme

Genauigkeit der mittels Dreiecksflächen angenäherten Schutzbereichskörper beträgt 1/3 der Kollisionstoleranz.

Auswirkungen

Je kleiner die Kollisionstoleranz eingestellt wird, umso größer wird die Anzahl der zur Modellierung der automatisch erzeugten Schutzbereiche benötigten Dreiecksflächen und der Rechenzeitbedarf für die Kollisionserkennung.

Einstellempfehlung

Kollisionstoleranz \approx 1 mm

Siehe auch

Reaktion der Steuerung bei Kollisionsgefahr (Seite 285)

7.2.2.2 Sicherheitsabstand

Mit dem Maschinendatum wird der Sicherheitsabstand für alle auf Kollision überwachten Schutzbereiche der NC eingestellt. Die Kollisionsvermeidung gewährleistet, dass der Sicherheitsabstand nicht unterschritten wird.

MD10622 \$MN_COLLISION_SAFETY_DIST = <Sicherheitsabstand>

Hinweis

Kollisionspaarspezifischer Sicherheitsabstand

Wenn für ein Kollisionspaar über die Systemvariable \$NP_SAFETY_DIST (Seite 300) ein spezifischer Sicherheitsabstand eingestellt wurde, hat dieser Vorrang vor dem im Maschinendatum eingestellten NC-spezifischen Sicherheitsabstand.

Siehe auch

Reaktion der Steuerung bei Kollisionsgefahr (Seite 285)

7.2.2.3 Maximaler Speicherplatz

Mit dem Maschinendatum wird der Maximalwert des Speicherplatzes in kByte eingestellt, der von der Kollisionsvermeidung belegt werden darf.

MD18896 \$MN_MM_MAXNUM_3D_COLLISION = <Wert>

Wert	Bedeutung
0	Der Maximalwert des Speicherplatzes wird von der Steuerung automatisch anhand folgender Maschinendaten ermittelt: <ul style="list-style-type: none"> • MD18890 \$MN_MM_MAXNUM_3D_PROT_AREAS • MD18892 \$MN_MM_MAXNUM_3D_PROT_AREA_ELEM • MD18894 \$MN_MM_MAXNUM_3D_FACETS_INTERN • MD18895 \$MN_MM_MAXNUM_3D_FACETS
> 0	Maximalwert = parametrierter Wert [kByte]

Hinweis

In das Maschinendatum muss nur ein Wert > 0 eingetragen werden, wenn einer der folgenden Alarme angezeigt wird:

- Alarm 26262 "Kein ausreichender Speicherplatz beim Kollisionstest zweier Schutzbereiche"
- Alarm 26263 "Kein ausreichender Speicherplatz bei der Abstandsbestimmung zweier Schutzbereiche"

Belegter Speicherplatz

Zur Ermittlung des von der Kollisionsvermeidung belegten Speicherplatzes stehen verschiedene Systemvariablen zur Verfügung. Siehe Kapitel "Speicherplatzbedarf (Seite 303)".

7.2.2.4 Maximale Anzahl an Kollisionspaaren

Die maximale Anzahl möglicher Kollisionspaare hat Auswirkungen auf:

- Die Länge m der Systemvariablenfelder (z.B. \$NP_COLL_PAIR[m, ...])
- Den für die Kollisionsvermeidung benötigten Anwenderspeicher
- Die Größe der Inbetriebnahmearchive

Mit dem Maschinendatum kann die maximale Anzahl möglicher Kollisionspaare eingeschränkt werden:

MD18898 \$MN_MM_MAXNUM_3D_COLL_PAIRS = <Wert>

<Wert>	Bedeutung
0	Für die maximale Anzahl möglicher Kollisionspaare MCP gilt: MCP = Maximalwert des Maschinendatums
x > 0	Für die maximale Anzahl möglicher Kollisionspaare MCP gilt: MCP = x, mit $0 < x \leq$ Maximalwert des Maschinendatums
Ein Wert größerer als der zulässige Maximalwert des Maschinendatums, wird intern auf den Maximalwert begrenzt. Es erfolgt dabei keine Rückmeldung an den Anwender.	

7.2.2.5 Schutzstufen für Kollisionsvermeidung Ein/Aus

Mit den Maschinendaten wird die Schutzstufe für das Ein-/Ausschalten der Kollisionsvermeidung über die Bedienoberfläche eingestellt. Die Schutzstufe kann, getrennt nach Betriebsart und Schutzbereichstyp, vorgegeben werden.

Maschinendatum = <Schutzstufe>

Nummer	Bezeichner: \$MN_	Bedeutung: Schutzstufe zum Ein-/Ausschalten der Kollisionsvermeidung
MD51160	ACCESS_WRITE_CA_MACH_JOG	Maschinenschutzbereiche, Betriebsart JOG / MDA
MD51161	ACCESS_WRITE_CA_MACH_AUTO	Maschinenschutzbereiche, Betriebsart AUTOMATIK,
MD51162	ACCESS_WRITE_CA_TOOL	Werkzeugschutzbereiche

Literatur

Eine ausführliche Beschreibung zu Schutzstufen findet sich in:

Funktionshandbuch Grundfunktionen; Kapitel: "A2: Diverse NC/PLC-Nahtstellensignale und Funktionen" > "Funktionen" > "Zugriffsschutz über Kennwort und Schlüsselschalter"

7.2.3 Systemvariablen

7.2.3.1 Übersicht

Mit folgenden Systemvariablen wird ein Kollisionspaar parametrier:

Name	Bedeutung
\$NP_COLL_PAIR	Name eines Schutzbereichs eines Kollisionspaars
\$NP_SAFETY_DIST	Sicherheitsabstand des Schutzbereichspars

Die Systemvariablen sind in den nachfolgenden Kapiteln ausführlich beschrieben.

Hinweis

Definierten Ausgangszustand herstellen

Es wird empfohlen, vor Parametrierung der Kollisionsvermeidung einen definierten Ausgangszustand zu erzeugen. Dazu sind die Systemvariablen der Kollisionsvermeidung mit der Funktion DELOBJ() (Seite 225) auf ihren Default-Wert zu setzen.

Ändern von Systemvariablenwerten

Wird der Wert einer der oben aufgeführten Systemvariablen geändert, wird die Änderung auf der Bedienoberfläche, z.B. SINUMERIK Operate, sofort sichtbar. Das Maschinenmodell der NC wird aber erst nach einer expliziten Anforderung zum Neuberechnen des Maschinenmodells durch Aufruf der Funktion PROTA() (Seite 305) bzw. PROTS() (Seite 306) aktualisiert.

7.2.3.2 \$NP_COLL_PAIR

Funktion

In die Systemvariable werden die Namen der beiden Schutzbereiche eingetragen, die zusammen ein Kollisionspaar bilden. Die Reihenfolge der beiden Schutzbereiche ist dabei beliebig.

Kollisionspaare

Da die Kollisionskontrolle eine sehr rechenzeitintensive Aufgabe ist, ist es nicht sinnvoll durch die Kollisionsvermeidung prinzipiell immer alle parametrisierten Schutzbereiche gegenseitig auf Kollision zu überwachen. Beispiele, in denen eine Kollisionskontrolle nicht sinnvoll ist:

- Schutzbereiche, die konstruktionsbedingt nicht miteinander kollidieren können
- Schutzbereiche, die ohne Verankerung an der kinematischen Kette definiert wurden

Aus der Menge der parametrisierten Schutzbereiche sind vom Anwender diejenigen zu bestimmen, die an der Maschine tatsächlich kollidieren können und als so genannte Kollisionspaare zu definieren. Nur diese Schutzbereiche werden von der Kollisionsvermeidung überwacht.

Zur Definition eines Kollisionspaars sind die Namen der beiden Schutzbereiche in zwei Systemvariablen mit dem gleichen Kollisionspaarindex einzutragen. Ein Schutzbereich unter dem Schutzbereichindex 0, der andere unter dem Schutzbereichindex 1.

Zugehörigkeit zu einem Kollisionspaar

Mit der Funktion COLLPAIR() (Seite 304) kann überprüft werden, ob zwei Schutzbereiche als Kollisionspaar parametrisiert sind.

Syntax

```
$NP_COLL_PAIR[<m>,<i>] = "<Name>"
```

Bedeutung

\$NP_COLL_PAIR:	Name des ersten oder zweiten Schutzbereichs eines Kollisionspaars	
	Datentyp:	STRING
	Defaultwert:	"" (Leerstring)
<m>:	Systemvariablen- bzw. Kollisionspaarindex	
	Datentyp:	INT
	Wertebereich:	0, 1, 2, ... (M -1) ¹⁾
<i>:	Schutzbereichsindex	
	Datentyp:	INT
	Wertebereich:	0 (erster Schutzbereich), 1 (zweiter Schutzbereich)

7.2 Inbetriebnahme

<Name>:	Schutzbereichsname	
	Datentyp:	STRING
1) $M = n * (n - 1) / 2$ mit $n = \$MN_MM_MAXNUM_3D_PROT_AREAS$		

Beispiel

Es sind zwei Schutzbereiche mit den Namen "Rundtisch" und "Werkzeug in Spindel" definiert, die auf Kollision geprüft werden sollen. Die beiden Schutzbereiche sollen gegenseitig auf Kollision überwacht werden. Die Definition dazu erfolgt im siebten Kollisionspaar:

Programmcode	Kommentar
N100 \$NP_COLL_PAIR[6,0] = "Rundtisch"	; 7. Kollisionspaar, ; 1. Schutzbereich
N110 \$NP_COLL_PAIR[6,1] = "Werkzeug in Spindel"	; 7. Kollisionspaar, ; 2. Schutzbereich

Randbedingungen

- Bei Bildung eines Kollisionspaars ist darauf zu achten, dass jeder Schutzbereich dieses Paares mindestens ein Schutzbereichselement besitzt, welches mit \$NP_USAGE (Seite 264) = "C" oder "A" gekennzeichnet ist. Ansonsten kann mit dem Schutzbereich keine Kollisions- oder Abstandsberechnung (Seite 307) durchgeführt werden.
- Die Schutzbereiche eines Kollisionspaars werden nur auf Kollision geprüft, wenn beide Schutzbereiche im Zustand "SB überwacht" sind. Siehe Kapitel "Zustandsdiagramm: Schutzbereich (Seite 288)".

7.2.3.3 \$NP_SAFETY_DIST

Funktion

In die Systemvariable wird der kollisionspaarspezifische Sicherheitsabstand eingetragen. Die Kollisionsvermeidung stellt sicher, dass dieser Sicherheitsabstand nicht unterschritten wird.

Ist in der Systemvariablen ein Wert ungleich 0.0 eingetragen, wird für dieses Kollisionspaar der allgemeine Sicherheitsabstand aus MD10622 \$MN_COLLISION_SAFETY_DIST (Seite 296) nicht beachtet.

Ist in der Systemvariablen der Wert 0.0 eingetragen, wirkt der im Maschinendatum eingestellte Sicherheitsabstand.

Syntax

`$NP_SAFETY_DIST[<m>] = <Wert>`

Bedeutung

\$NP_SAFETY_DIST:	Sicherheitsabstand des Kollisionspaars	
	Datentyp:	REAL
	Defaultwert:	0.0
<m>:	Systemvariablen- bzw. Schutzbereichindex	
	Datentyp:	INT
	Wertebereich:	0, 1, 2, ... (M - 1) ¹⁾
<Wert>:	Sicherheitsabstand	
	Datentyp:	REAL
	Wertebereich:	$0.0 \leq x \leq + \text{max. REAL-Wert}$
	Einheit:	mm oder Inch abhängig von der aktuellen Einstellung für Maßangaben
<p>1) $M = n * (n - 1) / 2$ mit $n = \\$MN_MM_MAXNUM_3D_PROT_AREAS$</p>		

Beispiel

Der Sicherheitsabstand für die Schutzbereiche des siebten Kollisionspaars soll 1.0 mm (Eingabesystem: metrisch) betragen.

Programmcode	Kommentar
N100 \$NP_SAFETY_DIST[6] = 1.0	; 7. Kollisionpaar, ; Sicherheitsabstand=1.0

Siehe auch

Reaktion der Steuerung bei Kollisionsgefahr (Seite 285)

7.2.4 Erweiternde Systemvariablen

7.2.4.1 Übersicht

Über folgende Systemvariablen können weitere Informationen zu internen Zuständen und Werten der Kollisionsvermeidung gelesen werden:

- Zustandsdaten (Seite 302)
- Speicherplatzbedarf (Seite 303)
- Bremswegschätzungen (Seite 303)

7.2.4.2 Zustandsdaten

Über die nachfolgenden Systemvariablen (BTSS-Variablen) können Zustandsdaten der Kollisionsvermeidung gelesen werden

Systemvariable	BTSS-Variable	Bedeutung
\$AN_COLL_STATE[<m>]	anCollState[<m>]	Aktueller Zustand eines Schutzbereiches (aktiv / inaktiv) bezüglich der Kollisionsvermeidung
\$AN_COLL_STATE_COND[<m>]	anCollStateCond[<m>]	Überwachungszustand (bitcodiert) eines Schutzbereiches
\$AN_COLL_IPO_ACTIVE	anCollIpoActive	Aktivierungszustand der Kollisionsvermeidung im Hauptlauf (aktiv / inaktiv)
\$AN_COLL_IPO_LIMIT	anCollIpoLimit	Geschwindigkeitsreduzierung durch Kollisionsvermeidung im Hauptlauf (aktiv / inaktiv)
\$AN_COLL_LOAD[<i>] ¹⁾	anCollLoad[<i>] ¹⁾	Rechenzeitbedarf für Kollisionsvermeidungsfunktion <i>
\$AN_ACTIVATE_COLL_CHECK[<j>]	anActivateColl-Check[<j>]	Aktueller Zustand der NC/PLC-Nahtstelle mit Index <j> für jeweils 8 Byte: DB10 DBX234.0 - DBX.241.7 (Schutzbereiche aktivieren)
\$AN_COLL_CHECK_OFF	anCollCheckOff	Aktueller Zustand der NC/PLC-Nahtstelle: DB10 DBX58.0 - 7 (Schutzbereichsgruppe deaktivieren)
\$AA_COLLPOS[<a>]	aaCollPos	Position der Achse <a> im Maschinenkoordinatensystem (MKS) beim zuletzt aufgetretenen Kollisionsalarm
\$AC_COLLPOS[<k>]	acCollPos	Vektor <k> zur Kollisionsposition im Weltkoordinatensystem beim zuletzt aufgetretenen Kollisionsalarm
a: Achsname i: 0 = Funktion 1, 1 = Funktion 2, 2 = Funktion 3, ... j: Index 0, 1, 2, ... für je ein Bitfeld von 8 Byte Breite. k: Koordinatenindex k = 1, 2, 3 für X-, Y-, Z-Koordinate m: Systemvariablen- bzw. Schutzbereichsindex 0, 1, 2, ... (MD18890 \$MN_MM_MAXNUM_3D_PROT_AREAS - 1)		
1) Die Systemvariable kann durch Beschreiben mit dem Wert 0 zurückgesetzt werden. Jeder andere Wert wird mit einer Fehlermeldung abgelehnt.		

Literatur

- Eine ausführliche Beschreibung der Systemvariablen findet sich in: Listenhandbuch Systemvariable
- Eine ausführliche Beschreibung der Nahtstellensignale findet sich in: Listenhandbuch Nahtstellensignale

7.2.4.3 Speicherplatzbedarf

Über die nachfolgenden Systemvariablen (BTSS-Variablen) können Daten bezüglich des Speicherplatzbedarfs der Kollisionsvermeidung gelesen werden:

Systemvariable	BTSS-Variable	Bedeutung
\$AN_COLL_MEM_AVAILABLE	anCollMemAvailable	Größe des von der Kollisionsvermeidung reservierten Speicherplatzes in kByte.
\$AN_COLL_MEM_USE_MIN ¹⁾	anCollMemUseMin ¹⁾	Minimalwert des von der Kollisionsvermeidung genutzten Speicherplatzes in Prozent des reservierten Speicherplatzes.
\$AN_COLL_MEM_USE_MAX ¹⁾	anCollMemUseMax ¹⁾	Maximalwert des von der Kollisionsvermeidung genutzten Speicherplatzes in Prozent des reservierten Speicherplatzes.
\$AN_COLL_MEM_USE_ACT ¹⁾	anCollMemUseAct ¹⁾	Aktueller Wert des von der Kollisionsvermeidung genutzten Speicherplatzes in Prozent des reservierten Speicherplatzes.
1) Die Systemvariable kann durch Beschreiben mit dem Wert 0 zurückgesetzt werden. Jeder andere Wert wird mit einer Fehlermeldung abgelehnt.		

Literatur

Eine ausführliche Beschreibung der Systemvariablen findet sich in:

Listenhandbuch Systemvariable

7.2.4.4 Bremswegschätzungen

Über die nachfolgenden Systemvariablen (BTSS-Variablen) kann für eine Achse der geschätzte Gesamtbremsweg (linear genähert) und die anteiligen Bremswege von überlagerten Bewegungen gelesen werden. Die Schätzung berücksichtigt dabei nur den momentanen Zustand der Achse. Sie liefert z.B. den Bremsweg 0.0 für eine Achse, die sich als Teil einer Kreisbahn gerade am Umkehrpunkt befindet.

Hinweis

Die Systemvariablen dienen zur Unterstützung bei der Entwicklung von anwenderspezifischen Funktionen im Rahmen der Kollisionsvermeidung und ähnlichen Funktionen.

Tabelle 7-1 Basiskoordinatensystem (BKS)

Systemvariable	BTSS-Variable	Bedeutung
Gesamtbremsweg		
\$AA_DTBREB[<a>]	aaDtbreb	Geschätzter, linear genäherter Gesamtbremsweg
Anteilige Bremswege bei überlagerten Bewegungen		
\$AA_DTBREB_CMD[<a>]	aaDtbrebCmd	Kommandoanteil
\$AA_DTBREB_CORR[<a>]	aaDtbrebCorr	Korrekturanteil
\$AA_DTBREB_DEP[<a>]	aaDtbrebDep	Kopplungsanteil
<a>: Achsname		

Tabelle 7-2 Maschinenkoordinatensystem (MKS)

Systemvariable	BTSS-Variable	Bedeutung
Gesamtbremsweg		
\$AA_DTBREM[<a>]	aaDtbrem	Geschätzter, linear genäherter Gesamtbremsweg
Anteilige Bremswege bei überlagerten Bewegungen		
\$AA_DTBREM_CMD[<a>]	aaDtbremCmd	Kommandoanteil
\$AA_DTBREM_CORR[<a>]	aaDtbremCorr	Korrekturanteil
\$AA_DTBREM_DEP[<a>]	aaDtbremDep	Kopplungsanteil
<a>: Achsname		

Literatur

Eine ausführliche Beschreibung der Systemvariablen findet sich in:

Listenhandbuch Systemvariable

7.3 Programmierung

7.3.1 Prüfen auf Kollisionspaar (COLLPAIR)

Die Funktion `COLLPAIR(. . .)` ermittelt, ob zwei Schutzbereiche ein Kollisionspaar bilden.

Syntax

```
[<RetVal> =] COLLPAIR(<Name_1>,<Name_2>[,<NoAlarm>])
```

Bedeutung

COLLPAIR:	Prüfen auf Zugehörigkeit zu einem Kollisionspaar			
<RetVal>:	Rückgabewert der Funktion			
	Datentyp:	INT		
	Wert:	≥ 0	Die beiden Schutzbereiche bilden ein Kollisionspaar. Rückgabewert == Kollisionspaarindex m (siehe \$NP_COLL_PAIR (Seite 299))	
		-1	Es wurden entweder weniger als zwei Strings angegeben, oder mindestens einer der beiden ist der Null-String.	
		-2	Der im ersten Parameter angegebene Schutzbereich wurde nicht gefunden.	
		-3	Der im zweiten Parameter angegebene Schutzbereich wurde nicht gefunden.	
		-4	Keiner der beiden angegebenen Schutzbereiche wurde gefunden.	
-5		Beide angegebenen Schutzbereiche wurden gefunden, aber nicht gemeinsam in einem Kollisionspaar.		

<Name_1>:	Name des ersten Schutzbereichs			
	Datentyp:	STRING		
	Wertebereich:	Parametrierte Schutzbereichsnamen		
<Name_2>:	Name des zweiten Schutzbereichs			
	Datentyp:	STRING		
	Wertebereich:	Parametrierte Schutzbereichsnamen		
<NoAlarm>:	Alarmunterdrückung (optional)			
	Datentyp:	BOOL		
	Wert:	FALSE (Default)	Im Fehlerfall (<RetVal> < 0) wird die Programmabarbeitung angehalten und ein Alarm angezeigt.	
		TRUE	Im Fehlerfall wird die Programmabarbeitung nicht angehalten und es wird kein Alarm angezeigt. Anwendungsfall: Anwenderspezifische Reaktion entsprechend Rückgabewert	

Siehe auch

Zustandsdiagramm: Schutzbereich (Seite 288)

7.3.2 Neuberechnung des Maschinenmodells der Kollisionsvermeidung anfordern (PROTA)

Werden Systemvariable der kinematischen Kette \$NK_..., der geometrischen Maschinenmodellierung oder der Kollisionsvermeidung \$NP_... im Teileprogramm geschrieben, muss anschließend die Prozedur `PROTA` aufgerufen werden, damit die Änderung im NC-internen Maschinenmodell der Kollisionsvermeidung wirksam wird.

Syntax

```
PROTA[ (<Par> ) ]
```

Bedeutung

PROTA:	Neuberechnung des Maschinenmodells der Kollisionsvermeidung anfordern	
	<ul style="list-style-type: none"> • Löst Vorlaufstopp aus. • Muss alleine im Satz stehen. 	

<Par>:	Parameter (optional)		
	Datentyp:	STRING	
	Wert:	---	Ohne Parameter. Es wird eine Neuberechnung des Maschinenmodells durchgeführt. Die Stati der Schutzbereiche bleiben erhalten.
"R"		Es wird eine Neuberechnung des Maschinenmodells durchgeführt. Die Schutzbereiche werden in ihren Initialisierungsstatus entsprechend \$NP_INIT_STAT (Seite 254) versetzt.	

Randbedingungen

Simulation

Die Prozedur `PROTA` darf in Teileprogrammen nicht im Zusammenhang mit der Simulation (`simNC`) verwendet werden.

Beispiel: Vermeidung des Aufrufs von `PROTA`, während die Simulation aktiv ist.

Programmcode	Kommentar
...	
<code>IF \$P_SIM == FALSE</code>	<code>; IF Simulation nicht aktiv</code>
<code>PROTA</code>	<code>; THEN Kollisionsmodell neu berechnen</code>
<code>ENDIF</code>	<code>; ENDIF</code>
...	

Siehe auch

Schutzbereichszustand setzen (`PROTS`) (Seite 306)

7.3.3 Schutzbereichszustand setzen (`PROTS`)

Die Prozedur `PROTS (. . .)` setzt den Status von Schutzbereichen auf den angegebenen Wert.

Syntax

`PROTS (<State>[, <Name_1>, . . . , <Name_n>])`

Bedeutung

PROTS:	Status von Schutzbereichen setzen
	<ul style="list-style-type: none"> Muss alleine im Satz stehen.

<State>:	Status, auf den die angegebenen Schutzbereiche gesetzt werden sollen		
	Datentyp:	CHAR	
	Wert:	"A"oder "a"	Status: Aktiv
		"I"oder "i"	Status: Inaktiv
		"P"oder "p"	Status: Voraktiviert bzw. PLC-gesteuert ¹⁾
"R"oder "r"		Status: NC-interner Wert des Initialisierungsstatus ²⁾	
<Name_1> . . . <Name_n>:	Name eines oder mehrerer Schutzbereiche, die auf den angegebenen Status gesetzt werden sollen (optional) Ist kein Name angegeben, wird der angegebenen Status für alle definierten Schutzbereiche gesetzt.		
	Datentyp:	STRING	
	Wertebereich:	Parametrierte Schutzbereichsnamen	
	Hinweis Die maximale Anzahl von Schutzbereichen, die als Parameter angegeben werden können, ist nur abhängig von der maximal möglichen Anzahl von Zeichen pro Programmzeile.		
¹⁾ Die Aktivierung / Deaktivierung erfolgt über: DB10.DBX234.0 - DBX241.7 ²⁾ Der Status wird auf den NC-internen Wert des Initialisierungsstatus gesetzt, d. h. auf den Wert, den die Systemvariable \$NP_INIT_STAT (Seite 254) zum Zeitpunkt des letzten Aufrufs von PROTA(...) (Seite 305) hatte.			

7.3.4

Abstandsbestimmung zweier Schutzbereiche (PROTD)

Die Funktion `PROTD (. . .)` berechnet den Abstand von zwei Schutzbereichen.

Funktionseigenschaften:

- Die Abstandsberechnung erfolgt unabhängig vom Status der Schutzbereiche (aktiviert, deaktiviert, voraktiviert).
- Zur Abstandsberechnung zweier Schutzbereiche werden nur solche Schutzbereichselemente herangezogen, welche mit `$NP_USAGE` (Seite 264) = "C" oder "A" gekennzeichnet sind. Schutzbereichselemente des Schutzbereichs, die mit `$NP_USAGE` = "V" gekennzeichnet sind, werden nicht betrachtet.
- Schutzbereiche, bei denen alle Schutzbereichselemente des Schutzbereichs mit `$NP_USAGE` = "V" gekennzeichnet sind, können nicht zur Abstandsberechnung herangezogen werden.

- Die Abstandsberechnung erfolgt mit den am Ende des Vorgängersatzes gültigen Positionen.
- Überlagerungen, die im Hauptlauf eingerechnet werden, z. B. DRF-Verschiebung oder externe Nullpunktverschiebungen, gehen mit den zum **Interpretationszeitpunkt** der Funktion gültigen Werten in die Abstandsberechnung ein.

Hinweis

Synchronisation

Bei Anwendung der Funktion `PROTD (. . .)` liegt es ausschließlich in der Verantwortung des Anwenders, Haupt- und Vorlauf gegebenenfalls mittels Vorlaufstopp `STOPRE` zu synchronisieren.

Kollision

Liegt zwischen den angegebenen Schutzbereichen eine Kollision vor, liefert die Funktion einen Abstand von 0,0. Eine Kollision liegt vor, wenn sich die beiden Schutzbereiche berühren oder durchdringen.

Der Sicherheitsabstand für die Kollisionsprüfung (MD10622 `$MN_COLLISION_SAFETY_DIST`) wird bei der Abstandsberechnung nicht berücksichtigt.

Syntax

```
[<RetVal> =] PROTD ([<Name_1>], [<Name_2>], VAR <Vector>[, <System>])
```

Bedeutung

PROTD:	Abstand der beiden angegebenen Schutzbereiche berechnen	
	<ul style="list-style-type: none"> • Muss alleine im Satz stehen. 	
<RetVal>:	Rückgabewert der Funktion: Absolutwert des Abstandes der beiden Schutzbereiche oder 0,0 bei Kollision (siehe oben: Absatz Kollision)	
	Datentyp:	REAL
	Wertebereich:	$0,0 \leq x \leq +\text{max. REAL-Wert}$
<Name_1>, <Name_2>:	Namen der beiden Schutzbereiche, deren Abstand zueinander berechnet werden soll (optional)	
	Datentyp:	STRING
	Wertebereich:	Parametrierte Schutzbereichnamen
	Defaultwert:	"" (Leerstring) Sind keine Schutzbereiche angegeben, berechnet die Funktion den aktuell kleinsten Abstand aus allen im Kollisionsmodell enthaltenen aktivierten und voraktivierten Schutzbereichen.

<Vector>:	Rückgabewert: 3-dimensionaler Abstandsvektor von Schutzbereich <Name_2> zu Schutzbereich <Name_1> mit:	
	<ul style="list-style-type: none"> • <Vector>[0]: X-Koordinate im Weltkoordinatensystem • <Vector>[1]: Y-Koordinate im Weltkoordinatensystem • <Vector>[2]: Z-Koordinate im Weltkoordinatensystem 	
	Bei Kollision: <Vector> == Nullvektor	
Datentyp:	VAR REAL [3]	
Wertebereich:	<Vector> [n]: $0,0 \leq x \leq \pm \text{max. REAL-Wert}$	
<System>:	Maßsystem (inch / metrisch) für Abstand und Abstandsvektor (optional)	
	Datentyp:	BOOL
	Wert:	FALSE (Default)
TRUE		Maßsystem entsprechend des eingestellten Grundsystems: MD52806 \$MN_ISO_SCALING_SYSTEM

7.4 Beispiel

7.4.1 Vorgaben

Allgemeines

Anhand einer vereinfachten 3-Achs-Fräsmaschine wird beispielhaft das prinzipielle Vorgehen zur Parametrierung der Kollisionsvermeidung über ein Teileprogramm gezeigt. Im Teileprogramm werden alle für die Kollisionsvermeidung relevanten Systemvariablen geschrieben:

- Kinematische Kette \$NK_...
- Geometrische Maschinenmodellierung \$NP_...
- Kollisionsvermeidung \$NP_...

Im Teileprogramm wird abschließend das Maschinenmodell aktiviert, so dass nach Ausführung des Teileprogramms, die Kollisionsvermeidung für die 3-Achs-Fräsmaschine vollständig parametriert und aktiv ist.

Options- und Maschinendaten

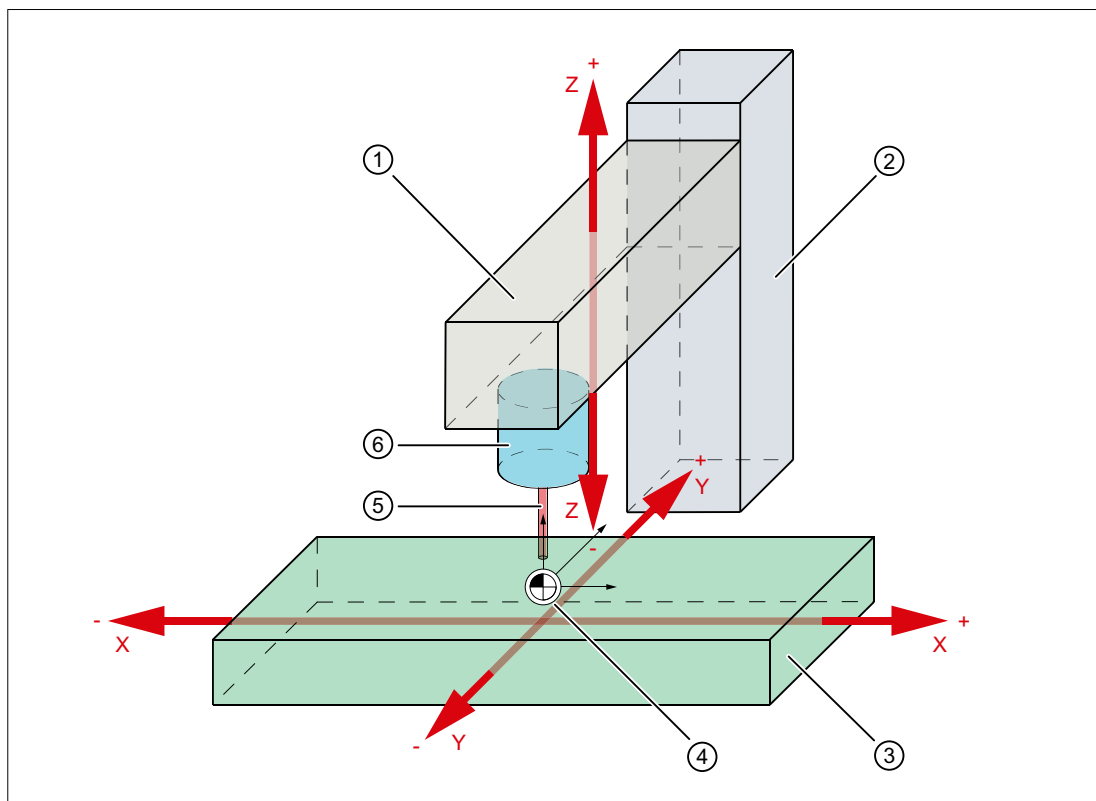
Für das Beispiel sind folgende Options- und Maschinendaten einzustellen:

Nr.	Optionsdatum: \$ON_	Wert
MD19830	COLLISION_MASK	1

Nr.	Maschinendatum: \$MN_	Wert
MD10619	COLLISION_TOLERANCE	1
MD18880	MM_MAXNUM_KIN_CHAIN_ELEM	10
MD18890	MM_MAXNUM_3D_PROT_AREAS	10
MD18892	MM_MAXNUM_3D_PROT_AREA_ELEM	10
MD18893	MM_MAXNUM_3D_T_PROT_ELEM	1
MD18894	MM_MAXNUM_3D_FACETS_INTERN	1000
MD18895	MM_MAXNUM_3D_FACETS	3000
MD18896	MM_MAXNUM_3D_COLLISION	0
MD18897	MM_MAXNUM_3D_INTERFACE_IN	16
MD18899	PROT_AREA_TOOL_MASK	1

Prinzipieller Aufbau der 3-Achs-Fräsmaschine

Das nachfolgende Bild zeigt den prinzipiellen Aufbau der Maschine.



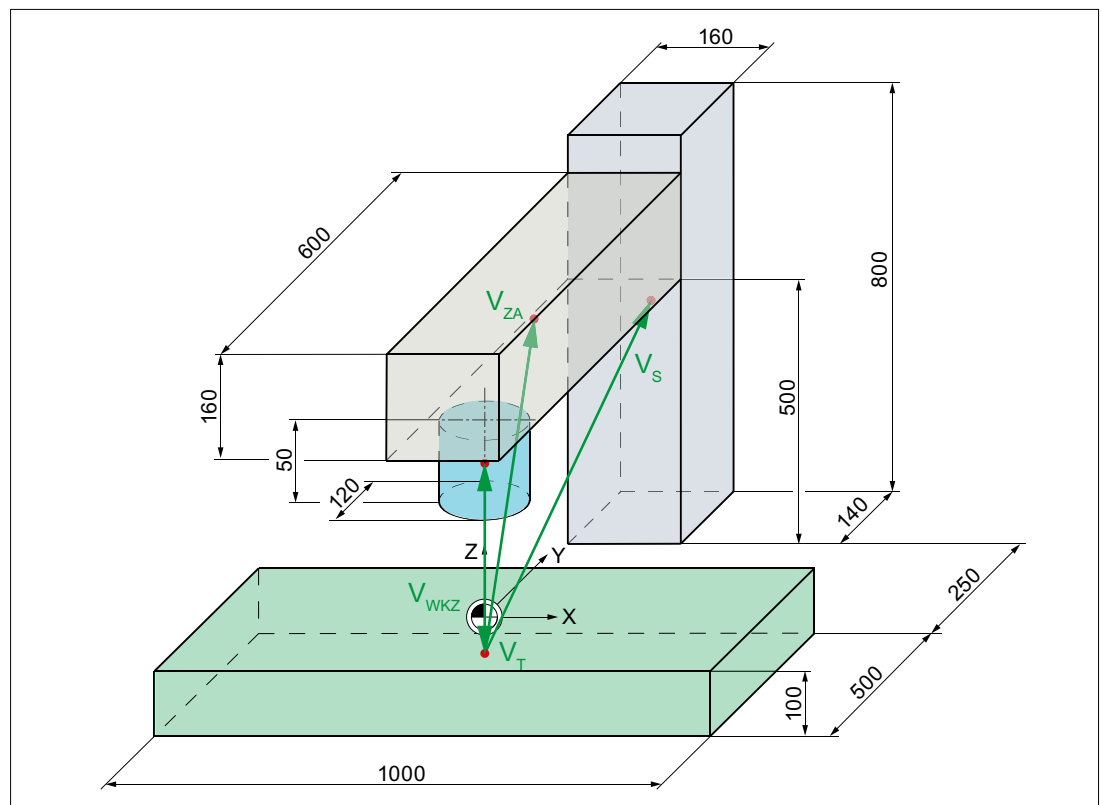
- ① Z-Achse
- ② Ständer
- ③ Tisch
- ④ Maschinennullpunkt = Referenzpunkt
- ⑤ Werkzeug
- ⑥ Werkzeugaufnahme

Den Maschinenteile bzw. Schutzbereichen sind folgende Maschinenachsen zugeordnet .

Maschinenteile bzw. Schutzbereiche	Maschinenachse
Tisch	X1, Y1
Z-Achse	Z1
Ständer	---
Werkzeugaufnahme	Z1
Werkzeug	Z1

Maßzeichnung

In der nachfolgende Maßzeichnung sind die Abmessungen der Schutzbereichselemente sowie deren Lage (Vektoren zum Mittelpunkt des Schutzbereichselements) bezogen auf den Maschinennullpunkt angegeben.



Vektoren zum Mittelpunkt der Schutzbereichselemente

v_{WKZ}	Werkzeugaufnahme (0;0;25)
v_{ZA}	Z-Achse (0;200;130)
v_S	Ständer (0;570;350)
v_T	Tisch (0;0;-50)

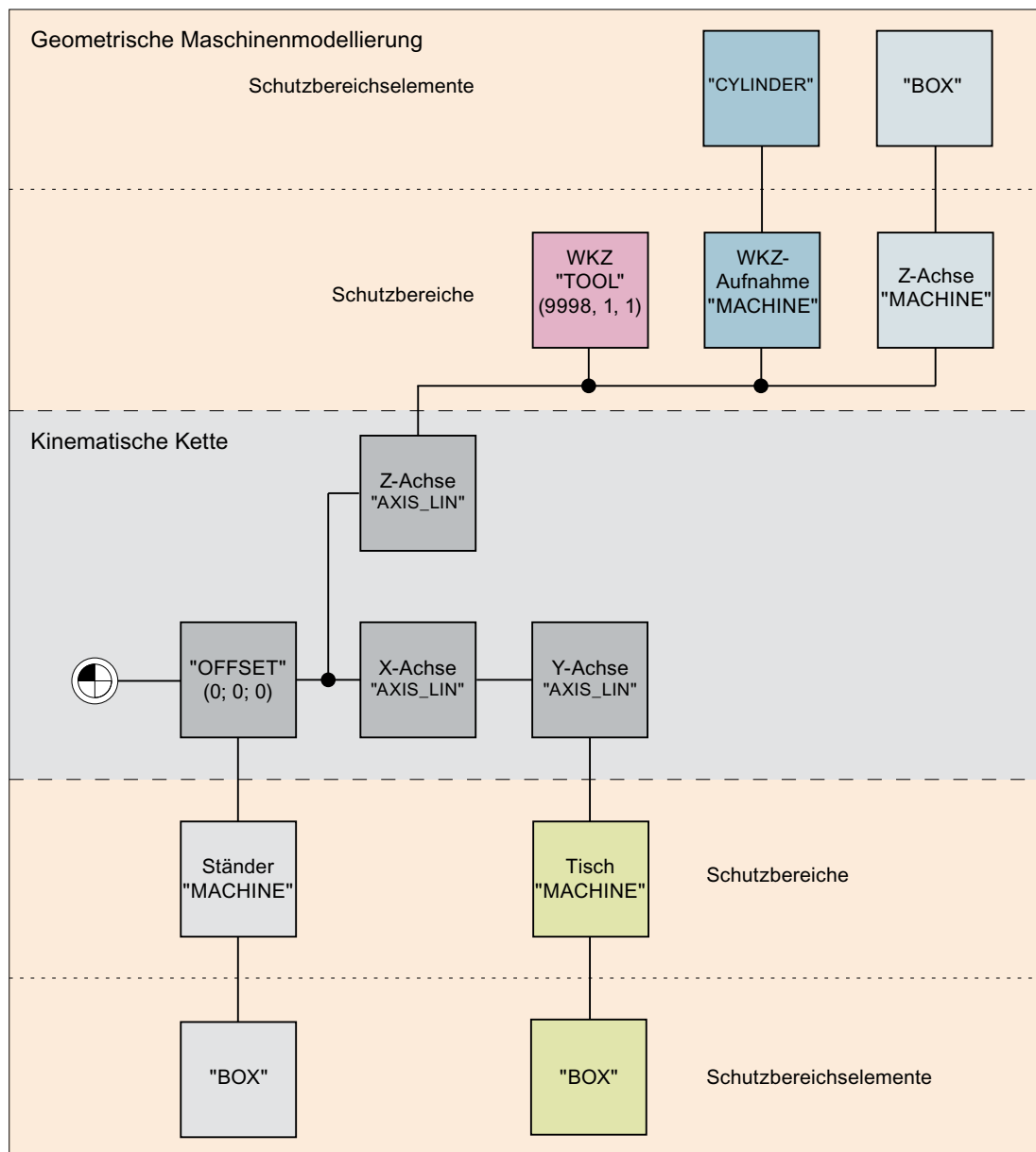
Kinematische Kette

Die kinematische Kette (siehe nächstes Bild) beginnt mit einem Element vom Typ "Offset". Diesem werden alle statischen Schutzbereiche der Maschine zugeordnet. Im Beispiel ist dies nur der Schutzbereich "Staender".

Auf das Offset-Element folgen die kinematischen Elemente der Maschinenachsen:

- Z-Achse und X-Achse verfahren unabhängig voneinander ⇒ \$NK_PARALLEL
- Die Y-Achse verfährt abhängig von der X-Achse ⇒ \$NK_NEXT

Den kinematischen Elementen der Maschinenachsen sind die verschiedenen Schutzbereiche der geometrischen Maschinenmodellierung zugeordnet



Kollisionspaare

Für das Beispiel wird angenommen, dass nur folgende Kollisionspaare zu berücksichtigen sind:

- Werkzeugaufnahme - Tisch
- Werkzeug - Tisch

7.4.2 Teileprogramm des Maschinenmodells

Programmcode

```

;*****
;***** Beispiel *****
; Fraesmaschine: 3 Linearachsen, 1 Spindel
; Tisch => X1, Y1
; Z-Achse, Werkzeugaufnahme, Werkzeug => Z1
;*****
; Stand: 11.02.2013, 15:34
;
;=====
; Verwendete Kollisionsmaschinenendaten
;=====
; MD10619 $MN_COLLISION_TOLERANCE = 1
;
; MD18880 $MN_MM_MAXNUM_KIN_CHAIN_ELEM = 100
; MD18890 $MN_MM_MAXNUM_3D_PROT_AREAS = 10
; MD18892 $MN_MM_MAXNUM_3D_PROT_AREA_ELEM = 10
; MD18893 $MN_MM_MAXNUM_3D_T_PROT_ELEM = 100
; MD18894 $MN_MM_MAXNUM_3D_FACETS_INTERN = 1000
; MD18895 $MN_MM_MAXNUM_3D_FACETS = 3000
; MD18896 $MN_MM_MAXNUM_3D_COLLISION = 0
; MD18897 $MN_MM_MAXNUM_3D_INTERFACE_IN = 16
; MD18899 $MN_PROT_AREA_TOOL_MASK = 1
;
; MD19830 $ON_COLLISION_MASK = 1 ; Option
;
;
;=====
; Definitionen
;=====
DEF INT RETVAL = 0 ; Rueckgabewert der Loeschfunktion
;
DEF INT C_NKE = 0 ; Index für kinematische Elemente
DEF INT C_NPC = 0 ; Index für Schutzbereiche
DEF INT C_NPE = 0 ; Index für Schutzbereichselemente
DEF INT C_NPP = 0 ; Index für Kollisionspaare
;
;

```

Programmcode

```

;=====
; Initialisierung der Kollisionsdaten
;=====
MSG("Schutzbereiche")
G4 F3
; Alle Parameter auf ihre Grundstellung zuruecksetzen
;
RETVAl = DELOBJ("KIN_CHAIN_ELEM")
IF (RETVAl <> 0)
    MSG("Fehler: DELOBJ KIN_CHAIN_ELEM")
    G4 F5
ENDIF
;
RETVAl = DELOBJ("PROT_AREA_ALL")
IF RETVAl <> 0
    MSG("Fehler: DELOBJ PROT_AREA_ALL")
    G4 F5
ENDIF
;
RETVAl = DELOBJ("PROT_AREA_COLL_PAIRS")
IF RETVAl <> 0
    MSG("Fehler: DELOBJ PROT_AREA_COLL_PAIRS")
    G4 F5
ENDIF
;
;
;=====
; Kinematische Kette
;=====
; KE1: ROOT
; -----
$NK_NAME[C_NKE]      = "ROOT"
$NK_NEXT[C_NKE]     = "X-Achse"
$NK_PARALLEL[C_NKE] = ""
$NK_TYPE[C_NKE]     = "OFFSET"
;
$NK_OFF_DIR[C_NKE, 0] = 0.0      ; X
$NK_OFF_DIR[C_NKE, 1] = 0.0      ; Y
$NK_OFF_DIR[C_NKE, 2] = 0.0      ; Z
;
$NK_AXIS[C_NKE]      = ""
$NK_A_OFF[C_NKE]     = 0.0
;
C_NKE = C_NKE + 1      ; naechstes kinematisches Element
;

```

Programmcode

```

; -----
; Kinematisches Element: X-Achse
; -----
$NK_NAME[C_NKE]      = "X-Achse"
$NK_NEXT[C_NKE]     = "Y-Achse"
$NK_PARALLEL[C_NKE] = "Z-Achse"
$NK_TYPE[C_NKE]     = "AXIS_LIN"
;
$NK_OFF_DIR[C_NKE, 0] = 1.0      ; X
$NK_OFF_DIR[C_NKE, 1] = 0.0      ; Y
$NK_OFF_DIR[C_NKE, 2] = 0.0      ; Z
;
$NK_AXIS[C_NKE]      = "X1"
$NK_A_OFF[C_NKE]     = 0.0
;
C_NKE = C_NKE + 1      ; naechstes kinematisches Element
;
; -----
; Kinematisches Element: Y-Achse
; -----
$NK_NAME[C_NKE]      = "Y-Achse"
$NK_NEXT[C_NKE]     = ""
$NK_PARALLEL[C_NKE] = ""
$NK_TYPE[C_NKE]     = "AXIS_LIN"
;
$NK_OFF_DIR[C_NKE, 0] = 0.0      ; X
$NK_OFF_DIR[C_NKE, 1] = 1.0      ; Y
$NK_OFF_DIR[C_NKE, 2] = 0.0      ; Z
;
$NK_AXIS[C_NKE]      = "Y1"
$NK_A_OFF[C_NKE]     = 0.0
;
C_NKE = C_NKE + 1      ; naechstes kinematisches Element
;
; -----
; Kinematisches Element: Z-Achse
; -----
$NK_NAME[C_NKE]      = "Z-Achse"
$NK_NEXT[C_NKE]     = ""
$NK_PARALLEL[C_NKE] = ""
$NK_TYPE[C_NKE]     = "AXIS_LIN"
;
$NK_OFF_DIR[C_NKE, 0] = 0.0      ; X
$NK_OFF_DIR[C_NKE, 1] = 0.0      ; Y
$NK_OFF_DIR[C_NKE, 2] = 1.0      ; Z
;
$NK_AXIS[C_NKE]      = "Z1"
$NK_A_OFF[C_NKE]     = 0.0
;
C_NKE = C_NKE + 1      ; naechstes kinematisches Element
;
;

```

Programmcode

```

;=====
; Schutzbereiche mit Schutzbereichselementen
;=====
; Schutzbereich 1: Staender
; -----
$NP_PROT_NAME[C_NPC] = "Staender"
$NP_PROT_TYPE[C_NPC] = "MACHINE"
$NP_CHAIN_ELEM[C_NPC] = "ROOT"
$NP_1ST_PROT[C_NPC] = "SBE-Staender"
$NP_PROT_COLOR[C_NPC] = 'HFFA0A0A4' ; AARRGGBB
$NP_BIT_NO[C_NPC] = -1
$NP_INIT_STAT[C_NPC] = "A"
;
C_NPC = C_NPC + 1 ; naechster Schutzbereich
;
; -----
; Schutzbereichselement 1.1: SBE Staender
; -----
$NP_NAME[C_NPE] = "SBE-Staender"
$NP_NEXT[C_NPE] = ""
$NP_NEXTP[C_NPE] = ""
$NP_TYPE[C_NPE] = "BOX"
;
$NP_PARA[C_NPE,0] = 160.0 ; Laenge
$NP_PARA[C_NPE,1] = 140.0 ; Breite
$NP_PARA[C_NPE,2] = 800.0 ; Hoehe
;
$NP_OFF[C_NPE,0] = 0.0 ; Mittelpunkt
$NP_OFF[C_NPE,1] = 570.0 ; X
$NP_OFF[C_NPE,2] = 350.0 ; Y: xxx hinter Tischkante
; Z: Unterkante gleich Unterkante Tisch
;
$NP_DIR[C_NPE,0] = 0.0
$NP_DIR[C_NPE,1] = 0.0
$NP_DIR[C_NPE,2] = 0.0
;
$NP_ANG[C_NPE] = 0.0
;
$NP_COLOR[C_NPE] = 0
$NP_D_LEVEL[C_NPE] = 0
$NP_USAGE[C_NPE] = "V" ; V = nur visualisieren
$NP_FILENAME[C_NPE] = ""
;
C_NPE = C_NPE + 1 ; naechstes Schutzbereichselement
;

```


Programmcode

```

;+++++
; Schutzbereich 2: Werkzeugaufnahme
; -----
$NP_PROT_NAME[C_NPC] = "WKZ-Aufnahme"
$NP_PROT_TYPE[C_NPC] = "MACHINE"
$NP_CHAIN_ELEM[C_NPC] = "Z-Achse"
$NP_1ST_PROT[C_NPC] = "SBE-WKZ-Aufnahme"
$NP_PROT_COLOR[C_NPC] = 'HFF0000FF' ; AARRGGBB
$NP_BIT_NO[C_NPC] = -1
$NP_INIT_STAT[C_NPC] = "A"
;
C_NPC = C_NPC + 1 ; naechster Schutzbereich
;
; -----
; Schutzbereichselement 2.1: SBE-WKZ-Aufnahme
; -----
$NP_NAME[C_NPE] = "SBE-WKZ-Aufnahme"
$NP_NEXT[C_NPE] = ""
$NP_NEXTP[C_NPE] = ""
$NP_TYPE[C_NPE] = "CYLINDER"
;
; Zylindermasse
$NP_PARA[C_NPE,0] = 50.0 ; Hoehe
$NP_PARA[C_NPE,1] = 60.0 ; Radius
$NP_PARA[C_NPE,2] = 0.0
;
; Mittelpunkt
$NP_OFF[C_NPE,0] = 0.0 ; X
$NP_OFF[C_NPE,1] = 0.0 ; Y
$NP_OFF[C_NPE,2] = 25.0 ; Z: Halbe Hoehe
;
$NP_DIR[C_NPE,0] = 0.0
$NP_DIR[C_NPE,1] = 0.0
$NP_DIR[C_NPE,2] = 0.0
;
$NP_ANG[C_NPE] = 0.0
;
$NP_COLOR[C_NPE] = 0
$NP_D_LEVEL[C_NPE] = 0
$NP_USAGE[C_NPE] = "A"
$NP_FILENAME[C_NPE] = ""
;
C_NPE = C_NPE + 1 ; naechstes Schutzbereichselement
;

```

7.4 Beispiel

Programmcode

```
; ++++++
; Schutzbereich 3: Werkzeug
; -----
$NP_PROT_NAME[C_NPC] = "WKZ"
$NP_PROT_TYPE[C_NPC] = "TOOL"
$NP_CHAIN_ELEM[C_NPC] = "Z-Achse"
$NP_1ST_PROT[C_NPC] = ""
$NP_PROT_COLOR[C_NPC] = 'HFFFF0000' ; AARRGGBB
$NP_BIT_NO[C_NPC] = -1
$NP_INIT_STAT[C_NPC] = "A"
;
$NP_INDEX[C_NPC,0] = 1 ; nur relevant bei Typ "TOOL"
$NP_INDEX[C_NPC,1] = 9998 ; WKZ-Platz-Nr. / Spindelnr.
$NP_INDEX[C_NPC,2] = 1 ; Magazin-Nr. / -
; TOA-Bereich
;
C_NPC = C_NPC + 1 ; naechster Schutzbereich
;
```

Programmcode

```

; ++++++
; Schutzbereich 4: Z-Achse
; -----
$NP_PROT_NAME[C_NPC] = "Z-Achse"
$NP_PROT_TYPE[C_NPC] = "MACHINE"
$NP_CHAIN_ELEM[C_NPC] = "Z-Achse"
$NP_1ST_PROT[C_NPC] = "SBE-Z-Achse"
$NP_PROT_COLOR[C_NPC] = 'HFFA0A0A4' ; AARRGGBB
$NP_BIT_NO[C_NPC] = -1
$NP_INIT_STAT[C_NPC] = "A"
;
C_NPC = C_NPC + 1 ; naechster Schutzbereich
;
; -----
; Schutzbereichselement 4.1: SBE-Horizontaler Staender
; -----
$NP_NAME[C_NPE] = "SBE-Z-Achse"
$NP_NEXT[C_NPE] = ""
$NP_NEXTP[C_NPE] = ""

$NP_TYPE[C_NPE] = "BOX"
;
$NP_PARA[C_NPE,0] = 160.0
$NP_PARA[C_NPE,1] = 600.0
$NP_PARA[C_NPE,2] = 160.0
;
$NP_OFF[C_NPE,0] = 0.0
$NP_OFF[C_NPE,1] = 200.0
$NP_OFF[C_NPE,2] = 130.0
;
$NP_DIR[C_NPE,0] = 0.0
$NP_DIR[C_NPE,1] = 0.0
$NP_DIR[C_NPE,2] = 0.0
;
$NP_ANG[C_NPE] = 0.0
;
$NP_COLOR[C_NPE] = 0
$NP_D_LEVEL[C_NPE] = 0
$NP_USAGE[C_NPE] = "A"
$NP_FILENAME[C_NPE] = ""
;
C_NPE = C_NPE + 1 ; naechstes Schutzbereichselement
;

```

Programmcode

```

; ++++++
; Schutzbereich 5: Tisch
; -----
$NP_PROT_NAME[C_NPC] = "Tisch"
$NP_PROT_TYPE[C_NPC] = "MACHINE"
$NP_CHAIN_ELEM[C_NPC] = "Y-Achse"
$NP_1ST_PROT[C_NPC] = "SBE-Tisch"
$NP_PROT_COLOR[C_NPC] = 'HFF00FF00' ; AARRGGBB
$NP_BIT_NO[C_NPC] = -1
$NP_INIT_STAT[C_NPC] = "A"
;
C_NPC = C_NPC + 1 ; naechster Schutzbereich
;
; -----
; Schutzbereichselement 5.1: SBE-Tisch
; -----
$NP_NAME[C_NPE] = "SBE-Tisch"
$NP_NEXT[C_NPE] = ""
$NP_NEXTP[C_NPE] = ""
$NP_TYPE[C_NPE] = "BOX"
;
$NP_PARA[C_NPE,0] = 1000.0
$NP_PARA[C_NPE,1] = 500.0
$NP_PARA[C_NPE,2] = 100.0
;
$NP_OFF[C_NPE,0] = 0.0
$NP_OFF[C_NPE,1] = 0.0
$NP_OFF[C_NPE,2] = -50.0
;
$NP_DIR[C_NPE,0] = 0.0
$NP_DIR[C_NPE,1] = 0.0
$NP_DIR[C_NPE,2] = 0.0
;
$NP_ANG[C_NPE] = 0.0
;
$NP_COLOR[C_NPE] = 0
$NP_D_LEVEL[C_NPE] = 0
$NP_USAGE[C_NPE] = "A"
$NP_FILENAME[C_NPE] = ""
;
C_NPE = C_NPE + 1 ; naechstes Schutzbereichselement
;
;

```

Programmcode

```

;=====
; Kollisionspaare
;=====
$NP_COLL_PAIR[C_NPP, 0] = "WKZ-Aufnahme"
$NP_COLL_PAIR[C_NPP, 1] = "Tisch"
;
C_NPP = C_NPP + 1           ; naechstes Kollisionspaar
;
$NP_COLL_PAIR[C_NPP, 0] = "WKZ"
$NP_COLL_PAIR[C_NPP, 1] = "Tisch"
;
C_NPP = C_NPP + 1           ; naechstes Kollisionspaar
;
;
;=====
; Aktivierung des Maschinenmodells
;=====
PROTA
PROTS("A")
;
M2
;===== ENDE =====

```

7.5 Datenlisten

7.5.1 Maschinendaten

7.5.1.1 NC-spezifische Maschinendaten

Nummer	Bezeichner: \$MN_	Beschreibung
MD10619	COLLISION_TOLERANCE	Kollisionstoleranz
MD10622	COLLISION_SAFETY_DIST	Sicherheitsabstand
MD18896	MM_MAXNUM_3D_COLLISION	Speicherplatz für die Kollisionsvermeidung

7.5.2 Systemvariablen

Bezeichner	Beschreibung
\$NP_COLL_PAIR	Name des ersten oder zweiten Schutzbereichs eines Kollisionspaars
\$NP_SAFETY_DIST	Sicherheitsabstand des Kollisionspaars
\$AN_COLL_STATE	Aktueller Zustand eines Schutzbereiches bezüglich der Kollisionsvermeidung
\$AN_COLL_STATE_COND	Überwachungszustand (bitcodiert) eines Schutzbereichs

7.5 Datenlisten

Bezeichner	Beschreibung
\$AN_COLL_IPO_ACTIVE	Aktivierungszustand der Kollisionsvermeidung im Hauptlauf
\$AN_COLL_IPO_LIMIT	Geschwindigkeitsreduzierung durch Kollisionsvermeidung im Hauptlauf
\$AN_COLL_LOAD	Rechenzeitbedarf für Kollisionsvermeidungsfunktion
\$AN_ACTIVATE_COLL_CHECK	Aktueller Zustand der NC/PLC-Nahtstelle DB10, DBX234.0 - DBX.241.7 (Schutzbereiche aktivieren)
\$AN_COLL_CHECK_OFF	Aktueller Zustand der NC/PLC-Nahtstelle DB10, DBB58 (Schutzbereichsgruppen betriebsartenabhängig ausschalten)
\$AA_COLLPOS	Position einer Achse im MKS beim zuletzt aufgetretenen Kollisionsalarm
\$AC_COLLPOS	Vektor zur Kollisionsposition im Weltkoordinatensystem beim zuletzt aufgetretenen Kollisionsalarm
\$AN_COLL_MEM_AVAILABLE	Größe des von der Kollisionsvermeidung reservierten Speicherplatzes in kByte.
\$AN_COLL_MEM_USE_MIN	Minimalwert des von der Kollisionsvermeidung genutzten Speicherplatzes in Prozent des reservierten Speicherplatzes.
\$AN_COLL_MEM_USE_MAX	Maximalwert des von der Kollisionsvermeidung genutzten Speicherplatzes in Prozent des reservierten Speicherplatzes
\$AN_COLL_MEM_USE_ACT	Aktueller Wert des von der Kollisionsvermeidung genutzten Speicherplatzes in Prozent des reservierten Speicherplatzes.
\$AA_DTBREB	Geschätzter, linear genäherter Gesamtbremsweg (BKS)
\$AA_DTBREB_CMD	Kommandoanteil (BKS)
\$AA_DTBREB_CORR	Korrekturanteil (BKS)
\$AA_DTBREB_DEP	Kopplungsanteil (BKS)
\$AA_DTBREM	Geschätzter, linear genäherter Gesamtbremsweg (MKS)
\$AA_DTBREM_CMD	Kommandoanteil (MKS)
\$AA_DTBREM_CORR	Korrekturanteil (MKS)
\$AA_DTBREM_DEP	Kopplungsanteil (MKS)

7.5.3 Signale

7.5.3.1 Signale an NC

Signalname	SINUMERIK 840D sl	SINUMERIK 828D
Kollisionsvermeidung: Schutzbereichsgruppe deaktivieren (PLC → NC)	DB10.DBX58.0 - 7	DB2600.DBX2.0 - 7
Kollisionsvermeidung: Schutzbereich aktivieren (PLC → NC)	DB10.DBX234.0 - DBX241.7	DB2600.DBX4.0 - DBX11.7

7.5.3.2 Signale von NC

Signalname	SINUMERIK 840D sl	SINUMERIK 828D
Kollisionsvermeidung: Schutzbereich aktiv	DB10.DBX226.0 - DBX233.7	DB2700.DBX20.0 - DBX27.7

7.5.3.3 Signale von Kanal

Signalname	SINUMERIK 840D sl	SINUMERIK 828D
Kollisionsvermeidung: Stopp	DB21,DBX377.0	DB33xx.DBX4005.0

7.5.3.4 Signale von Achse

Signalname	SINUMERIK 840D sl	SINUMERIK 828D
Kollisionsvermeidung: Geschwindigkeitsreduzierung	DB31,DBX77.0	DB39xx.DBX1003.0

K11 Kollisionsvermeidung, extern

8.1 Funktionsbeschreibung

8.1.1 Optionen

Die Funktion "Kollisionsvermeidung, extern " ist eine lizenzpflichtige Option:

- **Kollisionsvermeidung ADVANCED** (Maschine, Werkstück): 6FC5800-0AS04-0YB0

8.1.2 Merkmale

Die Funktion "Kollisionsvermeidung, extern" stellt an der Industrial Ethernet-Schnittstelle X120 der NCU-Baugruppe eine proprietäre Datenschnittstelle zur Verfügung. Über diese Schnittstelle werden folgende Daten in Echtzeit übertragen:

- Soll- und Istwerte der Maschinenachsen zur Überwachung und Visualisierung der Maschinenbewegungen
- Statusdaten und Vorgaben zur Beeinflussung der Verfahrbewegungen
- Signale zur Überwachung der externen Applikation

Basierend auf diesen Daten kann auf einem externen Rechner von einem SINUMERIK-Produktpartner eine eigenständige Applikation zur Maschinenvisualisierung und Kollisionsvermeidung realisiert werden.

8.2 Inbetriebnahme

8.2.1 Maschinendaten

8.2.1.1 Funktionen der externen Kollisionsvermeidung

Mit dem Maschinendatum werden die Funktionen der externen Kollisionsvermeidung freigegeben:

MD16900 \$MN_COLLISION_EXT_FUNKTION_MASK.Bit x = 1

Bit	Bedeutung
0	Aktivierung der UDP-Schnittstelle
1 - 31	reserviert

8.2.1.2 Vorschauzeit

Mit dem Maschinendatum wird die Vorschauzeit eingestellt:

MD16901 \$MN_COLLISION_EXT_PREVIEW_TIME = <Vorschauzeit>

8.2.1.3 Vorschau-Zeitschritt

Mit dem Maschinendatum wird die Größe eines Zeitschritts zur Berechnung der Bewegungsvorschau eingestellt:

MD16902 \$MN_COLLISION_EXT_PREVIEW_STEP = <Vorschauzeit>

8.2.1.4 Timeout-Zeit

Mit dem Maschinendatum wird die Zeit eingestellt, innerhalb der sich die externe Applikation im Rahmen der Lebenszeichenüberwachung spätestens bei der Steuerung zurückgemeldet haben muss:

MD16903 \$MN_COLLISION_EXT_TIMEOUT = <Timeout-Zeit>

Meldet sich die externe Applikation nicht in der hier eingestellten Zeit zurück, wird der Alarm 26300 "Lebenszeichenausfall der externen Kollisionsvermeidung" angezeigt.

K12 Transformationsdefinitionen mit kinematischen Ketten

9

9.1 Funktionsbeschreibung

9.1.1 Merkmale

Im vorliegenden Kapitel wird beschrieben, wie Transformationen mittels einer kinematischen Kette abgebildet und in der Steuerung über Systemvariablen parametrisiert werden. Die Systemvariablen werden in der NC remanent gespeichert und können über SINUMERIK Operate mittels Inbetriebnahme-archiv als "NC-Daten" archiviert bzw. eingelesen werden.

Transformationen über kinematische Ketten

Die für kinematische Transformationen relevante Maschinenkinematik wird durch zwei kinematische Ketten beschrieben:

- Eine Kette zeigt vom Maschinennullpunkt (Nullpunkt des Weltkoordinatensystems) zum Werkstückbezugspunkt (Part-Teil).
 - Die andere Kette zeigt vom Maschinennullpunkt zum Werkzeugbezugspunkt (Tool-Teil).
- Eine der beiden Ketten kann auch leer sein (für reine Tisch- bzw. Werkzeugkinematiken).

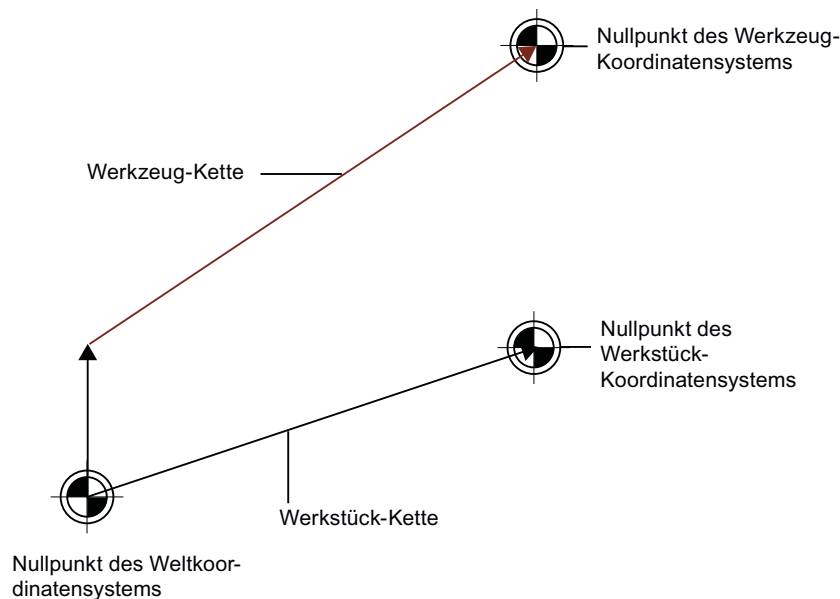


Bild 9-1 Definition einer Transformation über kinematische Ketten

9.1 Funktionsbeschreibung

Beide kinematischen Ketten werden intern zu einer einzigen kinematischen Kette zusammengefasst, die definitionsgemäß am Werkstückbezugspunkt beginnt und am Werkzeugbezugspunkt endet.

Definition der kinematischen Transformationen

Die Definition von kinematischen Transformationen über kinematische Ketten dient dazu, die Definition der bisherigen Transformationstypen zu vereinheitlichen.

Folgende Transformationen werden berücksichtigt:

- TRAORI
 - TRAORI_DYN; Dynamische Orientierungstransformation
 - TRAORI_STAT; Statische Orientierungstransformation
- TRANSMIT_K; Stirnseitentransformation
- TRACYL_K; Zylinderflächentransformation
- TRAANG_K; Schiefwinkeltransformation

Siehe auch

Systemvariablen (Seite 208)

K7: Kinematische Kette (Seite 201)

9.1.2 Definition kinematischer Transformationen

Die Definition von Transformationen mit kinematischen Ketten erfolgt über zwei Schritte:

- Definition der Maschinenkinematik
- Definition der Transformation

Maschinenkinematik definieren

Die Maschinenkinematik (Geometrie der Maschine) wird über kinematische Ketten beschrieben. Für detaillierte Informationen, siehe K7: Kinematische Kette (Seite 201).

Die dafür relevanten Systemvariablen sind unter Systemvariablen (Seite 208) beschrieben. Diese Systemvariablen sind nach dem Schema \$NK_... aufgebaut. Wichtige Systemvariablen sind z. B.:

Systemvariable	
\$NK_NAME	Name des aktuellen Elements
\$NK_NEXT	Name des nächsten Elements
\$NK_PARALLEL	Definiert eine parallele kinematische Kette. Damit kann parallel zur ersten Kette die zweite Kette definiert werden.

Systemvariable	
\$NK_TYPE	Definiert den Typ des Elements, z. B. AXIS_LIN, AXIS_ROT, OFFSET, ROT_CONST, SWITCH
\$NK_OFF_DIR	Definiert den Richtungsvektor bezogen auf die Maschinenachse
\$NK_AXIS	Name der Maschinenachse
\$NK_A_OFF	Definiert die Nullpunktverschiebung der zugeordneten Maschinenachse.

Transformation definieren

Die Beschreibung der Maschinenkinematik mit kinematischen Ketten reicht nicht aus, um eine kinematische Transformation vollständig zu spezifizieren. Über Systemvariablen mit dem Präfix \$NT_... können die Transformationen vollständig definiert werden. Die für die Transformationen verfügbaren Systemvariablen unterteilen sich in folgende Teile:

- Systemvariablen, die bei allen Transformationen relevant sind.
- Systemvariablen, die nur bei bestimmten Transformationen relevant sind.

Eine vollständige Liste finden Sie unter Systemvariablen für allgemeine Transformationstypen (Seite 349). Unter anderem stehen folgende allgemeine Systemvariablen zur Verfügung:

Systemvariable	Beschreibung
\$NT_NAME	Name des Transformationsdatensatzes.
\$NT_TRAFO_INDEX	Kennung für alternative Trafoaktivierung; Anstatt über die Methode TRAF00N, kann eine mit kinematischen Ketten definierte Transformation auch mit den konventionellen Sprachbefehlen (z. B. TRAORI) aktiviert werden.
\$NT_TRAFO_TYPE	Typ der Transformation (TRAORI_DYN, TRAORI_STAT, TRAANG_K, TRANSMIT_K, TRACYL_K)
\$NT_T_CHAIN_LAST_ELEM	Name des letzten Elements der kinematischen Kette zum Werkzeug .
\$NT_P_CHAIN_LAST_ELEM	Name des letzten Elements der kinematischen Kette zum Werkstück .
\$NT_T_CHAIN_FIRST_ELEM	Name des ersten Elements der Werkzeug-Kette
\$NT_P_CHAIN_FIRST_ELEM	Name des ersten Elements der Werkstück-Kette
\$NT_GEO_AX_NAME	Namen der Elemente der Geometrieachsen. Mit \$NT_GEO_AX_NAME wird auf maximal 3 Linearachsen verwiesen. Die Linearachsen führen die Ausgleichsbewegungen aus, die sich aus der kinematischen Transformation ergeben.
\$NT_ROT_AX_NAME	Namen der Elemente der Orientierungsachsen. Mit \$NT_ROT_AX_NAME wird auf maximal 3 rotatorische Achsen verwiesen.
\$NT_CLOSE_CHAIN_T	Referenzelement zum Schließen der kinematischen Kette zum Werkstück.
\$NT_ROT_AX_OFFSET	Positionsoffset einer/der Rundachse für die aktive Transformation.
\$NT_CNTRL	Steuerwort

Eine Orientierungstransformation wird beispielsweise über die in der Grafik dargestellten Elemente definiert.

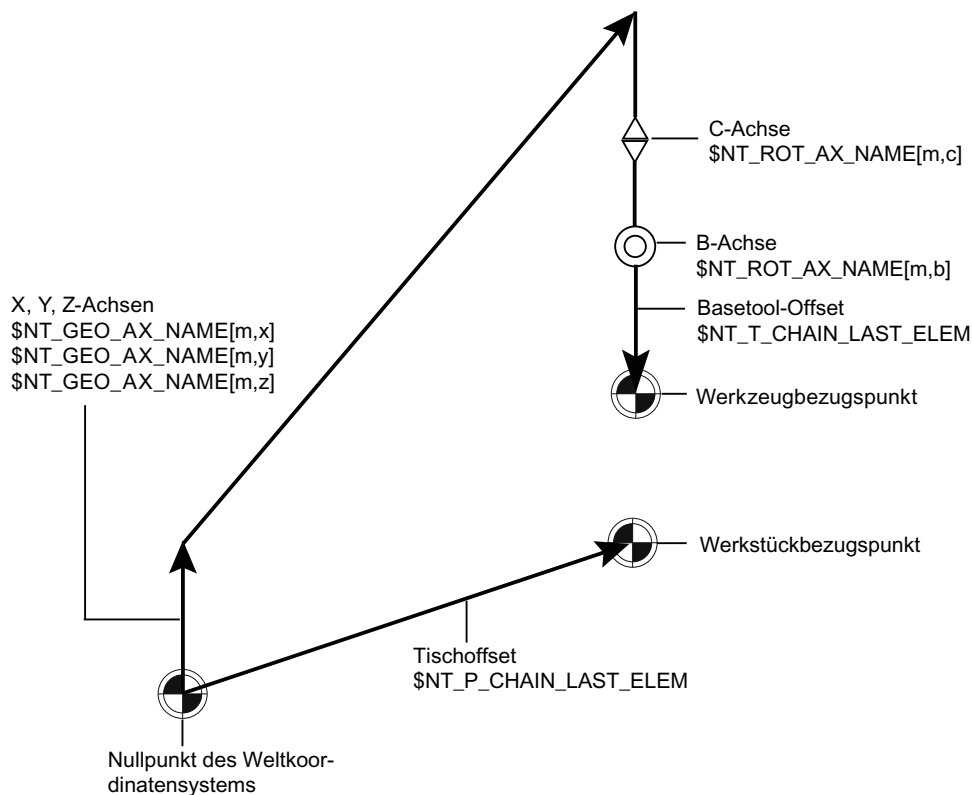


Bild 9-2 Transformationselemente in der kinematischen Kette

Siehe auch

Beispiele (Seite 394)

9.1.3 Dynamische Orientierungstransformation TRAORI_DYN

Unter einer dynamischen Orientierungstransformation wird eine kinematische Transformation verstanden, bei der die Bewegungen beliebiger Rundachsen durch Ausgleichsbewegungen von maximal drei Linearachsen so kompensiert werden, dass die Koordinaten der Werkzeugspitze im Werkstückkoordinatensystem unverändert bleiben. Gleiches gilt für Bewegungen zusätzlicher (redundanter) Linearachsen.

Die Systemvariablen speziell für Orientierungstransformationen sind unter Additive Systemvariablen für Orientierungstransformation (Seite 375) beschrieben.

Über die dynamische Orientierungstransformation werden die verschiedenen Orientierungstransformationen abgebildet:

5-Achstransformation

Für die 5-Achstransformation sind verschiedene Ausprägungen möglich, siehe 5-Achs-Transformation (Seite 44). Allgemein bestehen diese Transformation aus bis zu 3 Linearachsen und bis zu 2 Rundachsen.

3- und 4-Achstransformation

3- und 4-Achstransformationen sind Sonderfälle der 5-Achstransformation, siehe 3- und 4-Achs-Transformationen (Seite 58). Mögliche Ausprägungen sind unten aufgelistet:

Transformation	Merkmale
3-Achs-Transformation	2 translatorische Achsen (Linearachsen) 1 rotatorische Achse
4-Achs-Transformation	3 translatorische Achsen (Linearachsen) 1 rotatorische Achse

Maschinenkinematik definieren

Die entsprechende Maschinenkinematik muss über eine kinematische Kette definiert werden.

Linearachsen (Geometrieachsen)

Die Bewegungen der Rundachsen werden über Ausgleichsbewegungen der bis zu drei Linearachsen kompensiert. Die Linearachsen werden über die kinematische Kette als Element des Typs "AXIS_LIN" definiert. Für die Transformation wird auf sie mit \$NT_GEO_AX_NAME verwiesen. Die Geometrieachsen müssen nicht aufeinander senkrecht stehen, und ihre Stellung zueinander kann durch rotatorische Achsen verändert werden. Es gilt dabei jedoch die Einschränkung, dass sie dabei nicht linear voneinander abhängig werden dürfen.

Die Position von Linearachsen, die nicht als Geometrieachsen definiert sind, wird durch die Transformation nicht verändert.

Rund- und Orientierungsachsen

Die Rundachsen werden über die kinematische Kette als Element des Typs "AXIS_ROT" definiert. Für die Transformation wird auf sie mit \$NT_ROT_AX_NAME verwiesen und damit als Orientierungsachse gekennzeichnet. Die Position dieser Achsen wird als veränderlich angenommen, um eine bestimmte Orientierung zu erreichen.

Über die Systemvariable \$NT_ROT_AX_OFFSET kann für die Rundachsen (1, 2 oder 3) der Orientierungstransformation ein Offset gegenüber der Nullstellung vorgegeben werden.

Alle anderen Rundachsen sind redundante Rundachsen und gehen mit ihren aktuellen Positionswerten in die Berechnung ein (konstante Drehungen). Redundante Maschinenachsen (reale Rundachsen) müssen in der kinematischen Kette vor der ersten Orientierungsachse liegen. Weitere redundante Rundachsen (manuelle Rundachsen oder rotatorische Maschinenachsen) können an beliebiger Stelle eingefügt werden,

Die Anzahl der Rundachsen ist auf 6 begrenzt.

Werkzeugorientierung

Die Werkzeugorientierung kann vektoriell angegeben werden. Dafür stehen die Systemvariablen \$`NT_BASE_ORIENT` oder \$`NT_BASE_ORIENT_NORMAL` zur Verfügung. Die Orientierungen werden wirksam, wenn kein Werkzeug angewählt ist. Werden die Vektoren nicht explizit definiert, ist die VorabEinstellung (0, 0, 1).

- \$`NT_BASE_ORIENT`[n, 0 ...2]; definiert über einen Vektor die Werkzeuggrundorientierung, wie folgendes Beispiel zeigt:
 - \$`NT_BASE_ORIENT`[1, 0] = 1.0; Komponente in X-Richtung
 - \$`NT_BASE_ORIENT`[1, 1] = 1.5; Komponente in Y-Richtung
 - \$`NT_BASE_ORIENT`[1, 2] = 100; Komponente in Z-Richtung
- \$`NT_BASE_ORIENT_NORMAL`[n, 0 ...2] definiert einen Vektor, der bei Orientierungstransformationen mit drei Freiheitsgraden senkrecht auf der Werkzeuggrundorientierung steht. Werden die Vektoren nicht explizit definiert, ist die VorabEinstellung (0, 1, 0).
- \$`NT_IGNORE_TOOL_ORIENT`[n]; ist die Systemvariable gesetzt, werden auch bei aktivem Werkzeug die Werte aus den Systemvariablen \$`NT_BASE_ORIENT` und \$`NT_BASE_ORIENT_NORMAL` verwendet.

Behandlung von singulären Stellen (Pole)

Treten im Verlauf der Bearbeitung Pole (Seite 56) auf, lässt sich über zwei Systemvariablen das Verhalten am Pol einstellen:

- \$`NT_POLE_LIMIT`; definiert die Endwinkeltoleranz, also den Winkel, um den die Polachse bei der 5-Achstransformation vom programmierten Wert abweichen darf.
- \$`NT_POLE_TOL`; definiert die Endwinkel-Toleranz bei Interpolation durch den Pol.

Maschinenvermessung für Orientierungstransformation

Mit der Funktion `CORRTRAF0` (Seite 387) können gemessene Hebelarmlängen und Achsrichtungen in Korrektur Elemente geschrieben werden:

- \$`NT_CORR_ELEM_T`[n, 0 ...3]; Namen der Korrektur Elemente (Typ "OFFSET") in der kinematischen Kette (**Werkzeug-Kette**), die von `CORRTRAF0` beschrieben werden können.
- \$`NT_CORR_ELEM_P`[n, 0 ...3]; Namen der Korrektur Elemente (TYP "OFFSET") in der kinematischen Kette (**Werkstück-Kette**), die von `CORRTRAF0` beschrieben werden können.

Echtzeit-Geschwindigkeitsüberwachung

Mit den Maschinendaten \$`MC_TRAFO_ORI_ONLINE_CHECK_LIM` und \$`MC_TRAFO_ORI_ONLINE_CHECK_LIMR` wird definiert, ab welcher linearen bzw. rotatorischen Positionsdifferenz zwischen Vor- und Hauptlauf die Echtzeit-Geschwindigkeitsüberwachung und -begrenzung aktiviert wird. Die Positionsdifferenzen werden aus den Positionen der Linear- bzw. Rundachsen gebildet.

Beispiel-Kinematik

Das Beispiel zeigt schematisch eine Zwei-Achs-Schwenkkopf-Kinematik mit der Achsfolge CA, als räumliches Bild und als kinematische Kette.

Schwenkkopf:
Rundachsen C und A

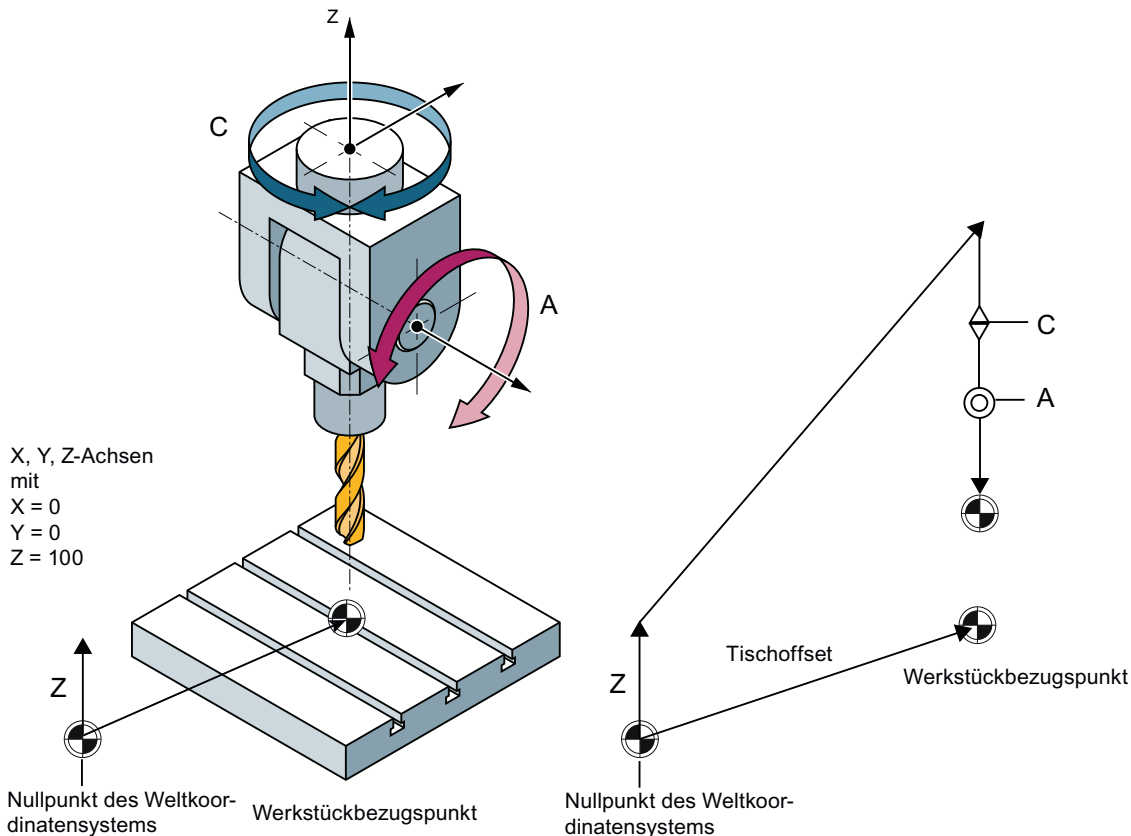


Bild 9-3 TRAORI Beispiel

9.1.4 Stirnseitentransformation TRANSMIT

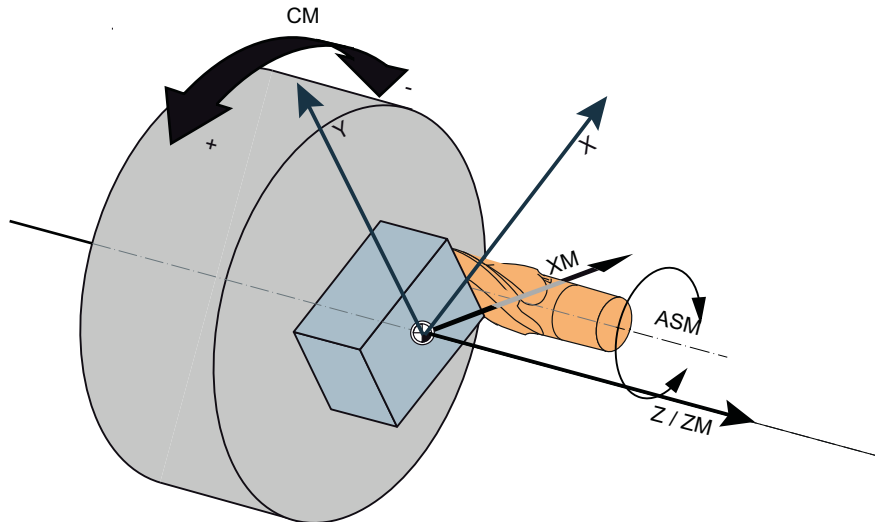
Im vorliegenden Kapitel wird beschrieben, wie eine Stirnseitentransformation mittels einer kinematischen Kette abgebildet und in der Steuerung über Systemvariablen parametrisiert wird. Die Transformation TRANSMIT ermöglicht die Stirnseitenbearbeitung (Bohrungen, Konturen) an Drehmaschinen.

Die Systemvariablen werden in der NC remanent gespeichert und können über SINUMERIK Operate mittels Inbetriebnahmearchiv als "NC-Daten" archiviert bzw. eingelesen werden.

9.1 Funktionsbeschreibung

Die Maschinenkinematik wird über zwei kinematische Ketten definiert, siehe Definition kinematischer Transformationen (Seite 328).

Die folgende Grafik zeigt beispielhaft eine Stirnseitentransformation:



- X, Y, Z Geometrieachsen
- CM Maschinenachse; Rundachse
- XM Maschinenachse; Linearachse, senkrecht zur Rundachse
- ZN Maschinenachse; Linearachse, parallel zur Rundachse
- ASM Maschinenachse; Arbeitspindel

Bild 9-4 TRANSMIT

Randbedingungen

- Es muss genau eine Rundachse (die Polachse) geben, deren Name in \$NT_ROT_AX_NAME[n,1] steht.
- Ein bis drei Einträge für Linearachsen in den Systemparametern \$NT_GEO_AX_NAME[n,i] sind möglich.

Koordinatensysteme für Stirnseitentransformationen

Das Werkstückkoordinatensystem und das Basiskoordinatensystem sind unabhängig vom Weltkoordinatensystem, in dem die kinematischen Ketten definiert sind.

- Die Z-Achse steht parallel zur Polachse (Achse CM - siehe oben) und parallel zur Längsachse, falls diese definiert ist.
- Die Geometrieachsen X, Y und Z stehen senkrecht zueinander und bilden ein rechtwinkeliges Koordinatensystem in der Reihenfolge 1., 2. und 3. Geometrieachse.
- Ist die Längsachse (\$NT_GEO_AX_NAME[n, 2]) vorhanden, muss sie immer parallel zur Rundachse (\$NT_ROT_AX_NAME[n, 1]) sein.

- Sind zwei Linearachsen vorhanden, müssen sie nicht orthogonal zueinander sein.
- Sind drei Linearachsen vorhanden, müssen zwei Achsen orthogonal zueinander stehen. Eine dritte Achse kann gegenüber einer der beiden anderen Achsen schräg stehen. Es kann nur ein Achspaar geben, bei dem der Winkel der beiden Achsen ungleich 90° ist. Eine schräge Achse darf nicht senkrecht auf einer weiteren Achse stehen.

Kinematische Ketten für Stirnseitentransformationen

Die Transmitttransformation wird über die Systemvariable \$NT_TRAFO_TYPE = "TRANSMIT_K" aktiviert.

Definition der Linearachsen

- Die X-Achse wird über die Systemvariable \$NT_GEO_AX_NAME[n,0] definiert. Das ist die Linearachse der eigentlichen TRANSMIT-Transformation. Im Standardfall steht diese Achse senkrecht auf der Polachse (siehe oben).
- Ein Eintrag in \$NT_GEO_AX_NAME[n, 2] ist optional. Diese Achse (typischerweise die Z-Achse) ist die Längsachse. Sie muss parallel zu Rundachse (\$NT_ROT_AX_NAME[n, 1]) definiert sein.
- Ein Eintrag in \$NT_GEO_AX_NAME[n, 1] ist optional. Diese Achse (typischerweise die Y-Achse) steht im Standardfall senkrecht auf der Radialachse (\$NT_GEO_AX_NAME[n, 0]) und der Polachse (\$NT_ROT_AX_NAME[n, 1]). Diese Achse (die Mittenversatzachse) dient in der Regel dazu, einen Werkzeugversatz in Y-Richtung zu kompensieren, sodass eine Bearbeitung bis zur Drehmitte möglich ist.

Definition der Rundachse

- Die Rundachse wird über die Systemvariable \$NT_ROT_AX_NAME[n,1] definiert. Diese Achse muss immer vorhanden sein. Die Drehrichtung der Rundachse wird über \$NT_CNTRL Bit 11 definiert.
- Über die Systemvariable \$NT_ROT_AX_OFFSET[n, 0] kann für die aktive TRANSMIT-Transformation ein Offset vorgegeben werden.

Reihenfolge der Achsen festlegen

Über die Systemvariable \$NT_CNTRL[n] (Seite 370) werden verschiedene Einstellungen für die Transformationen festgelegt. Mit den Bits 16-18 wird binär kodiert, wie die Kanalachsen, die in \$NT_ROT_AX_NAME[n, 1], in \$NT_GEO_AX_NAME[n, 0] und in \$NT_GEO_AX_NAME[n, 2] eingetragen sind, den Geometrieachsen zugeordnet werden. Die drei Binärzahlen müssen folgenden Dezimalzahlen entsprechen:

Dezimal	Reihenfolge der Geometrieachsen		
0	X	Y	Z
1	Z	X	Y
2	Y	Z	X
3	Z	Y	X
4	X	Z	Y
5	Y	X	Z

Verhalten beim Durchfahren des Pols

Das Verhalten beim Durchfahren des Pols wird über die Systemvariable \$NT_POLE_SIDE_FIX definiert.

WERT	Bedeutung
0	Poldurchfahrung Die Werkzeugmittelpunktsbahn (Linearachse) soll stetig durch den Pol führen.
1	Drehung um den Pol. Die Werkzeugmittelpunktsbahn soll sich ausschließlich im positiven Verfahrbereich der Linearachse befinden. (Vor der Drehmitte).
2	Drehung um den Pol. Die Werkzeugmittelpunktsbahn soll sich ausschließlich im negativen Verfahrbereich der Linearachse befinden. (Hinter der Drehmitte).

Offset für das Basiswerkzeug

Über die Systemvariable \$NT_BASE_TOOL_COMP (Seite 374) kann für die jede Geometrieachse getrennt eingestellt werden, ob beim Aktivieren der Transformation ein Offset für das Basiswerkzeug in den Transformationsframe (\$P_TRAFRAME) eingetragen wird. Mit dem Offset werden die Komponenten des Basiswerkzeugs so kompensiert, dass keine Änderung in der Komponente des WKS auftritt. Für diese Funktion muss \$P_TRAFRAME über das Maschinendatum \$MC_MM_SYSTEM_FRAME_MASK Bit 6 projiziert sein.

Die Kompensation kann z. B. bei einem Werkzeugträger angewendet werden, bei dem die Abmessungen des Kopfteils des Werkzeugträgers nicht Bestandteil der Werkzeuglänge sind.

Die Trennstelle zwischen Basiswerkzeug und Werkzeug kann über die Systemvariable \$NT_T_REF_ELEM definiert werden. Ist die Systemvariable leer, dann wird die Trennstelle am Endpunkt der Werkzeugkette definiert.

Nullpunktverschiebung der Rundachse

Über die Systemvariable \$NT_ROT_OFFSET_FROM_FRAME (Seite 366) wird eingestellt, ob der Offset für Orientierungsachsen automatisch übernommen werden soll. Wird in der Systemvariable ein Wert definiert, dann wird der programmierte Offset automatisch beim Einschalten aus der für Orientierungsachsen aktiven Nullpunktverschiebung übernommen:

Rundachsoffset bei Anwahl der Transformation aus der Nullpunktverschiebung übernehmen:

- 0 = axiale Verschiebung der Rundachse wird nicht berücksichtigt.
- 1 = axiale Verschiebung der Rundachse wird berücksichtigt.
- 2 = axiale Verschiebung der Rundachse wird bis zum ENS berücksichtigt. ENS entspricht dem Einstellbaren Nullpunktsystem. Die ENS-Frames enthalten transformierte Drehungen um die Rundachse.

Definition des letzten Elements der kinematischen Kette

Über \$NT_CNTRL Bit 19 wird definiert, dass das letzte Element der kinematischen Kette eine Rundachse oder eine konstante Drehung ist. Der Richtungsvektor der Rundachse definiert die Z-Richtung des Werkzeugkoordinatensystems. Enthält die Systemvariable \$NK_A_OFF dieses Kettenelements einen Wert ungleich Null, wird das Werkzeugkoordinatensystem zusätzlich mit diesem Winkel um die Koordinatenachse gedreht.

Ist zusätzlich \$NT_CNTRL Bit 20 definiert, wird das Vorzeichen der Z-Richtung der Achse für die Ermittlung des WKS invertiert.

Beispielprogramm für TRANSMIT

Ein einfaches Teileprogramm für TRANSMIT finden Sie unter Teileprogramm für TRANSMIT (Seite 401).

9.1.5 Zylindermanteltransformation TRACYL

Im vorliegenden Kapitel wird beschrieben, wie Zylindermanteltransformationen (TRACYL) mittels einer kinematischen Kette abgebildet und in der Steuerung über Systemvariablen parametrisiert werden.

Die Systemvariablen werden in der NC remanent gespeichert und können über SINUMERIK Operate mittels Inbetriebnahme-archiv als "NC-Daten" archiviert bzw. eingelesen werden.

Die Maschinenkinematik wird über zwei kinematische Ketten definiert, siehe Definition kinematischer Transformationen (Seite 328).

Die folgende Grafik zeigt beispielhaft eine Zylindermanteltransformation:

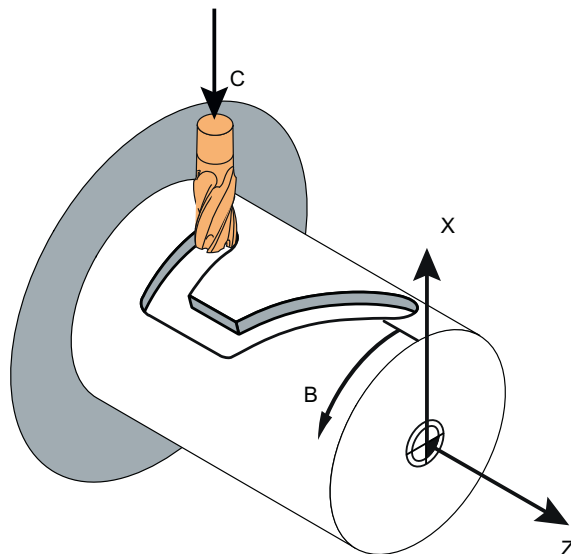


Bild 9-5 Zylindermanteltransformation (TRACYL)

Transformation mit TRAF00N aktivieren

Neben dem Namen der Transformation kann bei TRACYL noch der Referenzdurchmesser angegeben werden und eingestellt werden, ob die Funktion "Nutwandkorrektur" aktiv ist:

TRAF00N (<Trafoname>, <diameter>, <k>)

Mit $k = 1$ wird die Nutwandkorrektur ausgeführt siehe Transformation einschalten (TRAF00N) (Seite 386).

Randbedingungen

- Es muss genau eine Rundachse (die Polachse) geben, deren Name in \$NT_ROT_AX_NAME[n,1] steht.
- Es kann 1 bis 3 Einträge für Linearachsen in den Systemparametern \$NT_GEO_AX_NAME[n,i] geben.

Koordinatensysteme für Zylindermanteltransformationen

Das Werkstückkoordinatensystem und das Basiskoordinatensystem sind unabhängig vom Weltkoordinatensystem, in dem die kinematischen Ketten definiert sind.

- Die Z-Achse steht parallel zur Polachse (Achse B - siehe oben) und parallel zur Längsachse, falls diese definiert ist.
- Die Geometrieachsen X, Y und Z stehen senkrecht zueinander und bilden ein rechtwinkeliges Koordinatensystem in der Reihenfolge 1., 2. und 3. Geometrieachse.
- Ist die Längsachse (\$NT_GEO_AX_NAME[n, 2]) vorhanden, muss sie immer parallel zur Rundachse (\$NT_ROT_AX_NAME[n, 1]) sein.
- Sind zwei Linearachsen vorhanden, müssen sie nicht orthogonal zueinander sein.
- Sind drei Linearachsen vorhanden, müssen zwei Achsen orthogonal zueinander stehen. Eine dritte Achse kann gegenüber einer der beiden anderen Achsen schräg stehen. Es kann nur ein Achspaar geben, bei dem der Winkel der beiden Achsen ungleich 90° ist. Eine schräge Achse darf nicht senkrecht auf einer weiteren Achse stehen.

Kinematische Ketten für Zylindermanteltransformationen

Die Zylindermanteltransformation wird über die Systemvariable \$NT_TRAFO_TYPE = "TRACYL_K" aktiviert.

Definition der Linearachsen

- Die X-Achse wird über die Systemvariable $\$NT_GEO_AX_NAME[n,0]$ definiert. Das ist die Linearachse der eigentlichen TRACYL-Transformation. Im Standardfall steht diese Achse senkrecht auf der Rundachse (siehe oben). An dieser Achse wird die Nuttiefe eingestellt.
- Ein Eintrag in $\$NT_GEO_AX_NAME[n, 1]$ ist optional. Diese Achse (typischerweise die Y-Achse) steht im Standardfall senkrecht auf der Radialachse ($\$NT_GEO_AX_NAME[n, 0]$) und der Polachse ($\$NT_ROT_AX_NAME[n, 1]$).
Diese Achse ist die Nutwand-Korrekturachse. Mit dieser Achse wird bei aktiver Werkzeugradiuskorrektur der Versatz am Umfang des Bearbeitungszyllinders ausgeglichen, sodass die Nutseitenwände rechtwinkelig zum Nutboden bleiben.
- Ein Eintrag in $\$NT_GEO_AX_NAME[n, 2]$ ist optional. Diese Achse (typischerweise die Z-Achse) ist die Längsachse. Sie muss parallel zu Rundachse ($\$NT_ROT_AX_NAME[n, 1]$) definiert sein.

Definition der Rundachse

- Die Rundachse wird über die Systemvariable $\$NT_ROT_AX_NAME[n,1]$ definiert. Diese Achse muss immer vorhanden sein. Die Drehrichtung der Rundachse wird über $\$NT_CNTRL$ Bit 11 definiert.
- Über die Systemvariable $\$NT_ROT_AX_OFFSET[n, 0]$ kann für die aktive TRACYL-Transformation ein Offset vorgegeben werden.

Reihenfolge der Achsen festlegen

Über die Systemvariable $\$NT_CNTRL[n]$ (Seite 369) werden verschiedene Einstellungen für die Transformationen festgelegt. Mit den Bits 16-18 wird binär kodiert, wie die Kanalachsen, die in $\$NT_ROT_AX_NAME[n, 1]$, $\$NT_GEO_AX_NAME[n, 0]$ und $\$NT_GEO_AX_NAME[n, 2]$ eingetragen sind, den Geometrieachsen zugeordnet werden. Die drei Binärzahlen müssen folgenden Dezimalzahlen entsprechen:

Dezimal	Reihenfolge der Geometrieachsen		
0	X	Y	Z
1	Z	X	Y
2	Y	Z	X
3	Z	Y	X
4	X	Z	Y
5	Y	X	Z

Nutwandkorrektur

In Verbindung mit Bit 9 = 1 kann bei der Aktivierung der Transformation über TRAF00N eingestellt werden, ob TRACYL mit oder ohne Nutwandkorrektur betrieben wird. "Nutwand aktiv" entspricht dem Transformationstyp 514.

Falls mit Nutwandkorrektur gearbeitet wird, muss $\$NT_GEO_AX_NAME[n,1]$ einen Verweis auf eine Linearachse (Nutwandkorrekturachse) enthalten.

Folgende Einstellungen über \$NT_CNTRL sind mit Bit 9 und Bit 10 möglich:

Bit 10	Bit 9	Beschreibung
0	0	TRACYL ohne Nutwandkorrektur
0	1	TRACYL mit Nutwandkorrektur
1	0	TRACYL mit der Möglichkeit, zwischen den Funktionen mit und ohne Nutwandkorrektur über einen Transformationsparameter bei der Aktivierung der Transformation (TRAFOON) umzuschalten. Die Nutwandkorrektur ist defaultmäßig nicht aktiv.
1	1	TRACYL mit der Möglichkeit, zwischen den Funktionen mit und ohne Nutwandkorrektur über einen Transformationsparameter bei der Aktivierung der Transformation (TRAFOON) umzuschalten. Die Nutwandkorrektur ist defaultmäßig aktiv.

Nullpunktverschiebung der Rundachse

Über die Systemvariable \$NT_ROT_OFFSET_FROM_FRAME (Seite 366) wird eingestellt. Wenn in der Systemvariablen ein Wert definiert wird, dann wird der programmierte Offset automatisch beim Einschalten aus der für Orientierungsachsen aktiven Nullpunktverschiebung übernommen.

Rundachsoffset bei Anwahl der Transformation aus der Nullpunktverschiebung übernehmen:

- 0 = axiale Verschiebung der Rundachse wird nicht berücksichtigt.
- 1 = axiale Verschiebung der Rundachse wird berücksichtigt.
- 2 = axiale Verschiebung der Rundachse wird bis zum ENS berücksichtigt. ENS entspricht dem Einstellbaren Nullpunktsystem. Die ENS-Frames enthalten transformierte Drehungen um die Rundachse.

Definition des letzten Elements der kinematischen Kette

Über \$NT_CNTRL Bit 19 wird definiert, dass das letzte Element der kinematischen Kette eine Rundachse oder eine konstante Drehung ist. Der Richtungsvektor der Rundachse definiert die Z-Richtung des Werkzeugkoordinatensystems. Enthält die Systemvariable \$NK_A_OFF dieses Kettenelements einen Wert ungleich Null, wird das Werkzeugkoordinatensystem zusätzlich mit diesem Winkel um die Koordinatenachse gedreht.

Ist zusätzlich \$NT_CNTRL Bit 20 definiert, wird das Vorzeichen der Z-Richtung der Achse für die Ermittlung des WKS invertiert.

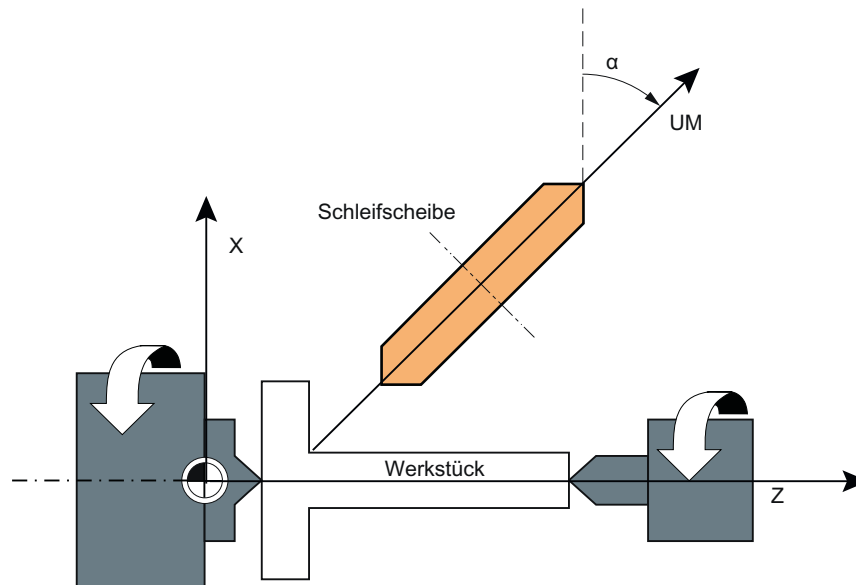
Beispielprogramm für TRACYL

Ein einfaches Teileprogramm für TRACYL finden Sie unter Teileprogramm für TRACYL (Seite 404).

9.1.6 Schiefwinkeltransformation (Schräge Achse) TRAANG_K

Im vorliegenden Kapitel wird beschrieben, wie eine Schiefwinkeltransformation (TRAANG_K) mittels einer kinematischen Kette abgebildet und in der Steuerung über Systemvariablen parametrisiert wird.

Die Systemvariablen werden in der NC remanent gespeichert und können über SINUMERIK Operate mittels Inbetriebnahme-archiv als "NC-Daten" archiviert bzw. eingelesen werden.



X Geometrieachse
 Z Geometrieachse
 ZM Maschinenachse
 UM Maschinenachse
 α Winkel der schrägen Achse

Bild 9-6 TRAANG_K Beispiel

Die Schiefwinkeltransformation ist ein Spezialfall der Orientierungstransformation.

Die Schiefwinkeltransformation wird über die Systemvariable \$NT_TRAFO_TYPE = "TRAANG_K" aktiviert.

Beispielprogramm für TRAANG

Ein einfaches Teileprogramm für TRAANG finden Sie unter Teileprogramm für TRAANG (Seite 407).

9.1.7 Statische Orientierungstransformation TRAORI_STAT

Statische Orientierungstransformationen unterscheiden sich dadurch von den dynamischen, dass es nicht möglich ist, interpolatorische Ausgleichsbewegungen so auszuführen, dass die Werkzeugspitze im Werkstückkoordinatensystem die im NC-Programm programmierte Bahn einhält. Es stehen jedoch alle Funktionen zur Verfügung, die korrekten Endpunkte unter Berücksichtigung der erforderlichen Ausgleichsbewegungen anzufahren.

Eine Statische Orientierungstransformation ist bei Maschinen sinnvoll, bei denen die einmal eingestellte Orientierung fest und während des Verfahrens nicht veränderbar ist, beispielsweise bei Drehen auf einer Fräsmaschine.

Die statische Orientierungstransformation wird mit denselben Systemvariablen definiert, wie eine dynamische Orientierungstransformation, siehe Dynamische Orientierungstransformation TRAORI_DYN (Seite 330) und Definition kinematischer Transformationen (Seite 328).

Achsen bei statischer Orientierungstransformation

Die Orientierungsachsen statischer Orientierungstransformationen können auch Spindeln oder Hirth-verzahnte Achsen sein. Spindeln verhalten sich wie konstante kinematische Drehungen, d. h. soll diesen in der kinematischen Transformation eine Position zugeordnet werden, muss dies in den zugehörigen Systemparametern \$NK_A_OFF[n] eingetragen werden.

Der Typ "Statische Orientierungstransformation" wird über die Systemvariable \$NT_TRAFO_TYPE = TRAORI_STAT aktiviert.

Systemvariablen zur Definition der statischen Orientierungstransformation

Die Maschinenkinematik wird über die kinematische Kette definiert, siehe Definition kinematischer Transformationen (Seite 328).

Die statische Orientierungstransformation wird über folgende Systemvariablen definiert:

- Systemvariablen für allgemeine Transformationstypen (Seite 349)
- Additive Systemvariablen für Orientierungstransformation (Seite 375)

Referenz

Informationen zu Orientierbaren Werkzeugträgern über kinematische Ketten finden Sie im Funktionshandbuch "Grundfunktionen" im Abschnitt "Werkzeugkorrektur" unter "Orientierbare Werkzeugträger".

9.1.8 Transformationen verketteten (TRACON_K)

Bei Transformationen mit kinematischen Ketten können schiefwinklige Achsen über die kinematische Kette definiert werden. Damit müssen Transformationen nicht mehr mit TRACON verkettet werden, um schiefwinklige Achsen abzubilden.

Der Transformationstyp "TRACON_K" (Seite 353) wird zusammen mit kinematischen Ketten verwendet, um eine "Open Architecture (OA)"-Transformation mit einer der Standard-Transformationen zu verketteten. Einzelheiten sind der "Open Architecture (OA)"-Dokumentation zu entnehmen.

Literatur:

Eine ausführliche Beschreibung der Verkettung von Transformationen ohne kinematische Ketten findet sich im Funktionshandbuch "Erweiterungsfunktionen" im Abschnitt "Kinematische Transformationen" unter "Verkettete Transformationen".

9.1.9 Bei Reset wirksame Transformationen

Bei Reset aktive Transformation

Die bei einem Reset wirksame kinematische Transformation wird über das Maschinendatum \$MC_TRAFO_RESET_NAME definiert.

Wurde kein Name für \$MC_TRAFO_RESET_NAME definiert, ist \$MC_TRAFO_RESET_VALUE aktiv beziehungsweise es ist keine Transformation bei einem Reset aktiv. Der Inhalt dieses Maschinendatums gibt die Nummer der zu aktivierenden konventionellen Transformation an. Konventionelle Transformationen sind nicht über kinematische Ketten definiert, sondern über Maschinendaten.

Weiterhin wirkt das Maschinendatum \$MC_RESET_MODE_MASK Bit7 wie bei konventionellen Transformationen. Ist Bit 7 gesetzt, bleibt die aktive Transformation über RESET bzw. Teileprogrammende hinweg erhalten.

9.1.10 Persistente Transformationen

Persistente Transformation

Eine persistente Transformation ist eine Transformation, die auch beim Ausschalten einer (anderen) Transformation mit TRAFOOF erhalten bleibt. Diese Transformation ist identisch mit der bei Reset (Seite 343) wirksamen Transformation.

Bei Anwahl einer anderen Transformation wird die persistente Transformation durch die neu angewählte Transformation ersetzt. Die neu angewählte Transformation wird NICHT mit der persistenten Transformation verkettet.

9.1.11 Migration von kinematischen Transformationen

Herkömmliche Transformationen

Unter herkömmlichen Transformationen werden solche verstanden, die über Maschinendaten definiert werden. Diese Transformationsdefinitionen bleiben weiterhin gültig und können ungeschränkt verwendet werden.

Die ab V4.8. SP2 verfügbare Variante erlaubt es, kinematische Transformationen auch über kinematische Ketten und Systemvariablen zu definieren.

Daten nur einmal halten

Bei der parallelen Nutzung herkömmlicher Transformationen und Transformationen über kinematische Ketten tritt das Problem auf, dass gegebenenfalls identische Maschinenkinematiken mit verschiedenen NC-Daten parametrisiert werden, nämlich einmal mit Maschinendaten und einmal mit Daten aus den Systemvariablen der kinematischen Ketten und der Transformationen. Dadurch können Inkonsistenzen in der Datenhaltung auftreten.

Transformation aktivieren

Für die Aktivierung von kinematischen Transformationen stehen die folgenden Funktionen zur Verfügung:

- Syntax für Transformationen über kinematische Ketten: TRAF00N (Seite 386)(<name>)
- Syntax für herkömmliche Transformationen: TRAORI(...), TRANSMIT(...), TRACYL(...), TRAANG(...).

Transformationen, die über kinematische Ketten definiert wurden, können über die Syntax für herkömmliche Transformationen aufgerufen werden.

Dazu muss die aufzurufende Transformation über die Systemvariable \$NT_TRAFO_INDEX (Seite 352) definiert werden.

Weiter gilt:

- Wurde eine Transformation herkömmlich und mit kinematischer Kette definiert, wird die Transformation über kinematische Kette aufgerufen.
- Es wird nicht überprüft, ob die aufgerufene Transformation von einem Typ ist, der mit dem Trafotyp des Originalaufrufs kompatibel ist.
- Aufrufparameter eines konventionellen Transformationsaufrufs werden an die mit kinematischen Ketten definierte Transformation weitergeleitet.

Einschränkungen für TRAANG_K

Bei einer konventionellen TRAANG-Transformation (Transformation „schräge Achse“) ist der Winkel der schrägen Achse gegenüber der entsprechenden Koordinatenachse in einem rechtwinkligen Koordinatensystem durch Maschinendaten (24700 \$MC_TRAANG_ANGLE_x) festgelegt.

Dieser Winkel kann aber beim Transformationsaufruf, z. B. TRAANG(< α >), über einen Parameters verändert werden.

Die Angabe des Winkelparameters ist deshalb nur dann zulässig, wenn dieser Winkel identisch ist mit dem Winkel, der sich aus der Definition der kinematischen Kette ergibt.

Orientierungstransformationen

Für Orientierungstransformationen, die mit kinematischen Ketten parametrisiert sind, ist aus Kompatibilitätsgründen der Inhalt der Systemvariablen \$P_TRAFO definiert.

\$P_TRAFO liefert bei aktiver Orientierungstransformation die Werte, die bei konventioneller Parametrierung einer entsprechenden Transformation im Maschinendatum \$MC_TRAFO_TYPE_x enthalten wären, d. h. für "Werkzeug"-Kinematiken den Wert 24, für "Werkstück"-Kinematiken den Wert 40 und für gemischte Kinematiken entweder den Wert 56 oder den Wert 57.

Der Wert 57 kann dabei nur bei 6-Achs-Transformationen auftreten, wenn die Werkzeug-Kette eine und die Werkstück-Kette zwei Orientierungsachsen enthält.

9.1.12 Frames bei kinematischen Transformationen

Frames zur Beschreibung von kinematischen Ketten

Mit den Systemvariablen \$P_TRAFRAME_T bzw. \$P_TRAFRAME_P können Frames ausgelesen werden, die die Verschiebung und die Rotation eines am Werkzeug Bezugspunkt bzw. am Werkstück Bezugspunkt befestigten Koordinatensystems gegenüber dem Nullpunkt des Weltkoordinatensystems beschreiben.

Der Inhalt dieser Framevariablen kann auch über die BTSS-Schnittstelle gelesen werden.

Der Nullpunkt des Weltkoordinatensystems ist der Punkt, von dem ausgehend die kinematischen Ketten (Seite 327) zur Beschreibung der Maschinenkinematik definiert sind.

9.1.13 Werkzeuglängen

Werkzeuglängen

Wenn der Anteil der kinematischen Kette, die zum Werkzeug führt, ein Teil des Werkzeugs ist, sind der Werkzeug Bezugspunkt und der Werkzeugreferenzpunkt, der über \$NT_T_REF_ELEM gesetzt wurde, nicht mehr identisch.

Für die NC-Funktionen, GETTCORR, SETTCORR und LENTOAX, für die der Werkzeug Bezugspunkt von Bedeutung ist, gilt Folgendes:

Der Anteil der kinematischen Kette, der zwischen dem Werkzeugreferenzpunkt und dem Werkzeug Bezugspunkt liegt, wird so behandelt wie ein orientierbarer Werkzeugträger, der über Maschinendaten definiert wurde.

9.2 Inbetriebnahme

9.2.1 Allgemein

9.2.1.1 Übersicht

Die Inbetriebnahme der Funktion "Transformationen über Kinematische Ketten" erfolgt mittels:

- Maschinendaten
 - Vorgabe des Mengengerüsts
 - Festlegung des ersten Elements der kinematischen Kette
- Systemvariablen
 - Festlegung des Transformationstyps
 - Festlegung der kinematischen Eigenschaften eines Elements
 - Verbindung der Elemente zur kinematischen Kette

9.2.1.2 Aufbau der Systemvariablen

Die Systemvariablen sind nach folgendem Schema aufgebaut:

- **\$NT_<Name>[<Index_1>]**
- **\$NT_<Name>[<Index_1>, <Index_2>]**

Allgemein

Die Systemvariablen zur Beschreibung der Elemente von kinematischen Ketten haben folgende Eigenschaften:

- Der Präfix für alle Systemvariablen der kinematischen Kette ist **\$NT_**, (N für NC, K für Transformation).
- Die Systemvariablen sind über NC-Programme les- und schreibbar.
- Die Systemvariablen können über Archive gesichert und wieder in die NC eingelesen werden.

Datentyp

STRING

Alle Systemvariablen vom Datentyp STRING haben folgende Eigenschaften:

- Maximale String-Länge: 31 Zeichen
- Es wird keine Unterscheidung zwischen Groß- und Kleinschreibung gemacht
Beispiel: "Achse1" identisch mit "ACHSE1"

- Leer- und Sonderzeichen sind zulässig
Beispiel: "Achse1" nicht identisch mit " Achse 1"
- Namen, die mit **zwei** Unterstrichen "__" **beginnen**, sind für Systemzwecke reserviert und dürfen **nicht** für anwenderdefinierte Namen verwendet werden.

Hinweis**Führendes Leerzeichen**

Da Leerzeichen gültige und der Unterscheidung dienende Zeichen sind, dürfen Namen, die mit einem **Leerzeichen**, gefolgt von **zwei** Unterstrichen "__" **beginnen**, prinzipiell für anwenderdefinierte Namen verwendet werden. Aufgrund der Verwechslungsgefahr mit Systemnamen wird dieses Vorgehen **nicht** empfohlen.

Index_1

Über Index_1 werden die einzelnen Elemente adressiert. Index 0 ist reversiert, Index 1 → 1. Element, Index 2 → 2. Element, ... (n+1) → (n+1) Element, mit $n = (\$MN_MM_NUM_KIN_TRAFOS - 1)$

Alle Systemvariablen eines Elements haben den gleichen Index.

Index_2

Bei Systemvariablen, die einen Vektor enthalten, werden über Index_2 die Koordinaten des Vektors adressiert.

- 0 → X-Achse
- 1 → Y-Achse
- 2 → Z-Achse

9.2.2 Maschinendaten**9.2.2.1 Maximale Anzahl von Transformationen mit Kinematischen Ketten**

Mit dem Maschinendatum wird die maximale Anzahl der mit kinematischen Ketten definierbaren Transformationen eingestellt

MD18866 \$MN_MM_NUM_KIN_TRAFOS = <Anzahl>

9.2.2.2 Name der Resettransformation

Mit dem Maschinendatum wird der Name einer Transformation spezifiziert, die im Hochlauf/Reset oder bei Teileprogrammende angewählt wird.

MD20142 \$MC_TRAFO_RESET_NAME = <Name>

9.2.2.3 Aktivierungsgrenze der Echtzeitdynamiküberwachung (Linearachsen)

Mit dem Maschinendatum wird die Aktivierungsgrenze der Echtzeitdynamiküberwachung für Linearachsen spezifiziert. Die Echtzeitdynamikbegrenzung wird dann aktiviert, wenn an einer an der Transformation beteiligten Linearachse oder die effektive Werkzeuglänge bei einer Orientierungstransformation um mehr als die definierte Grenze abweicht.

MD21198 \$MC_ORI_TRAFO_ONLINE_ - CHECK_LIM = <Value>

9.2.2.4 Aktivierungsgrenze der Echtzeitdynamiküberwachung (Rundachsen)

Mit dem Maschinendatum wird die Aktivierungsgrenze der Echtzeitdynamiküberwachung für Rundachsen spezifiziert. Die Echtzeitdynamikbegrenzung wird dann aktiviert, wenn an einer an der Transformation beteiligten Linearachse oder die effektive Werkzeuglänge bei einer Orientierungstransformation um mehr als die definierte Grenze abweicht.

MD21198 \$MC_ORI_TRAFO_ONLINE_ - CHECK_LIMR = <Value>

9.2.2.5 Korrekturwert für Offsetvektoren bei CORRTRAFO

Mit dem Settingdatum wird der maximal zulässige Korrekturwert für Offsetvektoren bei CORRTRAFO parametrieret.

SD41610 \$SN_CORR_TRAFO_LIN_MAX = <Value>

9.2.2.6 Winkelabweichung für Richtungsvektoren bei CORRTRAFO

Mit dem Settingdatum wird die maximal zulässige Winkelabweichungen für Richtungsvektoren bei CORRTRAFO parametrieret.

SD41611 \$SN_CORR_TRAFO_DIR_MAX = <Name>

9.2.3 Verwendung der Systemvariablen bei kinematischen Transformationen

Verwendungstabelle

In der nachfolgenden Tabelle ist dargestellt, welche \$NT-Daten für die einzelnen Trafotypen relevant sind.

Variable	BTSS (Spaltenind.)	TRAO-RI_STAT	TRAO-RI_DYN	TRANSMIT_K	TRACYL_K	TRAANG_K
\$NT_NAME[n]	1200	x	x	x	x	x
\$NT_TRAFO_INDEX[n]	1295	x	x	x	x	x
\$NT_TRAFO_TYPE[n]	1201-1203	x	x	x	x	x
\$NT_T_CHAIN_LAST_ELEM[n]	1204	x	x	x	x	x
\$NT_P_CHAIN_LAST_ELEM[n]	1207	x	x	x	x	x
\$NT_T_CHAIN_FIRST_ELEM[n]	1203	x	x	x	x	x
\$NT_P_CHAIN_FIRST_ELEM[n]	1206	x	x	x	x	x

Variable	BTSS (Spaltenind.)	TRAO-RI_STAT	TRAO-RI_DYN	TRANS-MIT_K	TRA-CYL_K	TRAANG_K
\$NT_T_REF_ELEM[n]	1208	x	x	x	x	x
\$NT_GEO_AX_NAME[n,0..2]	1210-1222	0,1,2	0,1,2	0,1,2	0,1,2	0,1,2
\$NT_ROT_AX_NAME[n,0..2]	1220-1222	0,1,2	0,1,2	1	1	-
\$NT_CLOSE_CHAIN_T[n]	1226	x	x	x	x	x
\$NT_BASE_ORIENT[n,0..2]	1280-1285	x	x	-	-	-
\$NT_BASE_ORIENT_NORMAL[n,0..2]	1283-1285	x	x	-	-	-
\$NT_ROT_OFF-SET_FROM_FRAME[n]	1288	x	x	x	x	-
\$NT_POLE_LIMIT[n]	1286	-	x	-	-	-
\$NT_POLE_TOL[n]	1287	x	x	-	-	-
\$NT_IGNORE_TOOL_ORIENT[n]	1289	x	x	-	-	-
\$NT_ROT_AX_POS[n]	1230-1232	x	x	-	-	-
\$NT_CORR_ELEM_T[n,0..3]	1235-1238	x	x	-	-	-
\$NT_CORR_ELEM_P[n,0..3]	1320-1323	x	x	-	-	-
\$NT_TRAFO_INCLUDES_TOOL[n]	1290	-	-	-	-	x
\$NT_POLE_SIDE_FIX[n]	1291	-	-	x	-	-
\$NT_CNTRL[n]	1294	x	x	x	x	x
\$NT_AUX_POS[n,0..2]	1300-1302	~	~	~	~	~
\$NT_IDENT[n,0..2]	1310-1312	~	~	~	~	~
\$NT_ROT_AX_CNT[n,0..1]	1316-1317	x	x	x	x	x
\$NT_BASE_TOOL_COMP	1396	-	-	x	x	-
\$NT_ROT_AX_OFFSET[n,0...2]	1324-1326	x	x	x	x	-

"x" Wirkt sich auf die Transformation aus.

"-" hat keine Auswirkung auf die Transformation

"~" hat im NCK keine Auswirkung auf die Transformation, kann aber bestimmungsgemäß für Zwecke außerhalb von NC genutzt werden.

"Ziffern" zulässige Achsindizes (0 bis 2)

"Spaltenindex"; Spaltenindex der Variablen in der BTSS-Schnittstelle im Baustein N_PA (0x35).

9.2.4 Systemvariablen für allgemeine Transformationstypen

9.2.4.1 Übersicht

Systemvariable	Bedeutung
\$NT_NAME	Name des Transformationsdatensatzes
\$NT_TRAFO_INDEX	Kennung für alternative Trafoaktivierung

Systemvariable	Bedeutung
\$NT_TRAFO_TYPE	Trafotyp
\$NT_T_CHAIN_LAST_ELEM	Name des letzten Elements der Werkzeug-Kette
\$NT_P_CHAIN_LAST_ELEM	Name des letzten Elements der Werkstück-Kette
\$NT_T_CHAIN_FIRST_ELEM	Name des ersten Elements der Werkzeug-Kette
\$NT_P_CHAIN_FIRST_ELEM	Name des ersten Elements der Werkstück-Kette
\$NT_T_REF_ELEM	Name des Referenzelements für Werkzeuglänge
\$NT_GEO_AX_NAME	Namen der Elemente der Geometrieachsen
\$NT_ROT_AX_NAME	Namen der Elemente der Orientierungsachsen
\$NT_CLOSE_CHAIN_P	Referenzelement zum Schließen der kinematischen Kette zum Werkzeug
\$NT_CLOSE_CHAIN_T	Referenzelement zum Schließen der kinematischen Kette zum Werkstück
\$NT_ROT_OFF- SET_FROM_FRAME	Den Rundachsoffset bei Anwahl der Transformation aus der Nullpunktverschiebung übernehmen
\$NT_TRAFO_INCLUDES_TOOL	Legt fest, ob die Werkzeuglänge in der Transformation oder - falls die Transformation das zulässt - außerhalb der Transformation behandelt wird.
\$NT_AUX_POS	Hilfsposition für Zyklen, z. B. Freifahrposition Hinweis Ohne Bedeutung für die Transformation
\$NT_IDENT	Identifikationsnummer für anwenderspezifische Programmverwaltung Hinweis Ohne Bedeutung für die Transformation
\$NT_CORR_ELEM_T	Namen von maximal vier Korrektur-elementen in der kinematischen Kette zum Werkzeug
\$NT_CORR_ELEM_P	Namen von maximal vier Korrektur-elementen in der kinematischen Kette zum Werkstück
\$NT_CNTRL	Steuerwort

Die Systemvariablen sind in den nachfolgenden Kapiteln ausführlich beschrieben.

Hinweis

Namen von Elementen

Es wird nicht überwacht ob die Namen von Elementen der kinematischen Kette, auf die zur Parametrierung der Transformation referenziert wird, mehrfach vergeben wurden. Sind solche Namen mehrfach vorhanden, verweisen die Systemdaten der Transformation immer auf das entsprechende Element mit dem niedrigsten Index.

Definierten Ausgangszustand herstellen

Es wird empfohlen, vor Parametrierung der kinematischen Kette einen definierten Ausgangszustand zu erzeugen. Dazu sind die Systemvariablen der kinematischen Kette mit der Funktion DELOBJ() (Seite 225) auf ihren Defaultwert zu setzen.

Ändern von Systemvariablenwerten

Eine Änderung von Systemvariablenwerten der Transformationsdaten \$NT_... wird erst durch das Wirksamsetzen von Maschinendaten NEWCONF wirksam. Dies kann über den Softkey auf der Bedienoberfläche im Bedienbereich "Inbetriebnahme" > "Maschinendaten" > "MD wirksam setzen", in einem Programm durch den Befehl `NEWCONF` oder durch Programmende-Reset, Kanal-Reset oder Warm- oder Kaltstart erfolgen.

9.2.4.2 \$NT_NAME

Funktion

In die Systemvariable ist der NC-weit eindeutige Name des Transformationsdatensatzes bzw. der Transformation einzutragen. Über diesen Namen wird die Transformation z. B. bei der Aktivierung mit TRAF00N, referenziert. Ist als Name der Null-String ("") eingetragen, gilt die Transformation als nicht definiert.

Syntax

```
$NT_NAME [<n>] = "<Name>"
```

Bedeutung

\$NT_NAME:	Name der Transformation	
	Datentyp:	STRING
	Defaultwert:	"" (Leerstring)
<n>:	Systemvariablen- bzw. Transformations-Index	
	Datentyp:	INT
	Wertebereich:	1, 2, ... (MD18866 \$MN_MM_NUM_KIN_TRAFOS - 1)
<Name>:	Transformationsname, max. Stringlänge: 31 Zeichen	
	Datentyp:	STRING

Beispiel

Der ersten Transformation mit kinematischen Ketten wird der Name "5-Achs-Trafo C-B" zugewiesen:

Programmcode	Kommentar
N100 \$NT_NAME[1] = "5-Achs-Trafo C-B"	; 1. Transformation, ; Name = "5-Achs-Trafo C-B"

9.2.4.3 \$NT_TRAFO_INDEX

Funktion

Damit eine Transformation auch über den entsprechenden konventionellen Befehl, z. B. die Orientierungstransformation über TRAORI, aufgerufen werden kann, ist in die Systemvariable eine NC-weit, eindeutige Kennung einzutragen. Die Kennung kennzeichnet dabei mittels Transformationstyp, Transformationsnummer und Kanalnummer eine Transformation mit kinematischen Ketten auf die eine konventionell mit dieser Kennung aufgerufene Transformation (TRAORI (<Kennung>, . . .)) umgeleitet wird. Die Kennung ist dezimal codiert.

Syntax

\$NT_TRAFO_INDEX[<n>] = <Id>

Bedeutung

\$NT_TRAFO_INDEX:	Kennung, dezimal codiert (CCBBA)		
	Datentyp:	INT	
	Defaultwert:	0	
	Dezimalstelle	Bedeutung	
	Einerstelle (xxxxA)	Transformationstyp Der Typ der Transformation, die in diesem Datensatz parametrisiert ist:	
		Wert	Bedeutung
		1	TRAORI
		2	TRANSMIT
3		TRACYL	
4	TRAANG		
5	TRACON		
Wertebereich:	1, 2, 3, 4, 5		

\$NT_TRAFO_INDEX: (Fortsetzung)	Zehner- und Hunderterstelle (xxBBx)	Transformationsnummer Die Nummer <n>, die bei einem konventionellen Aufruf eines Transformationstyps z. B. Orientierungstransformation TRAORI (<n>), den n-ten kanal-spezifischen Maschinendatensatz dieses Typs (MD24100ff bzw. MD25100ff) referenziert. Bei einem konventionellen Aufruf der ersten Transformation eines Typs kann die Nummer angegeben werden (z. B. TRAORI, TRAORI (), TRAORI (0), TRAORI (1)). Alle Aufrufe referenzieren dabei den 1. Datensatz des Transformationstyps im Kanal. Ab dem Aufruf der zweiten Transformation eines Typs muss die Nummer angegeben werden (z. B. TRAORI (2)).
		Wertebereich:
\$NT_TRAFO_INDEX: (Fortsetzung)	Tausender- und Zahntausenderstelle (CCxxx)	Kanalnummer Über die Kanalnummer wird festgelegt, für welchen Kanal der aktuelle Datensatz gültig ist. Erfolgt der konventionelle Transformationsaufruf einer Orientierungstransformation TRAORI z. B. im zweiten Kanal, muss eine Transformation mit kinematischen Ketten mit Kanalnummer gleich 2 parametrierbar sein.
		Wertebereich:
<n>:	Systemvariablen- bzw. Transformations-Index	
	Datentyp:	INT
	Wertebereich:	1, 2, ... (MD18866 \$MN_MM_NUM_KIN_TRAFOS - 1)
<Id>:	Kennung	
	Datentyp:	INT

Beispiel

Der Aufruf der ersten Orientierungstransformation TRAORI (2) im dritten Kanal soll auf die zweite Transformation mit kinematischen Ketten umgeleitet werden.

Programmcode	Kommentar
N100 \$NT_TRAFO_INDEX[1] = 03021	; 1. Transformation mit kin. Ketten, ; 1. Orientierungstransformation für Kanal 3

9.2.4.4 \$NT_TRAFO_TYPE

Funktion

In die Systemvariable ist die Bezeichnung des Transformationstyps einzutragen.

Syntax

```
$NT_TRAFO_TYPE[<n>] = "<TrafoType>"
```

Bedeutung

\$NT_TRAFO_TYPE:	Transformationstyp		
	Datentyp:	STRING	
	Defaultwert:	""	
	Wertebereich:	Wert	Bedeutung
		"TRAORI_DYN"	Dynamische Orientierungstransformation mit Orientierungsachsen
		"TRAORI_STAT"	Statische Orientierungstransformation ohne Orientierungsachsen
		"TRAANG"	Schiefwinkeltransformation ohne Orientierungsachsen
"TRAANG_K"		Schiefwinkeltransformation (Die Erweiterung "_K" dient zur Unterscheidung von einer konventionell definierten TRAANG-Transformation)	
"TRANSMIT_K"	Stirnseitentransformation TRANSMIT		
"TRACON_K"	Transformationen verketten		
<n>:	Systemvariablen- bzw. Transformations-Index		
	Datentyp:	INT	
	Wertebereich:	1, 2, ... (MD18866 \$MN_MM_NUM_KIN_TRAFOS - 1)	
<TrafoType>:	Transformationstyp		
	Datentyp:	STRING	

Beispiel

Die erste Transformation mit kinematischen Ketten beschreibt eine dynamische Orientierungstransformation mit Orientierungsachsen:

Programmcode	Kommentar
N100 \$NT_TRAFO_TYPE[1] = "TRAORI_DYN"	; 1. Transformation, ; Transformationstyp = "TRAORI_DYN"

9.2.4.5 \$NT_T_CHAIN_FIRST_ELEM

Funktion

Die Kinematik einer Transformation wird durch maximal zwei kinematische Teilketten beschrieben, die jeweils im Root-Element (Seite 201), d. h. dem ersten Element der wirksamen kinematischen Kette, beginnen. Die eine Teilkette führt vom Root-Element zum **Werkstück**bezugspunkt (siehe Definition kinematischer Transformationen (Seite 328)), die andere zum **Werkzeug**bezugspunkt. Falls die Teilkette nicht, wie üblich im Root-Element beginnt, kann mit der Systemvariablen das erste Element der wirksamen kinematischen Teilkette jeweils für die Werkzeug- und Werkstück-Kette festgelegt werden

In die Systemvariable \$NT_T_CHAIN_FIRST_ELEM ist der Name (\$NK_NAME (Seite 209)) des Elements der aktuell wirksamen kinematischen Teilkette einzutragen, das den Startpunkt der Teilkette zum Werkzeugbezugspunkt definiert.

Hinweis

Reine Werkzeugkinematik

Die Angabe des Startelements in \$NT_P_CHAIN_FIRST_ELEM ist nur in Sonderfällen erforderlich, ansonsten kann die Systemvariable leer bleiben. Dann wird als erstes Element das ROOT-Element verwendet.

Syntax

```
$NT_T_CHAIN_FIRST_ELEM[<n>] = "<ElementName>"
```

Bedeutung

\$NT_T_CHAIN_FIRST_ELEM:	Name des Elementes, das den Startpunkt der aktuell wirksamen kinematischen Kette zum Werkzeugbezugspunkt definiert.	
	Datentyp:	STRING
	Defaultwert:	""
	Wertebereich:	Elementnamen der aktuell wirksamen kinematischen Kette
<n>:	Systemvariablen- bzw. Transformations-Index	
	Datentyp:	INT
	Wertebereich:	1, 2, ... (MD18866 \$MN_MM_NUM_KIN_TRAFOS - 1)
<ElementName>:	Name eines Elements der aktuell wirksamen kinematischen Kette	
	Datentyp:	STRING

Beispiel

Für die erste Transformation ist der Name des Elements der kinematischen Teilkette, das den Startpunkt der Teilkette zum Werkstückbezugspunkt definiert, "Baseoffset_2":

Programmcode	Kommentar
N100 \$NT_T_CHAIN_FIRST_ELEM[1] = "Baseoffset_2"	; 1. Transformation, ; Startelement für den Werkzeugbezugspunkt

9.2.4.6 \$NT_P_CHAIN_FIRST_ELEM

Funktion

Die Kinematik einer Transformation wird durch maximal zwei kinematische Teilketten beschrieben, die jeweils im Root-Element (Seite 201), d. h. dem ersten Element der wirksamen kinematischen Kette, beginnen. Die eine Teilkette führt vom Root-Element zum **Werkzeug**bezugspunkt (siehe Definition kinematischer Transformationen (Seite 328)), die andere, hier beschriebene Teilkette, zum **Werkstück**bezugspunkt. Falls die Teilkette nicht, wie üblich im Root-Element beginnt, kann mit der Systemvariablen das erste Element der wirksamen kinematischen Teilkette jeweils für die Tool- und Part-Kette festgelegt werden.

In die Systemvariable \$NT_P_CHAIN_FIRST_ELEM ist der Name (\$NK_NAME (Seite 209)) des Elements der aktuell wirksamen kinematischen Kette einzutragen, das den Startpunkt der Teilkette zum Werkstückbezugspunkt definiert.

Hinweis

Reine Werkstückkinematik

Die Angabe des Startelements in \$NT_T_CHAIN_FIRST_ELEM ist nur in Sonderfällen erforderlich, ansonsten kann die Systemvariable leer bleiben. Dann wird als erstes Element das ROOT-Element genommen.

Syntax

\$NT_P_CHAIN_FIRST_ELEM[<n>] = "<ElementName>"

Bedeutung

\$NT_P_CHAIN_FIRST_ELEM:	Name des Startelements der aktuell wirksamen kinematischen Teilkette zum Endpunkt des Werkzeugbezugspunkts definiert.	
	Datentyp:	STRING
	Defaultwert:	""
	Wertebereich:	Elementnamen der aktuell wirksamen kinematischen Kette
<n>:	Systemvariablen- bzw. Transformations-Index	
	Datentyp:	INT
	Wertebereich:	1, 2, ... (MD18866 \$MN_MM_NUM_KIN_TRAFOS - 1)
<ElementName>:	Name eines Elements der aktuell wirksamen kinematischen Kette	
	Datentyp:	STRING

Beispiel

Für die erste Transformation ist der Name des Elements der kinematischen Kette, das den Endpunkt der Teilkette zum Werkstückbezugspunkt definiert, "TableOffset":

Programmcode	Kommentar
N100 \$NT_P_CHAIN_FIRST_ELEM[1] = "Base- offset"	; 2. Transformation, ; Element für den Werkstückbezugspunkt

9.2.4.7 \$NT_T_CHAIN_LAST_ELEM

Funktion

Die Kinematik einer Transformation wird durch maximal zwei kinematische Teilketten beschrieben, die jeweils im Root-Element (Seite 201), d. h. dem ersten Element der wirksamen kinematischen Kette, beginnen. Die eine Teilkette führt vom Root-Element zum **Werkstückbezugspunkt** (siehe Definition kinematischer Transformationen (Seite 328)), die andere, hier beschriebene Teilkette, zum **Werkzeugbezugspunkt**.

In die Systemvariable \$NT_T_CHAIN_LAST_ELEM ist der Name (\$NT_NAME (Seite 351)) des Elements der aktuell wirksamen kinematischen Kette einzutragen, das den Endpunkt der Teilketten zum Werkzeugbezugspunkt definiert.

Hinweis

Reine Werkzeugkinematik

Wird bei einer reinen Werkzeugkinematik die Kinematik der Transformation durch die Teilkette zum Werkzeugbezugspunkt vollständig beschrieben, ist das Parametrieren der Systemvariable \$NT_T_CHAIN_LAST_ELEM ausreichend. Die Systemvariable zur Beschreibung der Teilkette zum Werkstückbezugspunkt (\$NT_P_CHAIN_LAST_ELEM) kann leer bleiben.

Syntax

```
$NT_T_CHAIN_LAST_ELEM[<n>] = "<ElementName>"
```

Bedeutung

\$NT_T_CHAIN_LAST_ELEM:	Name des Elements der aktuell wirksamen kinematischen Kette, das, ausgehend vom Root-Element, den Endpunkt zum Werkzeugbezugspunkt definiert.	
	Datentyp:	STRING
	Defaultwert:	""
	Wertebereich:	Elementnamen der aktuell wirksamen kinematischen Kette
<n>:	Systemvariablen- bzw. Transformations-Index	
	Datentyp:	INT
	Wertebereich:	1, 2, ... (MD18866 \$MN_MM_NUM_KIN_TRAFOS - 1)

<ElementName>:	Name eines Elements der aktuell wirksamen kinematischen Kette	
	Datentyp:	STRING

Beispiel

Für die erste Transformation ist der Name des Elements der kinematischen Kette, das den Endpunkt der Teilkette zum Werkzeugbezugspunkt definiert, "Basetool":

Programmcode	Kommentar
N100 \$NT_T_CHAIN_LAST_ELEM[1] = "Base-tool"	; 1. Transformation, ; Element für den Werkzeugbezugspunkt

9.2.4.8 \$NT_P_CHAIN_LAST_ELEM

Funktion

Die Kinematik einer Transformation wird durch maximal zwei kinematische Teilketten beschrieben, die jeweils im Root-Element (Seite 201), d. h. dem ersten Element der wirksamen kinematischen Kette, beginnen. Die eine Teilketten führt vom Root-Element zum **Werkzeugbezugspunkt** (siehe Definition kinematischer Transformationen (Seite 328)), die andere, hier beschriebene Teilkette, zum **Werkstückbezugspunkt**.

In die Systemvariable \$NT_P_CHAIN_LAST_ELEM ist der Name (\$NT_NAME (Seite 351)) des Elements der aktuell wirksamen kinematischen Kette einzutragen, das den Endpunkt der Teilketten zum Werkstückbezugspunkt definiert.

Hinweis

Reine Werkstückkinematik

Wird bei einer reinen Werkstückkinematik die Kinematik der Transformation durch die Teilkette zum Werkstückbezugspunkt vollständig beschrieben, ist das Parametrieren der Systemvariable \$NT_P_CHAIN_LAST_ELEM ausreichend. Die Systemvariable zur Beschreibung der Teilkette zum Werkzeugbezugspunkt (\$NT_T_CHAIN_LAST_ELEM) kann leer bleiben.

Syntax

\$NT_P_CHAIN_LAST_ELEM[<n>] = "<ElementName>"

Bedeutung

\$NT_P_CHAIN_LAST_ELEM:	Name des Elements der aktuell wirksamen kinematischen Kette, das, ausgehend vom Root-Element, den Endpunkt zum Werkstückbezugspunkt definiert	
	Datentyp:	STRING
	Defaultwert:	""
	Wertebereich:	Elementnamen der aktuell wirksamen kinematischen Kette
<n>:	Systemvariablen- bzw. Transformations-Index	
	Datentyp:	INT
	Wertebereich:	1, 2, ... (MD18866 \$MN_MM_NUM_KIN_TRAFOS - 1)
<ElementName>:	Name eines Elements der aktuell wirksamen kinematischen Kette	
	Datentyp:	STRING

Beispiel

Für die erste Transformation ist der Name des Elements der kinematischen Kette, das den Endpunkt der Teilkette zum Werkstückbezugspunkt definiert, "TableOffset":

Programmcode	Kommentar
N100 \$NT_P_CHAIN_LAST_ELEM[1] = "Table-Offset"	; 2. Transformation, ; Element für den Werkstückbezugspunkt

9.2.4.9 \$NT_T_REF_ELEM

Funktion

In die Systemvariable ist der Name (\$NT_NAME (Seite 351)) des Elements der aktuell wirksamen kinematischen Kette einzutragen, das den Werkzeugreferenzpunkt, d. h. den Referenzpunkt zur Werkzeuglängenberechnung, definiert. Der Werkzeugreferenzpunkt liegt am **Anfangspunkt** dieses kinematischen Elements.

Wird in die Systemvariable kein Name eines Elements eingetragen, ist der Werkzeugreferenzpunkt identisch mit dem Werkzeugbezugspunkt.

Syntax

```
$NT_T_REF_ELEM[<n>] = "<RefElementName>"
```

Bedeutung

\$NT_T_REF_ELEM:	Name des Elements der aktuell wirksamen kinematischen Kette, das den Werkzeugreferenzpunkt definiert	
	Datentyp:	STRING
	Defaultwert:	"" (Werkzeugreferenzpunkt = Werkzeugbezugspunkt)
	Wertebereich:	Elementnamen der aktuell wirksamen kinematischen Kette
<n>:	Systemvariablen- bzw. Transformations-Index	
	Datentyp:	INT
	Wertebereich:	1, 2, ... (MD18866 \$MN_MM_NUM_KIN_TRAFOS - 1)
<RefElementName>:	Name eines Elements der aktuell wirksamen kinematischen Kette	
	Datentyp:	STRING

Beispiel

Für die erste Transformation ist der Name des Elements der aktuell wirksamen kinematischen Kette, das den Werkzeugreferenzpunkt definiert, "ToolRefPoint":

Programmcode	Kommentar
N100 \$NT_T_REF_ELEM[1] = "ToolRef-Point"	; 1. Transformation, ; Element für den Werkzeugreferenzpunkt

9.2.4.10 \$NT_GEO_AX_NAME

Funktion

In die Systemvariable sind die maximal drei Namen (\$NT_NAME (Seite 351)) der Elemente der aktuell wirksamen kinematischen Kette einzutragen, die die Geometrieachsen (Linearachsen) des zu parametrierenden Transformationstyps (Seite 328) definieren. Über die Geometrieachsen erfolgen die linearen Ausgleichsbewegungen die sich aus der kinematischen Transformation ergeben.

Syntax

\$NT_GEO_AX_NAME [<n>,<k>] = "<GeoAxElementName>"

Bedeutung

\$NT_GEO_AX_NAME:	Namen der Elemente der aktuell wirksamen kinematischen Kette, die die Geometrieachsen definieren	
	Datentyp:	STRING
	Defaultwert:	("", "", "")
	Wertebereich:	Elementnamen der aktuell wirksamen kinematischen Kette

<n>:	Systemvariablen- bzw. Transformations-Index	
	Datentyp:	INT
	Wertebereich:	1, 2, ... (MD18866 \$MN_MM_NUM_KIN_TRAFOS - 1)
<k>:	Index für die Elemente die die Geometrieachsen definieren	
	Datentyp:	INT
	Wertebereich:	0: X-Koordinate (Abszisse) 1: Y-Koordinate (Ordinate) 2: Z-Koordinate (Applikate)
<GeoAxElementName>:	Name eines Elements der aktuell wirksamen kinematischen Kette, das eine Geometrieachse definiert	
	Datentyp:	STRING

Beispiel

Für die erste Transformation sind die Namen der Elemente der aktuell wirksamen kinematischen Kette, die die Geometrieachsen "X_AXIS", "Y_AXIS" und "Z_AXIS" definieren, einzutragen.

Programmcode	Kommentar
	; 1. Transformation,
	; Elemente der Geometrieachsen:
N100 \$NT_GEO_AX_NAME[1,0] = "X_AXIS"	; X-Koordinate (Abszisse)
N110 \$NT_GEO_AX_NAME[1,1] = "Y_AXIS"	; Y-Koordinate (Ordinate)
N120 \$NT_GEO_AX_NAME[1,2] = "Z_AXIS"	; Z-Koordinate (Applikate)

9.2.4.11 \$NT_ROT_AX_NAME

Funktion

In die Systemvariable sind die maximal drei Namen (\$NT_NAME (Seite 351)) der Elemente der aktuell wirksamen kinematischen Kette einzutragen, die die Rundachsen des zu parametrierenden Transformationstyps (Seite 328) definieren:

- Orientierungstransformation: Die Orientierungsachsen A, B, C
- Stirnseiten- (TRANSMIT) und Zylindermanteltransformation (TRACYL): Die Rundachse, die das Werkstück dreht und zusammen mit einer Linearachse das Polarkoordinatensystem definiert.

Definitionsreihenfolge

Die maximal drei Rundachsen müssen, beginnend beim Index 0, in der Reihenfolge in die Systemvariablen eingetragen werden, in der sie in der aktuell wirksamen kinematischen Kette definiert sind.

Dabei ist die kinematischen Kette folgendermaßen zu durchlaufen:

1. Startpunkt: **Werkstück**bezugspunkt
2. Elemente der Kette
3. Endpunkt: **Werkzeug**bezugspunkt

Die Elemente, die die Rundachsen definieren, müssen mit ihren Namen lückenlos, beginnend beim Index 0, in die Systemvariable eingetragen werden.

Syntax

```
$NT_ROT_AX_NAME [<n>,<k>] = "<RotAxElementName>"
```

Bedeutung

\$NT_ROT_AX_NAME:	Namen der Elemente der aktuell wirksamen kinematischen Kette, die die Rundachsen definieren	
	Datentyp:	STRING
	Defaultwert:	("", "", "")
	Wertebereich:	Elementnamen der aktuell wirksamen kinematischen Kette
<n>:	Systemvariablen- bzw. Transformations-Index	
	Datentyp:	INT
	Wertebereich:	1, 2, ... (MD18866 \$MN_MM_NUM_KIN_TRAFOS - 1)
<k>:	Index für die Elemente die die Rundachsen definieren	
	Datentyp:	INT
	Wertebereich:	0: 1. Rundachse in der Definitionsreihenfolge 1: 2. Rundachse in der Definitionsreihenfolge 2: 3. Rundachse in der Definitionsreihenfolge
<RotAxElementName>:	Name eines Elements der aktuell wirksamen kinematischen Kette, das eine Rundachse definiert	
	Datentyp:	STRING

Beispiel

Für die erste Transformation sind die Namen der Elemente der aktuell wirksamen kinematischen Kette, die die Rundachsen "ROT_AXIS_1", "ROT_AXIS_2" und "ROT_AXIS_3" definieren .

Programmcode	Kommentar
	; 1. Transformation,
	; Elemente der Rundachsen:
N100 \$NT_ROT_AX_NAME[1,0] = "ROT_AXIS_1"	; 1. Rundachse
N110 \$NT_ROT_AX_NAME[1,1] = "ROT_AXIS_2"	; 2. Rundachse
N120 \$NT_ROT_AX_NAME[1,2] = "ROT_AXIS_3"	; 3. Rundachse

9.2.4.12 \$NT_ROT_AX_OFFSET

Funktion

Die Systemvariable \$NT_ROT_AX_OFFSET erlaubt es ein Winkeloffset für die Rundachsen der aktiven Transformation einzugeben.

- Für TRAORI_K sind es die Rundachsen 1, 2 und 3.
- Für TRANSMIT_K und TRACYL_K ist es die Rundachse 1.

Randbedingungen

- Die Systemvariable \$NT_ROT_AX_OFFSET entspricht bei Transmit- bzw. Tracyl-Transformationen ohne kinematische Kette den Maschinendaten \$MC_TRANSMIT_ROT_AX_OFFSET_n bzw. \$MC_TRACYL_ROT_AX_OFFSET_n.
- Die Systemvariable \$NT_ROT_AX_OFFSET entspricht bei Orientierungstransformationen ohne kinematische Kette dem Maschinendatum \$MC_TRAFO5_ROT_AX_OFFSET_n.

Syntax

```
$NT_ROT_AX_OFFSET[<n,m>] = <Offset_Angle>
```

Bedeutung

\$NT_ROT_AX_OFFSET	Winkeloffset für Rundachsen	
	Datentyp:	REAL
	Defaultwert:	-1; Inhalt wird nicht ausgewertet
<n>:	Systemvariablen- bzw. Transformations-Index	
	Datentyp:	INT
	Wertebereich:	1, 2, ... (MD18866 \$MN_MM_NUM_KIN_TRAFOS - 1)
<m>:	Index der Variable \$NT_ROT_AX_CNT; gibt an, für welche Rundachse das Offset gesetzt wird.	
	Datentyp:	INT
	Wertebereich	<ul style="list-style-type: none"> • 0 ... 2; für Orientierungstransformation • 1; für TRANSMIT_K und TRACYL_K

Beispiel

Liefert die Anzahl der Rundachsen für die Partkette:

Programmcode	Kommentar
N100 \$NT_ROT_AX_CNT[1,0] = 180	; Weist der Rundachse "0" der Transformation "1" das Offset "180°" zu.

9.2.4.13 \$NT_CLOSE_CHAIN_P

Funktion

In die Systemvariable kann der Name (\$NT_NAME (Seite 351)) des Elements der aktuell wirksamen kinematischen Kette eingetragen werden, an dessen Ende die Teilkette zum **Werkstück**bezugspunkt geschlossen wird, wenn \$NT_CNTRL, Bit 7 == 1.

Ist in die Systemvariable kein Name eingetragen, wird die Teilkette, unabhängig von \$NT_CNTRL, Bit 7, am Ende des letzten Elements (**Werkstück**bezugspunkt) geschlossen.

Syntax

\$NT_CLOSE_CHAIN_P[<n>] = "<ElementName>"

Bedeutung

\$NT_CLOSE_CHAIN_P:	Name des Elements der aktuell wirksamen kinematischen Kette, an dessen Ende die Teilkette zum Werkstück bezugspunkt geschlossen wird	
	Datentyp:	STRING
	Defaultwert:	""
	Wertebereich:	Elementnamen der aktuell wirksamen kinematischen Kette
<n>:	Systemvariablen- bzw. Transformations-Index	
	Datentyp:	INT
	Wertebereich:	1, 2, ... (MD18866 \$MN_MM_NUM_KIN_TRAFOS - 1)
<ElementName>:	Name eines Elements der aktuell wirksamen kinematischen Kette	
	Datentyp:	STRING

Beispiel

Für die erste Transformation ist der Name des Elements der kinematischen Kette an dessen Ende die Teilkette zum **Werkstück**bezugspunkt geschlossen wird: "P_Offset_3"

Programmcode	Kommentar
N100 \$NT_CLOSE_CHAIN_P[1] = "P_Offset_3"	; 1. Transformation, ; Element zum Schließen der Teilkette ; zum Werkstück bezugspunkt

Siehe auch

\$NT_CNTRL (Seite 370)

9.2.4.14 \$NT_CLOSE_CHAIN_T

Funktion

In die Systemvariable kann der Name (\$NT_NAME (Seite 351)) des Elements der aktuell wirksamen kinematischen Kette eingetragen werden, an dessen Ende die Teilkette zum **Werkzeug**bezugspunkt geschlossen wird, wenn \$NT_CNTRL, Bit 8 == 1 (\$NT_CNTRL (Seite 370)).

Ist in die Systemvariable kein Name eingetragen wird die Teilkette, unabhängig von \$NT_CNTRL, Bit 8, am Ende des letzten Elements (**Werkzeug**bezugspunkt) geschlossen.

Syntax

```
$NT_CLOSE_CHAIN_T[<n>] = "<ElementName>"
```

Bedeutung

\$NT_CLOSE_CHAIN_T:	Name des Elements der aktuell wirksamen kinematischen Kette, an dessen Ende die Teilkette zum Werkzeug bezugspunkt geschlossen wird	
	Datentyp:	STRING
	Defaultwert:	""
	Wertebereich:	Elementnamen der aktuell wirksamen kinematischen Kette
<n>:	Systemvariablen- bzw. Transformations-Index	
	Datentyp:	INT
	Wertebereich:	1, 2, ... (MD18866 \$MN_MM_NUM_KIN_TRAFOS - 1)
<ElementName>:	Name eines Elements der aktuell wirksamen kinematischen Kette	
	Datentyp:	STRING

Beispiel

Für die erste Transformation ist der Name des Elements der kinematischen Kette an dessen Ende die Teilkette zum **Werkzeug**bezugspunkt geschlossen wird: "T_Offset_3"

Programmcode	Kommentar
N100 \$NT_CLOSE_CHAIN_T[1] = "T_Offset_3"	; 1. Transformation, ; Element zum Schließen der Teilkette ; zum Werkzeug bezugspunkt

9.2.4.15 \$NT_ROT_OFFSET_FROM_FRAME

Funktion

In die Systemvariable ist \$NT_ROT_OFFSET_FROM_FRAME ist einzutragen, ob der programmierbare Offset für Orientierungsachsen automatisch aus der bei Einschalten einer Orientierungstransformation für die Orientierungsachsen aktiven Nullpunktverschiebung übernommen wird.

- TRAORI bzw. TRAORY_DYN:
 - Bei \$NT_ROT_OFFSET_FROM_FRAME=1 gilt:
Die Offsets für die Orientierungsachsen der 5/6-Achs Transformation der Rundachsen, bei denen die Werkzeugorientierung sich in Grundstellung befindet, werden automatisch festgelegt. Der Offset wird somit aus einer aktiven Nullpunktverschiebung der Rundachsen, die beim Einschalten der Transformation aktiv ist, übernommen.
 - Bei \$NT_ROT_OFFSET_FROM_FRAME=0 gilt:
Die Übernahme in den Offset der Orientierungsachsen erfolgt nur, falls die Nullpunktverschiebungen beim Einschalten der Transformation erhalten bleiben, d.h. das MD10602 \$MN_FRAME_GEOAX_CHANGE_MODE > 0 ist.
- TRACYL bzw. TRANSMIT
 - Bei \$NT_ROT_OFFSET_FROM_FRAME=2:
Die axiale Verschiebung der Rundachse wird bis zum ENS berücksichtigt. ENS entspricht dem Einstellbaren Nullpunktsystem. Die ENS-Frames enthalten transformierte Drehungen um die Rundachse.

Hinweis

Automatische Übernahme von Nullpunktverschiebungen

Die automatische Übernahme von aktiven Nullpunktverschiebungen in den Offset ist im Normalfall nur für die Polachse bei Tischkinematiken sinnvoll. D.h. zum Beispiel bei einer AC-Kinematik für die C-Achse. Falls bei einer anderen Orientierungsachse, insbesondere bei der Nichtpolachse, der Offset verändert wird, ändert sich dadurch das kinematische Verhalten der Transformation. Das kann dazu führen, dass die Ausgleichsbewegungen der Linearachsen nicht mehr korrekt sind.

Syntax

\$NT_ROT_OFFSET_FROM_FRAME [<n>] = <ROTOffset>

Bedeutung

\$NT_ROT_OFFSET_FROM_FRAME:	Rundachsoffset bei Anwahl der Transformation aus der Nullpunktverschiebung übernehmen:	
	<ul style="list-style-type: none"> • 0 = axiale Verschiebung der Rundachse wird nicht berücksichtigt. • 1 = axiale Verschiebung der Rundachse wird berücksichtigt. • 2 = axiale Verschiebung der Rundachse wird bis zum ENS bzw. automatisch übernommen (siehe oben). 	
	Datentyp:	INT
	Defaultwert:	0
<n>:	Systemvariablen- bzw. Transformations-Index	
	Datentyp:	INT
	Wertebereich:	0, 1, 2
ROffset	Rundachsoffset	

Beispiel

Der 2. Transformation wird der Name "B-Achse" zugewiesen:

Programmcode	Kommentar
N100 \$NT_ROT_OFFSET_FROM_FRAME[1] = 1	; axiale Verschiebung der Rundachse wird berücksichtigt.

9.2.4.16 \$NT_TRAFO_INCLUDES_TOOL**Funktion**

In die Systemvariable ist einzutragen, ob die Werkzeuglänge in der Transformation oder außerhalb der Transformation behandelt wird.

Syntax

```
$NT_TRAFO_INCLUDES_TOOL[<n>] = "<>"
```

Bedeutung

\$NT_TRAFO_INCLUDES_TOOL:	Legt fest, ob die Werkzeuglänge innerhalb oder außerhalb der Transformation behandelt wird.	
	Datentyp:	BOOL
		<ul style="list-style-type: none"> • 0: Werkzeuglänge wird außerhalb der Transformation behandelt. • 1: Werkzeuglänge wird innerhalb der Transformation behandelt.
	Defaultwert:	0

9.2 Inbetriebnahme

<n>:	Systemvariablen- bzw. Transformations-Index	
	Datentyp:	INT
	Wertebereich:	1, 2, ... (MD18866 \$MN_MM_NUM_KIN_TRAFOS - 1)

Beispiel

Der 2. Transformation wird der Name "B-Achse" zugewiesen:

Programmcode	Kommentar
N100 \$NT_TRAFO_INCLUDE_TOOL[1] = "1"	; 1. Transformation, ; Werkzeuglänge wird innerhalb der Transformation bearbeitet.

9.2.4.17 \$NT_AUX_POS

Funktion

In die Systemvariable kann ein Positionsvektor, z.B. zur Verwendung in anwenderspezifischen Zyklen, eingetragen werden. Die Systemvariable wird in der NC nicht ausgewertet. Die Werte werden aber bei einer Inch/Metric-Umschaltung automatisch umgerechnet.

Syntax

\$NT_AUX_POS[<n>,<k>] = <Value>

Bedeutung

\$NT_AUX_POS:	Anwenderspezifischer Positionsvektor	
	Datentyp:	REAL
	Defaultwert:	(0.0, 0.0, 0.0)
<n>:	Systemvariablen- bzw. Transformations-Index	
	Datentyp:	INT
	Wertebereich:	1, 2, ... (MD18866 \$MN_MM_NUM_KIN_TRAFOS - 1)
<k>:	Index der Vektor-Koordinaten	
	Datentyp:	INT
	Wertebereich:	0: X-Koordinate (Abszisse) 1: Y-Koordinate (Ordinate) 2: Z-Koordinate (Applikate)
<Value>:	Koordinatenwert	
	Datentyp:	REAL

Beispiel

Für die erste Transformation wird ein Positionsvektor (1.0, 1.0, 1.0) eingetragen:

Programmcode	Kommentar
	; 1. Transformation,
	; Positionsvektor:
N100 \$NT_AUX_POS[1,0] = 1.0	; X-Koordinate (Abszisse)
N110 \$NT_AUX_POS[1,1] = 1.0	; Y-Koordinate (Ordinate)
N120 \$NT_AUX_POS[1,2] = 1.0	; Z-Koordinate (Applikate)

9.2.4.18 \$NT_IDENT**Funktion**

In die Systemvariable können bis zu drei anwenderspezifische numerische Werte, analog zur Systemvariablen \$TC_CARR37, als Kennung für den Werkzeugträger oder die aktuelle Transformation für eine Programmverwaltung, eingetragen werden. Die Systemvariable wird in der NC nicht ausgewertet.

Syntax

\$NT_IDENT [<n>, <k>] = <Value>

Bedeutung

\$NT_IDENT:	Anwenderspezifische Verwaltungsdaten	
	Datentyp:	INT
	Defaultwert:	(0, 0, 0)
<n>:	Systemvariablen- bzw. Transformations-Index	
	Datentyp:	INT
	Wertebereich:	1, 2, ... (MD18866 \$MN_MM_NUM_KIN_TRAFOS - 1)
<k>:	Index der Verwaltungsdaten	
	Datentyp:	INT
	Wertebereich:	0, 1, 2
<Value>:	Anwenderspezifischer Wert	
	Datentyp:	INT

Beispiel

Für die erste Transformation werden folgende Verwaltungsdaten eingetragen: (1000, 100, 0)

Programmcode	Kommentar
	; 1. Transformation,
	; Positionsvektor:
N100 \$NT_AUX_POS[1,0] = 1000	; Kennung: Werkzeugträger
N110 \$NT_AUX_POS[1,1] = 100	; Kennung: Transformation
N120 \$NT_AUX_POS[1,2] = 0	; -

9.2.4.19 \$NT_CNTRL

Funktion

Mit dieser Systemvariable kann bitcodiert das Verhalten der Transformation in bestimmten Situationen beeinflusst werden.

Bit	Wert	Bedeutung
0	---	Nicht belegt
1 - 3		Orientierungsachsen als drehzahlgesteuerte Spindeln Bit 1: 1. Orientierungsachse Bit 2: 2. Orientierungsachse Bit 3: 3. Orientierungsachse
	0	Die Orientierungsachse wird als drehzahlgeregelte Achse betrieben.
	1	Die Orientierungsachse wird als drehzahlgesteuerte Spindel betrieben. Hinweis Aktuell werden folgende Fälle unterstützt: <ul style="list-style-type: none"> • nur die 1. Orientierungsachse (Bit 1 = 1) wird als drehzahlgesteuerte Spindel betrieben • nur die 3. Orientierungsachse (Bit 3 = 1) wird als drehzahlgesteuerte Spindel betrieben Beispiele sind die Anwendungen "Drehen auf Fräsmaschinen" bzw. "5-Achs-Fräsen", bei denen die dritte Orientierungsachse nicht lagegeregelt betrieben wird.

Bit	Wert	Bedeutung
4 - 6		Orientierungsachsen mit Hirth-Verzahnung Bit 4: 1. Orientierungsachse Bit 5: 2. Orientierungsachse Bit 6: 3. Orientierungsachse
	0	Die Orientierungsachse ist nicht Hirth-verzahnt.
	1	Die Orientierungsachse ist Hirth-verzahnt. Hinweis <ul style="list-style-type: none"> • Parameter der Hirthverzahnung Für die Parameter der Hirthverzahnung werden folgende Maschinendaten ausgewertet: <ul style="list-style-type: none"> – MD30501 \$MA_INDEX_AX_NUMERATOR – MD30502 \$MA_INDEX_AX_DENOMINATOR – MD30503 \$MA_INDEX_AX_OFFSET MD30505 \$MA_HIRTH_IS_ACTIVE wird nicht ausgewertet. D.h. die Achse muss nicht als echte Hirthachse parametrisiert sein. • Modulorundachsen Bei Modulorundachsen (MD30310 \$MA_ROT_IS_MODULO==TRUE), wird statt MD30501 \$MA_INDEX_AX_NUMERATOR, MD30330 \$MA_MODULO_RANGE ausgewertet. Die Abstände der zulässigen Achspositionen berechnen sich dann zu: Abstand = \$MA_MODULO_RANGE / \$MA_INDEX_AX_DENOMINATOR MD30503 \$MA_INDEX_AX_OFFSET wird auch bei Modulorundachsen ausgewertet
7		In die Teilkette zum Werkstück bezugspunkt wird NC-intern automatisch ein zusätzliches konstantes Element vom Typ "OFFSET" so eingefügt, dass die Kette geschlossen ist.
	0	Das Element verbindet den Werkstück bezugspunkt und mit dem Maschinennullpunkt.
	1	Das Element verbindet den Endpunkt des durch \$NT_CLOSE_CHAIN_P (Seite 364) referenzierten Elements mit dem Maschinennullpunkt.
8		In die Teilkette zum Werkzeug bezugspunkt wird NC-intern automatisch ein zusätzliches konstantes Element vom Typ "OFFSET" so eingefügt, dass die Kette geschlossen ist.
	0	Das Element verbindet den Werkzeug bezugspunkt und mit dem Maschinennullpunkt.
	1	Das Element verbindet den Endpunkt des durch \$NT_CLOSE_CHAIN_T (Seite 365) referenzierten Elements mit dem Maschinennullpunkt.
9		Der Funktionsumfang von TRANSMIT- bzw. TRACYL-Transformationen wird genauer spezifiziert. Wurde eine Mitterversatzachse über \$NT_ROT_AX_NAME spezifiziert, kann das Verhalten eingestellt werden.
	0	Die Mitterversatzachse kann beliebig interpoliert werden und hat keinen Einfluss auf die Transformation.
	1	Die Mitterversatzachse kompensiert den Werkzeugversatz in Y-Richtung. Eine Bearbeitung ist bis zur Drehmitte möglich.

Bit	Wert	Bedeutung
10	0	Parametriert TRACYL ohne Nutwandkorrektur.
	1	Parametriert TRACYL mit Nutwandkorrektur. In Verbindung mit Bit 9 = 1 kann bei der Aktivierung der Transformation über TRAFOON eingestellt werden, ob TRACYL mit oder ohne Nutwandkorrektur betrieben wird. Nutwandkorrektur aktiv ist die Standardeinstellung. Falls mit Nutwandkorrektur gearbeitet wird, muss \$NT_GEO_AX_NAME einen Verweis auf eine Linearachse (Nutwandkorrekturachse) enthalten.
11	0	Drehrichtung der Polachse bleibt unverändert.
	1	Drehrichtung der Polachse wird invertiert.
12 ... 15		Reserviert für OEM-Transformationen.
16 ... 18		Über die Bits 16, 17 und 18 (Bits H10000 - H40000) wird binär kodiert, wie die Kanalachsen, die in \$NT_GEO_AX_NAME [n,1], \$NT_GEO_AX_NAME [n,2] und \$NT_GEO_AX_NAME [n,3] eingetragen sind, den Geometrieachsen zugeordnet sind. Die Geometrieachsbezeichner sind in der Reihenfolge (X, Y, Z) vergeben.
	Dezimal	Reihenfolge der Geometrieachsen
	0	X Y Z
	1	Z X Y
	2	Y Z X
	3	Z Y X
	4	X Z Y
5	Y X Z	
19	0	
	1	Das letzte kinematische Kettenelement, das den Werkzeugbezugspunkt definiert, muss eine Rundachse sein. Der Richtungsvektor der Rundachse definiert dann die Z-Richtung des Werkzeugkoordinatensystems. Die Drehung um die so definierte Werkzeug-Z-Achse ergibt sich dabei aus der entsprechenden Definition für das lokale Koordinatensystem einer Achse in kinematischen Ketten. Enthält die Systemvariable \$NK_A_OFF dieses Kettenelements einen Wert ungleich Null, wird das Werkzeugkoordinatensystem zusätzlich mit diesem Winkel um die Koordinatenachse gedreht.
20	0	Das Vorzeichen der Achse für die Ermittlung des Werkzeugkoordinatensystems in Z-Richtung bleibt erhalten.
	1	Das Vorzeichen der Achse für die Ermittlung des Werkzeugkoordinatensystems in Z-Richtung wird invertiert, wenn auch Bit 19 gesetzt ist.

Syntax

\$NT_CNTRL[<n>] = "<Value>"

Bedeutung

\$NT_CNTRL:	Bitcodiert Steuervariable der Transformation ('Bdcbbaaa0')	
	Datentyp:	INT
	Defaultwert:	0

<n>:	Systemvariablen- bzw. Transformations-Index	
	Datentyp:	INT
	Wertebereich:	1, 2, ... (MD18866 \$MN_MM_NUM_KIN_TRAFOS - 1)
<Value>:	Bitcodiert Steuerwert	
	Datentyp:	INT

Beispiel

Die dritte Orientierungsachse der ersten Transformation ist Hirth-verzahnt.

Programmcode	Kommentar
N100 \$NT_CNTRL[1] = 'B001000000'	; 1. Transformation,

9.2.4.20 \$NT_ROT_AX_CNT

Funktion

Die Systemvariable `$NT_ROT_AX_CNT` liefert die Anzahl der relevanten Rundachsen in der Werkstück- bzw. in der Werkzeug-Kette. Relevante Rundachsen in diesem Sinne sind die Rundachsen, die in der Systemvariable `$NT_ROT_AX_Name` definiert sind.

Syntax

`$NT_ROT_AX_CNT[<n,m>] = <NumberRotAxes>`

Bedeutung

\$NT_ROT_AX_CNT:	Anzahl der relevanten Rundachsen einer Transformation in der Werkstück-Kette bzw. Werkzeug-Kette	
	Datentyp:	INT
	Defaultwert:	-1; Inhalt wird nicht ausgewertet
<n>:	Systemvariablen- bzw. Transformations-Index	
	Datentyp:	INT
	Wertebereich:	1, 2, ... (MD18866 \$MN_MM_NUM_KIN_TRAFOS - 1)
<m>:	Index der Variable <code>\$NT_ROT_AX_CNT</code> ; gibt an welche kinematische Kette ausgewählt wird.	
	Datentyp:	INT
	Wertebereich	<ul style="list-style-type: none"> • 0; wählt Werkstück-Kette an • 1; wählt Werkzeug-Kette an

Beispiel

Liefert die Anzahl der Rundachsen für die Parkette:

Programmcode	Kommentar
N100 \$NT_ROT_AX_CNT[1,0] = 2	; Die Transformation 1 verfügt über 2 Rundachsen.

9.2.4.21 \$NT_BASE_TOOL_COMP

Funktion

Mit der bitcodierten Systemvariable \$NT_BASE_TOOL_COMP kann für jede der Geometrieachsen getrennt eingestellt werden, ob für das BaseTool ein Offset in den Transformationsframe \$P_TRAFRAME eingetragen wird, so dass sich bei der Anwahl der Transformation keine Änderung in der WKS-Komponente ergibt.

Diese Funktion ist nur verfügbar, wenn das Systemframe \$P_TRAFRAME über MD28082 \$MC_MM_SYSTEM_FRAME_MASK Bit6 projiziert worden ist.

Die Systemvariable ist nur für TRANSMIT und TRACYL relevant.

Syntax

\$NT_BASE_TOOL_COMP [<n>, <m>]

Bedeutung

\$NT_BASE_TOOL_COMP:	Bitkodierte Systemvariable	
	Datentyp:	INT
	Defaultwert:	
<n>:	Systemvariablen- bzw. Transformations-Index	
	Datentyp:	INT
	Wertebereich:	1, 2, ... (MD18866 \$MN_MM_NUM_KIN_TRAFOS - 1)
<m>:	Bits für das Codieren der Offset:	
	<ul style="list-style-type: none"> • Bit 0 wirkt in Richtung X-Achse (Geometrieachse 1) • Bit 1 wirkt in Richtung Y-Achse (Geometrieachse 2) • Bit 2 wirkt in Richtung Z-Achse (Geometrieachse 3) 	
	Datentyp:	INT

Beispiel

Der 2. Transformation wird der Name "B-Achse" zugewiesen:

Programmcode	Kommentar
N100 \$NT_BASE_TOOL_COMP[n,0];	; Transformation "n" wird in Richtung der X-Achse kompensiert.

9.2.5 Additive Systemvariablen für Orientierungstransformation**9.2.5.1 Übersicht**

Systemvariable	Bedeutung
\$NT_BASE_ORIENT	Werkzeuggrundorientierung
\$NT_BASE_ORIENT_NORMAL	Normalenvektor der Orientierung
\$NT_ROT_AX_POS	Achspositionen der Orientierungsachsen, die als konstante Drehungen parametrisiert sind
\$NT_POLE_LIMIT	Endwinkeltoleranz bei Interpolation durch den Pol
\$NT_POLE_TOL	Endwinkeltoleranz bei Polinterpolation
\$NT_IGNORE_TOOL_ORIENT	Werkzeugorientierung immer aus \$NT_BASE_ORIENT / \$NT_BASE_ORIENT_NORMAL übernehmen.
\$NT_CORR_ELEM_T	Namen der Korrektur Elemente in der Tool-Kette, die von der Funktion CORRTRAF0(...) beschrieben werden können.
\$NT_CORR_ELEM_P	Namen der Korrektur Elemente in der Part-Kette, die von der Funktion CORRTRAF0(...) beschrieben werden können.

Die Systemvariablen sind in den nachfolgenden Kapiteln ausführlich beschrieben.

Hinweis**Namen von Elementen**

Es wird nicht überwacht, ob die Namen von Elementen der kinematischen Kette, auf die zur Parametrierung der Transformation referenziert wird, mehrfach vergeben wurden. Sind solche Namen mehrfach vorhanden, verweisen die Systemdaten der Transformation immer auf das entsprechende Element mit dem niedrigsten Index.

Definierten Ausgangszustand herstellen

Es wird empfohlen, vor Parametrierung der kinematischen Kette einen definierten Ausgangszustand zu erzeugen. Dazu sind die Systemvariablen der kinematischen Kette mit der Funktion DELOBJ() (Seite 225) auf ihren Defaultwert zu setzen.

Ändern von Systemvariablenwerten

Eine Änderungen von Systemvariablenwerten der Transformationsdaten \$NT_... wird erst durch das Wirksamsetzen von Maschinendaten NEWCONF wirksam. Dies kann über den Softkey auf der Bedienoberfläche im Bedienbereich "Inbetriebnahme" > "Maschinendaten" > "MD wirksam setzen", in einem Programm durch den Befehl NEWCONF oder durch Programmende-Reset, Kanal-Reset oder Warm- oder Kaltstart erfolgen.

9.2.5.2 \$NT_BASE_ORIENT

Funktion

In die Systemvariable ist der Richtungsvektor der Werkzeuggrundorientierung einzutragen. Die Werkzeuggrundorientierung wird wirksam, wenn kein Werkzeug ausgewählt ist.

Syntax

`$NT_BASE_ORIENT[<n>,<k>] = <VectorComp>`

Bedeutung

\$NT_BASE_ORIENT:	Richtungsvektor der Werkzeuggrundorientierung (X; Y; Z)	
	Datentyp:	REAL
	Defaultwert:	(0.0, 0.0, 1.0)
	Wertebereich:	Richtungsvektor: $1 \cdot 10^{-6} < \text{Vektor} \leq \text{max. REAL-Wert}$
<n>:	Systemvariablen- bzw. Transformations-Index	
	Datentyp:	INT
	Wertebereich:	1, 2, ... (MD18866 \$MN_MM_NUM_KIN_TRAFOS - 1)
<k>:	Index der Koordinaten	
	Datentyp:	INT
	Wertebereich:	0: X-Koordinate (Abszisse) 1: Y-Koordinate (Ordinate) 2: Z-Koordinate (Applikate)
	Koordinatenwert der Vektor-Komponente	
<VectorComp>:	Koordinatenwert der Vektor-Komponente	
	Datentyp:	STRING

Beispiel

Der Richtungsvektor der Werkzeuggrundorientierung der ersten Transformation hat den Wert (1.0, 2.0, 3.0):

Programmcode	Kommentar
	; 1. Transformation,
	; Vektor-Komponenten:
N100 \$NT_BASE_ORIENT[1,0] = 1.0	; X-Koordinate (Abszisse)
N110 \$NT_BASE_ORIENT[1,1] = 2.0	; Y-Koordinate (Ordinate)
N120 \$NT_BASE_ORIENT[1,2] = 3.0	; Z-Koordinate (Applikate)

9.2.5.3 \$NT_BASE_ORIENT_NORMAL

Funktion

In die Systemvariable ist der Normalenvektor der Werkzeuggrundorientierung einzutragen. Die Systemvariable bzw. der Normalenvektor der Werkzeuggrundorientierung ist nur bei Transformationen mit drei Orientierungsfreiheitsgraden relevant, wenn kein Werkzeug ausgewählt ist.

Syntax

\$NT_BASE_ORIENT_NORMAL[<n>,<k>] = <VectorComp>

\$NT_BASE_ORIENT_NORMAL: L:	Normalenvektor der Werkzeuggrundorientierung (X; Y; Z)	
	Datentyp:	REAL
	Defaultwert:	(0.0, 1.0, 0.0)
	Wertebereich:	Normalenvektor: $1 \cdot 10^{-6} < \text{Vektor} \leq \text{max. REAL-Wert}$
<n>:	Systemvariablen- bzw. Transformations-Index	
	Datentyp:	INT
	Wertebereich:	1, 2, ... (MD18866 \$MN_MM_NUM_KIN_TRAFOS - 1)
<k>:	Index der Koordinaten	
	Datentyp:	INT
	Wertebereich:	0: X-Koordinate (Abszisse)
		1: Y-Koordinate (Ordinate) 2: Z-Koordinate (Applikate)
<VectorComp>:	Koordinatenwert der Vektor-Komponente	
	Datentyp:	STRING

Beispiel

Der Normalenvektor der Werkzeuggrundorientierung der ersten Transformation hat den Wert (2.0, -1.0, 3.0):

Programmcode	Kommentar
	; 1. Transformation,
	; Vektor-Komponenten:
N100 \$NT_BASE_ORIENT_NORMAL[1,0] = 2.0	; X-Koordinate (Abszisse)
N110 \$NT_BASE_ORIENT_NORMAL[1,1] = -1.0	; Y-Koordinate (Ordinate)
N120 \$NT_BASE_ORIENT_NORMAL[1,2] = 3.0	; Z-Koordinate (Applikate)

9.2.5.4 \$NT_ROT_AX_POS

Funktion

In die Systemvariable sind die Positionen der Orientierungsachsen einzutragen, die sich aus den konstanten Drehungen ergeben, die in Elementen der kinematischen Kette vom Typ ROT_CONST (Seite 219) parametrier sind. Die durch diese Elemente wirksamen Positionen der Orientierungsachsen können somit beschrieben werden, ohne dass die kinematische Struktur der Transformation bzw. die Indices der zugehörigen kinematischen Elemente bekannt sein müssen.

Syntax

\$NT_ROT_AX_POS [<n>, <k>] = <OriAxPos>

Bedeutung

\$NT_ROT_AX_POS:	Positionen der Orientierungsachsen aufgrund der konstanten Drehung	
	Datentyp:	REAL
	Defaultwert:	(0.0, 0.0, 0.0)
	Wertebereich:	- max. REAL-Wert ≤ x ≤ + max. REAL-Wert
<n>:	Systemvariablen- bzw. Transformations-Index	
	Datentyp:	INT
	Wertebereich:	1, 2, ... (MD18866 \$MN_MM_NUM_KIN_TRAFOS - 1)
<k>:	Index der Orientierungsachsen	
	Datentyp:	INT
	Wertebereich:	0: 1. Orientierungsachse 1: 2. Orientierungsachse 2: 3. Orientierungsachse
	Positionswert	
<OriAxPos>:	Positionswert	
	Datentyp:	REAL

Beispiel

Die Positionen der Orientierungsachsen, die sich aus der parametrierten konstanten Drehung ergeben, sind: (0, 0, 45.0°)

Programmcode	Kommentar
	; 1. Transformation,
	; Positionen der Orientierungsachsen:
N100 \$NT_ROT_AX_POS[1,0] = 0	; 1. Orientierungsachse
N110 \$NT_ROT_AX_POS[1,1] = 0	; 2. Orientierungsachse
N120 \$NT_ROT_AX_POS[1,2] = 45.0	; 3. Orientierungsachse

9.2.5.5 \$NT_POLE_LIMIT

Funktion

In die Systemvariable wird der maximal zulässige Winkel eingetragen, um den die C-Achse bei einer Großkreisinterpolation am Satzende von ihrer programmierten Position abweichen darf.

Endwinkel

Bei der 5-Achs-Transformation spannen die beiden Orientierungsachsen des Werkzeugs ein Kugelkoordinatensystem, bestehend aus Längen- und Breitenkreisen, auf.

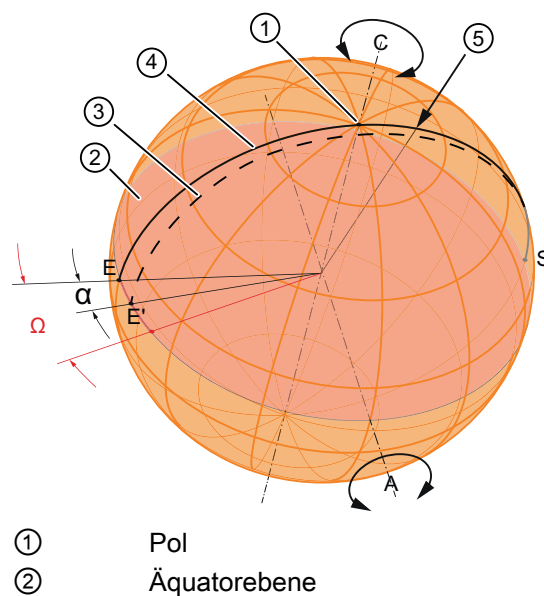
Wird eine Verfahrbewegung programmiert, die nicht exakt durch den Pol, aber innerhalb des durch MD 24530 (\$MC_TRAFO5_NON_POLE_LIMIT_n) gegebenen Bereichs in der Nähe des Pols verlaufen soll, wird von der vorgegebenen Bahn abgewichen, da die Interpolation exakt durch den Polpunkt verläuft. Dadurch ergibt sich im Endpunkt der vierten Achse (der Polachse) eine Positionsabweichung gegenüber dem programmierten Wert.

Die Systemvariable gibt den Winkel an, um den die Polachse bei der 5-Achs-Transformation vom programmierten Wert abweichen darf, wenn von der programmierten Interpolation auf die Interpolation durch den Polpunkt umgeschaltet wird.

Ergibt sich eine größere Abweichung, wird eine Fehlermeldung ausgegeben (Alarm 14112) und die Interpolation nicht durchgeführt.

Bei konventioneller 5- bzw. 6-Achsinterpolation ist eine Polstellung dadurch gekennzeichnet, dass sich bei Drehung einer Rundachse die Werkzeugorientierung nicht ändert. Verläuft dann eine programmierte Bahn in der Nähe des Pols, können sich für einzelne Rundachsen extrem hohe Verfahrgeschwindigkeiten ergeben, um die Orientierung beizubehalten. Verläuft die programmierte Bahn exakt durch den Pol, ergibt sich diese Problematik u. U. nicht.

In der Systemvariablen kann ein Polwinkel parametrisiert werden, der einen Toleranzkreis um den Pol definiert. Führt eine programmierte Werkzeugbahn innerhalb des Toleranzkreises am Pol vorbei, wird von Orientierungsinterpolation auf Linear- / Rundachsinterpolation umgeschaltet.



9.2 Inbetriebnahme

- ③ Programmierter Bahn
- ④ Gefahrene Bahn durch den Pol
- ⑤ Endwinkel
- A Rundachse um die X-Koordinatenachse des WKS
- C Rundachse um die Z-Koordinatenachse des WKS
- S Startpunkt der Interpolation / Verfahrssatz
- E Angefahrener Endpunkt
- E' Programmierter Endpunkt
- α Endwinkel
- Ω Maximal zulässiger Endwinkel (\$NT_POLE_LIMIT)

Syntax

\$NT_POLE_LIMIT[<n>] = <PoleTolAngle>

Bedeutung

\$NT_POLE_LIMIT:	Polwinkel	
	Einheit:	Grad
	Datentyp:	REAL
	Defaultwert:	2.0
	Wertebereich:	- max. REAL-Wert ≤ x ≤ + max. REAL-Wert
<n>:	Systemvariablen- bzw. Transformations-Index	
	Datentyp:	INT
	Wertebereich:	1, 2, ... (MD18866 \$MN_MM_NUM_KIN_TRAFOS - 1)
<PoleTolAngle>:	Toleranzwert	
	Datentyp:	REAL

Beispiel

Für die erste Transformation wird ein Polwinkel von 2.54° eingestellt:

Programmcode	Kommentar
N100 \$NT_POLE_LIMIT[1] = 2.54	; 1. Transformation, ; Polwinkel

9.2.5.6 \$NT_POLE_TOL

Funktion

In die Systemvariable ist die Endwinkel-Toleranz bei Interpolation durch den Pol für die erste 5/6-Achs-Transformation einzutragen.

Die Systemvariable wird nur von der generischen 5/6-Achs Transformation ausgewertet.

Liegt die programmierte Endorientierung innerhalb des Polkegels und innerhalb des mit diesem MD angegebenen Toleranzkegels, bewegt sich die Polachse nicht und behält ihre Startpositionen bei. Die andere Rundachse nimmt dagegen den programmierten Winkel an.

Dadurch gibt es eine Abweichung der Endorientierung von der programmierten Orientierung.

Eine weitere Bedeutung dieser Systemvariable ist die Behandlung der programmierten Endorientierung bei nicht rechtwinkligen Kinematiken. Bei diesen Maschinenkinematiken können in der Regel nicht alle Werkzeugorientierungen eingestellt werden. Wird eine Orientierung programmiert, die außerhalb des einstellbaren Bereichs liegt, wird der Alarm 14112 "Programmierter Orientierungsweg nicht möglich" ausgegeben.

Syntax

```
$NT_POLE_TOL[<n>] = <PoleTolLim>
```

Bedeutung

\$NT_POLE_TOL:	Endwinkeltoleranz bei Polinterpolation	
	Datentyp:	REAL
	Defaultwert:	0.0
	Wertebereich:	$0.0 \leq x \leq 45.0$
<n>:	Systemvariablen- bzw. Transformations-Index	
	Datentyp:	INT
	Wertebereich:	1, 2, ... (MD18866 \$MN_MM_NUM_KIN_TRAFOS - 1)
<PoleTolLim>:	Winkel	
	Datentyp:	REAL

Beispiel

Für die erste Transformation wird eine Endwinkeltoleranz von 2.54° definiert:

Programmcode	Kommentar
N100 \$NT_POLE_TOL[1] = 2.54	; 1. Transformation, ; Endwinkeltoleranz

9.2.5.7 \$NT_IGNORE_TOOL_ORIENT

Funktion

Jedes in der Steuerung bekannte Werkzeug hat eine definierte Orientierung. Diese Werkzeugorientierung wird normalerweise für die Berechnungen der Bewegungen der Orientierungsachsen zu Grunde gelegt. Über die Systemvariable kann festgelegt werden, dass auch bei aktivem Werkzeug nicht die Werkzeugorientierung, sondern die in den Systemvariablen \$NT_BASE_ORIENT (Seite 376) und \$NT_BASE_ORIENT_NORMAL (Seite 377) parametrisierte Orientierung für die Berechnungen der Bewegungen der Orientierungsachsen verwendet wird.

Syntax

`$NT_IGNORE_TOOL_ORIENT[<n>] = <Value>`

Bedeutung

\$NT_IGNORE_TOOL_ORIENT:	Umschalten auf die in den Systemvariablen \$NT_BASE_ORIENT und \$NT_BASE_ORIENT_NORMAL parametrisierte Orientierung zur Berechnungen der Bewegungen der Orientierungsachsen	
	Datentyp:	BOOL
	Defaultwert:	FALSE
<n>:	Systemvariablen- bzw. Transformations-Index	
	Datentyp:	INT
	Wertebereich:	1, 2, ... (MD18866 \$MN_MM_NUM_KIN_TRAFOS - 1)
<Value>:	Wert	
	Datentyp:	BOOL

Beispiel

Für die erste Transformation wird die in den Systemvariablen \$NT_BASE_ORIENT und \$NT_BASE_ORIENT_NORMAL parametrisierte Orientierung zur Berechnungen der Bewegungen der Orientierungsachsen aktiviert:

Programmcode	Kommentar
<code>N100 \$NT_IGNORE_TOOL_ORIENT[1] = TRUE</code>	<code>; 1. Transformation, ; Orientierung gemäß Systemvariablen</code>

9.2.5.8 \$NT_CORR_ELEM_T

Funktion

Mit diesem Systemdatum wird auf maximal 4 konstante Kettenelemente (\$NK_NAME) in der Tool-Kette verwiesen, die zur Aufnahme von Korrekturwerten (linearen Offsets), wie sie z. B. in Messzyklen ermittelt werden, vorgesehen sind. Die Korrekturwerte werden von der Funktion CORRTRAFO (Seite 387) ermittelt. Es ist nur bei Orientierungstransformationen von Bedeutung.

Zwischen zwei dieser Elemente muss in der kinematischen Kette immer eine Orientierungsachse liegen. Das bedeutet, dass nur bei 6-Achs-Transformationen, bei denen alle 3 Orientierungsachsen in der Tool-Kette definiert sind, alle 4 Kettenelemente belegt sein können, während z. B. bei 5-Achstransformationen dieses Systemdatum nur maximal drei Einträge enthalten darf.

Die gesamte kinematische Kette vom Maschinennullpunkt (Bezugspunkt der kinematischen Kette) bis zur Werkzeugaufnahme wird durch die Orientierungsachsen in maximal 4 Abschnitte geteilt. In jedem dieser Abschnitte kann es maximal ein Korrekturwert geben. Das Korrekturwertelement mit dem Index n muss im n-ten Abschnitt liegen (Beispiel: \$NT_CORR_ELEM_T[k, 1] muss auf ein Kettenelement zwischen der ersten und der zweiten Orientierungsachse der Tool-Kette verweisen).

Syntax

```
$NT_CORR_ELEM_T[<n>, <k>] = "<CORR_TElementName>"
```

Bedeutung

\$NT_CORR_ELEM_T:	Namen der Elemente der aktuell wirksamen kinematischen Kette.	
	Datentyp:	STRING
	Defaultwert:	""
	Wertebereich:	Elementnamen der aktuell wirksamen kinematischen Kette
<n>:	Systemvariablen- bzw. Transformations-Index	
	Datentyp:	INT
	Wertebereich:	1, 2, ... (MD18866 \$MN_MM_NUM_KIN_TRAFOS - 1)
<k>:	Position in der kinematischen Tool-Kette	
	Datentyp:	INT
	Wertebereich:	0..3
<CORR_TElementName>:	Name eines Elements der aktuell wirksamen kinematischen Kette, das einen Korrekturwert (lineares Offset) aufnimmt.	
	Datentyp:	STRING

Beispiel

Definiert ein Korrekturlement in der Werkzeug-Kette, das zur Aufnahme von Korrekturwerten dient.

Programmcode	Kommentar
\$NT_CORR_ELEM_T(1,2) = Corr_1;	Verweist auf ein Korrekturlement in der Werkzeug-Kette der Transformation "1" und der Position "2".

9.2.5.9 \$NT_CORR_ELEM_P

Funktion

Mit diesem Systemdatum wird auf maximal 4 konstante Kettenelemente (\$NK_NAME) in der Part-Kette verwiesen, die zur Aufnahme von Korrekturwerten (linearen Offsets), wie sie z. B. in Messzyklen ermittelt werden, vorgesehen sind. Die Korrekturwerte werden von der Funktion CORRTRAF0 (Seite 387) ermittelt. Es ist nur bei Orientierungstransformationen von Bedeutung.

Zwischen zwei dieser Elemente muss in der kinematischen Kette immer eine Orientierungsachse liegen. Das bedeutet, dass nur bei 6-Achs-Transformationen, bei denen alle 3 Orientierungsachsen in der Tool-Kette definiert sind, alle 4 Kettenelemente auch belegt sein können, während z. B. bei 5-Achstransformationen dieses Systemdatum nur maximal drei Einträge enthalten darf.

9.2 Inbetriebnahme

Die gesamte kinematische Kette vom Maschinennullpunkt (Bezugspunkt der kinematischen Kette) bis zum Werkstückmittelpunkt (Part) wird durch die Orientierungsachsen in maximal 4 Abschnitte geteilt. In jedem dieser Abschnitte kann es maximal ein Korrektorelement geben. Das Korrektorelement mit dem Index n muss im n-ten Abschnitt liegen (Beispiel: \$NT_CORR_ELEM_T[k, 1] muss auf ein Kettenelement zwischen der ersten und der zweiten Orientierungsachse der Part-Kette verweisen).

Syntax

\$NT_CORR_ELEM_P[<n>,<m>,<k>] = "<CORR_PElementName>"

Bedeutung

\$NT_CORR_ELEM_P:	Namen der Elemente der aktuell wirksamen kinematischen Kette.	
	Datentyp:	STRING
	Defaultwert:	""
	Wertebereich:	Elementnamen der aktuell wirksamen kinematischen Kette
<n>:	Systemvariablen- bzw. Transformations-Index	
	Datentyp:	INT
	Wertebereich:	1, 2, ... (MD18866 \$MN_MM_NUM_KIN_TRAFOS - 1)
<k>:	Position in der kinematischen Tool-Kette	
	Datentyp:	INT
	Wertebereich:	0...3
<CORR_PElementName>:	Name eines Elements der aktuell wirksamen kinematischen Kette, das einen Korrekturwert (lineares Offset) aufnimmt.	
	Datentyp:	STRING

Beispiel

Definiert ein Korrektorelement in der Werkstück-Kette, das zur Aufnahme von Korrekturwerten dient.

Programmcode	Kommentar
\$NT_CORR_ELEM_P(1,2) = Corr_1	Verweist auf ein Korrektorelement in der Werkstück-Kette der Transformation "1" und der Position "2".

9.2.6 Additive Systemvariablen für Stirnflächentransformationen (TRANSMIT)

9.2.6.1 Übersicht

Systemvariable	Bedeutung
\$NT_POLE_SIDE_FIX	Einschränkung des Arbeitsbereichs vor bzw. hinter dem Pol

Die Systemvariablen sind in den nachfolgenden Kapiteln ausführlich beschrieben.

Hinweis

Namen von Elementen

Es wird nicht überwacht ob die Namen von Elementen der kinematischen Kette, auf die zur Parametrierung der Transformation referenziert wird, mehrfach vergeben wurde. Sind solche Namen mehrfach vorhanden, verweisen die Systemdaten der Transformation immer auf das entsprechende Element mit dem niedrigsten Index.

Definierten Ausgangszustand herstellen

Es wird empfohlen, vor Parametrierung der kinematischen Kette einen definierten Ausgangszustand zu erzeugen. Dazu sind die Systemvariablen der kinematischen Kette mit der Funktion DELOBJ() (Seite 225) auf ihren Defaultwert zu setzen.

Ändern von Systemvariablenwerten

Eine Änderungen von Systemvariablenwerten der Transformationsdaten \$NT_... wird erst durch das Wirksamsetzen von Maschinendaten NEWCONF wirksam. Dies kann über den Softkey auf der Bedienoberfläche im Bedienbereich "Inbetriebnahme" > "Maschinendaten" > "MD wirksam setzen" , in einem Programm durch den Befehl NEWCONF oder durch Programmende-Reset, Kanal-Reset oder Warm- oder Kaltstart erfolgen.

9.2.6.2 \$NT_POLE_SIDE_FIX

Funktion

Mit der Systemvariablen ist die Einschränkung des Arbeitsbereichs vor und hinter dem Pol einzustellen.

Syntax

```
$NT_POLE_SIDE_FIX[<n>] = <POLESIDEFIX>
```

Bedeutung

\$NT_POLE_SIDE_FIX:	Namen der Elemente der aktuell wirksamen kinematischen Kette, die die Rundachsen definieren	
	Datentyp:	INT
	Defaultwert:	""
	Wertebereich:	Elementnamen der aktuell wirksamen kinematischen Kette
<n>:	Systemvariablen- bzw. Transformations-Index	
	Datentyp:	INT
	Wertebereich:	1, 2, ... (MD18866 \$MN_MM_NUM_KIN_TRAFOS - 1)

<code><PoleSIDEFIX></code> :	Einschränkung des Arbeitsbereichs:	
	<ul style="list-style-type: none"> • 0 = keine Einschränkung; Durchfahren des Pols ist erlaubt. • 1 = Arbeitsbereich der Linearachse für Positionen ≥ 0 (wenn Werkzeuglängenkorrektur parallel zu Linearachse = 0). • 2 = Arbeitsbereich der Linearachse für Positionen ≤ 0 (wenn Werkzeuglängenkorrektur parallel zu Linearachse = 0). 	
	Datentyp:	STRING

Beispiel

Für die erste Transformation sind die Name der Elemente der aktuell wirksamen kinematischen Kette, die die Rundachsen definieren "ROT_AXIS_1", "ROT_AXIS_2" und "ROT_AXIS_3".

Programmcode	Kommentar
	; 1. Transformation,
	; Elemente der Rundachsen:
N100 \$NT_ROT_AX_NAME[1,0] = "ROT_AXIS_1"	; 1. Rundachse
N110 \$NT_ROT_AX_NAME[1,1] = "ROT_AXIS_2"	; 2. Rundachse
N120 \$NT_ROT_AX_NAME[1,2] = "ROT_AXIS_3"	; 3. Rundachse

9.3 Programmierung

9.3.1 Transformation einschalten (TRAFOON)

Eine mit kinematischen Ketten definierte Transformation wird mit der vordefinierten Prozedur TRAFOON aktiviert. Der Aufruf muss alleine in einem Satz stehen.

Hinweis

Alternativ kann eine mit kinematischen Ketten definierte Transformation auch über konventionelle Sprachbefehle wie z. B. TRAORI oder TRANSMIT aktiviert werden. Dazu muss in der Systemvariablen \$NT_TRAFO_INDEX ein entsprechender Wert ungleich null eingetragen sein.

Weitere Informationen zu \$NT_TRAFO_INDEX siehe "Listenhandbuch Systemvariablen".

Syntax

TRAFOON (<Trafoname>, <Diameter>, <k>)

Bedeutung

TRAFOON:	Prozedur zur Aktivierung einer mit kinematischen Ketten definierten Transformation		
<Trafoname>:	Name des Transformationsdatensatzes		
	Datentyp:	STRING	
	Wertebereich:	Alle über \$NT_NAME (Seite 351) definierte Namen von Transformationsdatensätzen	
	Hinweis: Der Name des Transformationsdatensatzes muss eindeutig sein. Er darf in \$NT_NAME nur einmal vorkommen.		
<Diameter>:	Bezugs- bzw. Arbeitsdurchmesser (nur TRACYL)		
	Datentyp:	REAL	
	Der Wert muss > 1 sein.		
<k>:	Definiert die Verwendung der Nutwandkorrektur (nur TRACYL).		
	Datentyp:	BOOL	
	Wert:	FALSE	ohne Nutwandkorrektur
		TRUE	mit Nutwandkorrektur
	Entspricht dem TRACYL-Transformationstyp 514 (Nutwandkorrektur programmierbar). Ist <k> nicht angegeben, wirkt die parametrisierte Einstellung von Bit 10 in \$NT_CNTRL[<n>].		

Beispiel

Programmcode	Kommentar
TRAFOON["Trans_1"]	Aktiviert die Transformation mit dem Namen Trans_1.

9.3.2 Orientierungstransformation nach Maschinenvermessung modifizieren (CORRTRAFO)

Bei Maschinen mit Orientierungstransformationen, die mittels kinematischer Ketten definiert wurden, kann der Anwender die vordefinierte Funktion CORRTRAFO verwenden, um nach einer Maschinenvermessung die Offsetvektoren oder die Richtungsvektoren der Orientierungsachsen im kinematischen Modell der Maschine zu modifizieren.

Syntax

```
<Corr_Status> = CORRTRAFO(<Corr_Vect>, <Corr_Index>, <Corr_Mode>,
[ <No_Alarm>])
```

Bedeutung

CORRTRAF0:	Funktionsaufruf	
<Corr_Status>:	Rückgabewert der Funktion	
	Datentyp:	INT
	Werte:	0 Die Funktion wurde ohne Fehler ausgeführt.
		1 Es ist keine Transformation aktiv.
		2 Die aktuell aktive Transformation ist keine Orientierungstransformation.
		3 Die aktive Orientierungstransformation wurde nicht mit kinematischen Ketten definiert.
		10 Der Aufrufparameter <Corr_Index> ist negativ.
		11 Der Aufrufparameter <Corr_Mode> ist negativ.
		12 Ungültiger Verweis auf einen Abschnitt einer Teilkette (1er-Stelle von <Corr_Index>). Der Wert darf nicht größer sein als die Zahl der Orientierungsachsen in der Teilkette.
		13 Ungültiger Verweis auf die Orientierungsachse einer Teilkette (1er-Stelle von <Corr_Index>). Der Wert muss kleiner sein als die Zahl der Orientierungsachsen in der Teilkette.
		14 Ungültiger Verweis auf eine Teilkette (10er-Stelle von <Corr_Index>). Es sind nur die Werte 0 und 1 zulässig (Verweis auf Part- bzw. Tool-Kette). Diese Fehlernummer tritt auch auf, wenn die Teilkette, auf die <Corr_Index> verweist, nicht existiert.
		15 Im Abschnitt, auf den mit Parameter <Corr_Index> verwiesen wird, ist kein Korrekturlement definiert (\$NT_CORR_ELEM_P bzw. \$NT_CORR_ELEM_T).
		20 Ungültiger Korrekturmodus (1er-Stelle von <Corr_Mode>). Es sind nur die Werte 0 und 1 zulässig.
		21 Ungültiger Korrekturmodus (10er- und/oder 100er-Stelle von <Corr_Mode>). Beim Schreiben einer Achsrichtung darf nur die 1er-Stelle ungleich null sein.
		30 Die 100er-Stelle von <Corr_Mode> ist ungültig. Es sind nur die Werte 0 und 1 zulässig.
		31 Die 1000er-Stelle von <Corr_Mode> ist ungültig. Es sind nur die Werte 0 und 1 zulässig.
		40 Der Richtungsvektor, der als Achsrichtung übernommen werden soll, ist der Nullvektor. Dieser Fehler kann nur auftreten, wenn die 1000er-Stelle von <Corr_Mode> gleich 0 ist. Ist die 1000er-Stelle dieses Parameters gleich 1 (Überwachung der maximalen Korrektur deaktiviert), kann auch der Nullvektor geschrieben werden.

		41	Bei der Korrektur eines Offsetvektors ist die Abweichung gegenüber dem aktuellen Werte in mindestens einer Koordinate größer als der durch das Settingdatum SD41610 \$SN_CORR_TRAFO_LIN_MAX vorgegebene Maximalwert. Der Parameter <Corr_Vect> wird durch einen Fehlervektor überschrieben. Das gilt auch dann, wenn die Bearbeitung mit Alarm abgebrochen wird (siehe Parameter <No_Alarm>). Der Fehlervektor enthält in den Komponenten, deren Korrekturwert das zulässige Limit überschritten hat, die vorzeichenrichtige Differenz zwischen dem ermittelten Korrekturwert und dem Limit. Der Inhalt der Komponenten, die ihr Limit nicht überschritten haben, ist null.
		42	Bei der Korrektur eines Richtungsvektors ist die Winkelabweichung gegenüber der aktuellen Richtung größer als der durch das Settingdatum SD41611 \$SN_CORR_TRAFO_DIR_MAX vorgegebene Maximalwert.
		43	Der Versuch, eine Systemvariable zu beschreiben, wurde wegen fehlender Schreibrechte abgewiesen.
<Corr_Vect>:	Korrekturvektor Der Inhalt des Korrekturvektors wird durch die nachfolgenden Parameter <Corr_Index> und <Corr_Mode> definiert. Ist <Corr_Status> = 41, wird der Inhalt des Vektors überschrieben (s. o.). Datentyp: REAL		
<Corr_Index>:	Abschnitt, dessen Korrekturlement modifiziert werden soll / Index der Orientierungsachse, deren Richtungsvektor modifiziert werden soll Datentyp: INT Der Parameter <Corr_Index> ist dezimalcodiert (1er- bis 10er-Stelle):		
	1er-Stelle:	Enthält den Index des Abschnitts bzw. der Orientierungsachse in der Teilkette.	
	10er-Stelle:	Verweist auf die Teilkette.	
		0x	Werkstück-Kette
		1x	Werkzeug-Kette

<Corr_Mode>:	Korrekturmodus		
	Datentyp:	INT	
	Der Parameter <Corr_Mode> ist dezimalcodiert (1er- bis 1000er-Stelle):		
	1er-Stelle:	Bestimmt, welches Element korrigiert werden soll.	
		xxx0	Korrektur eines linearen Offsetvektors
		xxx1	Korrektur des Richtungsvektors einer Orientierungsachse
	10er-Stelle:	Bestimmt, wie das Korrekturelement, auf das der Inhalt von <Corr_Index> verweist, modifiziert werden soll.	
		xx0x	Der Korrekturvektor wird unmittelbar in das Korrekturelement geschrieben. Diese Variante kann dazu verwendet werden, das Korrekturelement unmittelbar zu beschreiben, ohne dass der Index <n> des betreffenden Systemdatums (\$NK_OFF_DIR[<n>, ...]) bekannt sein muss.
		xx1x	Wie 0, jedoch mit dem Unterschied, dass der übergebene Korrekturwert in Weltkoordinaten interpretiert wird. Ein Unterschied zwischen den Varianten 0 und 1 kann immer dann entstehen, wenn die kinematische Kette in Grundstellung (Positionen aller Orientierungsachsen gleich 0) weitere Drehungen enthält.
		xx2x	Wie 1, jedoch mit dem Unterschied, dass sich der Korrekturwert auf den gesamten Abschnitt bezieht, d. h., in das Korrekturelement wird ein solcher Wert eingetragen, dass der gesamte Abschnitt die durch den Korrekturwert definierte Länge erreicht.
Hinweis: Die Werte 1 und 2 sind beim Beschreiben des Richtungsvektors einer Orientierungsachse nicht erlaubt.			
100er-Stelle:	Bestimmt, wie der Inhalt des Parameters <Corr_Vect> zu interpretieren ist.		
	x0xx	Der übergebene Korrekturvektor <Corr_Vect> enthält die gesamte neue Länge des Korrekturelements bzw. des Abschnitts, auf den <Corr_Index> in Verbindung mit der 10er-Stelle von <Corr_Mode> verweist (absolute Korrektur).	
	x1xx	Der übergebene Korrekturvektor <Corr_Vect> enthält nur die Differenz gegenüber der aktuellen Länge des Korrekturelements bzw. des Abschnitts, auf den <Corr_Index> in Verbindung mit der 10er-Stelle von <Corr_Mode> verweist (inkrementelle Korrektur).	
	Hinweis: Bei der Korrektur des Richtungsvektors einer Orientierungsachse muss der Inhalt der 100er-Stelle 0 sein.		
1000er-Stelle:	Bestimmt, ob die Korrektur durch folgenden Maximalwert begrenzt werden soll:		
	<ul style="list-style-type: none"> SD41610 \$SN_CORR_TRAFO_LIN_MAX (Seite 348) bzw. SD41611 \$SN_CORR_TRAFO_DIR_MAX (Seite 348) 		
	0xxx	Überwachung der maximalen Korrektur ist aktiv.	
	1xxx	Überwachung der maximalen Korrektur ist nicht aktiv.	

<No_Alarm>:	Verhalten im Fehlerfall (Rückgabewert > 0) (optional)				
	Datentyp:	BOOL			
	Wert:	<table border="1"> <tr> <td>FALSE (Default)</td> <td>Im Fehlerfall wird die Programmabarbeitung angehalten und Alarm 14103 angezeigt.</td> </tr> <tr> <td>TRUE</td> <td>Im Fehlerfall wird die Programmabarbeitung nicht angehalten und es wird kein Alarm angezeigt. Anwendungsfall: Anwenderspezifische Reaktion entsprechend Rückgabewert</td> </tr> </table>	FALSE (Default)	Im Fehlerfall wird die Programmabarbeitung angehalten und Alarm 14103 angezeigt.	TRUE
FALSE (Default)	Im Fehlerfall wird die Programmabarbeitung angehalten und Alarm 14103 angezeigt.				
TRUE	Im Fehlerfall wird die Programmabarbeitung nicht angehalten und es wird kein Alarm angezeigt. Anwendungsfall: Anwenderspezifische Reaktion entsprechend Rückgabewert				

Hinweis

Falls beim Aufruf der Funktion ein Fehler auftritt, wird entweder ein Alarm ausgegeben oder eine Fehlernummer zurückgegeben (siehe Parameter <No_Alarm>), sodass der Anwender selbst geeignet auf den Fehlerzustand reagieren kann. Die Fehlerursache wird durch einen Alarmparameter näher bezeichnet. Eine an Stelle eines Alarms zurückgegebene Fehlernummer ist identisch mit dem Alarmparameter.

Siehe auch

Winkelabweichung für Richtungsvektoren bei CORRTRAFO (Seite 348)

Weitere Informationen zu CORRTRAFO

Die kinematische Struktur einer Maschine mit Orientierungstransformation wird durch eine oder zwei kinematische Ketten (Teilketten) beschrieben, die vom Nullpunkt des Weltkoordinatensystems ausgehen. Eine der beiden Ketten, die **Werkzeug-Kette**, endet am Bezugspunkt des Werkzeugs, die andere, die **Werkstück-Kette** im Nullpunkt des Basiskoordinatensystems.

Die Funktion CORRTRAFO schreibt Hebelarmlängen und Achsrichtungen bei Maschinen mit Orientierungstransformation in spezielle Korrekturalemente. Eine kinematische Kette wird unter anderem mit Elementen vom Typ OFFSET beschrieben, die über \$NK_TYPE definiert werden.

CORRTRAFO arbeitet mit Abschnitten

Die beiden Teilketten können sich jeweils in maximal vier Abschnitte aufteilen:

- Abschnitt 1 beginnt am Startpunkt der Kette und endet an der ersten Orientierungsachse.
- Abschnitt 2 ist der Abschnitt zwischen Orientierungsachse 1 und Orientierungsachse 2.
- Abschnitt 3 ist der Abschnitt zwischen Orientierungsachse 2 und Orientierungsachse 3.
- Abschnitt 4 ist der Abschnitt zwischen Orientierungsachse 3 und dem Ende der Werkzeug- bzw. Werkstück-Kette.

Die Abschnitte können jeweils konstante Kettenelemente des Typs OFFSET oder ROT_CONST enthalten.

Die folgende Grafik zeigt eine Orientierungstransformation mit 2 Orientierungsachsen.

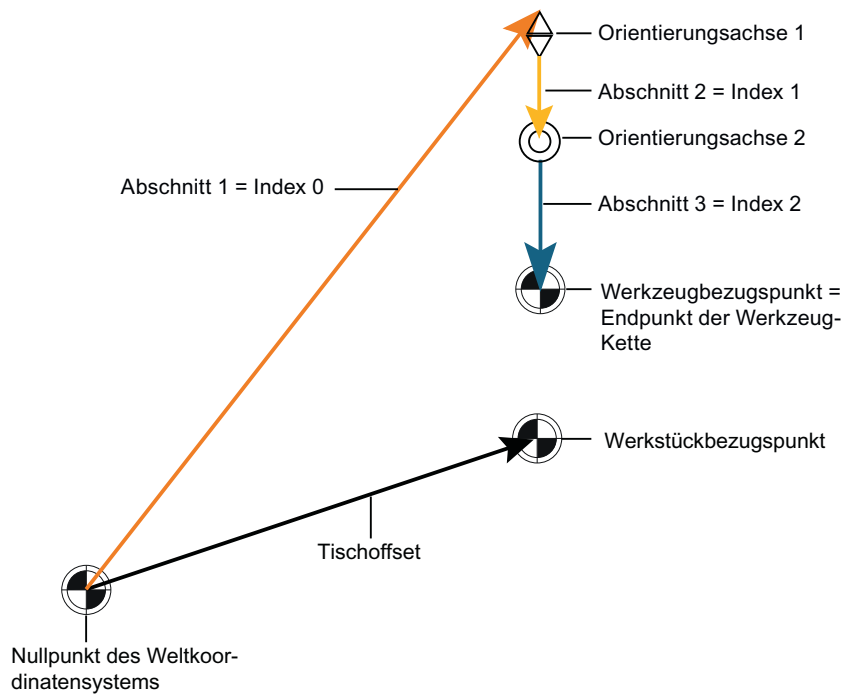


Bild 9-7 CORRTRAFO Beispiel

Die Abschnitte sind eindeutig definiert: Durchläuft man die kinematische Teil-Kette von ihrem Startpunkt bis zum Endpunkt, so hat der erste Abschnitt den Index 0, der nächste den Index 1 usw. Der Index des letzten Abschnitts ist dann immer gleich der Zahl der Orientierungsachsen.

Korrekturelemente

Auf jeweils ein konstantes kinematisches Kettenelement (Kettenelement des Typs $\$NK_TYPE[\langle n \rangle] = "OFFSET"$) in jedem dieser Abschnitte kann mit der Systemvariablen $\$NT_CORR_ELEM_T[\langle n \rangle, 0 \dots 3]$ bzw. $\$NT_CORR_ELEM_P[\langle n \rangle, 0 \dots 3]$ verwiesen werden. In diese so bezeichneten Elemente werden mithilfe der Funktion CORRTRAFO die Korrekturwerte geschrieben, die bei der Maschinenvermessung ermittelt wurden.

Beispiel mit Transformations-Index = 1:

- $\$NT_CORR_ELEM_T[1,0] = "C_AXIS_OFFSET"$; Offset der C-Achse (Orientierungsachse 1) in Abschnitt 1 ist als Korrekturlement definiert.
- $\$NT_CORR_ELEM_T[1,1] = "B_AXIS_OFFSET"$; Offset der B-Achse (Orientierungsachse 2) in Abschnitt 2 ist als Korrekturlement definiert.
- $\$NT_CORR_ELEM_T[1,2] = "BASE_TOOL_OFFSET"$; Offset von der B-Achse zum Werkzeugbezugspunkt in Abschnitt 3 ist als Korrekturlement definiert.

Die Reihenfolge der Verweise in $\$NT_CORR_ELEM_T/P[\langle n \rangle, 0 \dots 3]$ muss mit den oben beschriebenen Abschnitten korrespondieren, d. h. in $\$NT_CORR_ELEM_T/P[\langle n \rangle, 0]$ kann nur ein Kettenelement stehen, das vor der ersten Orientierungsachse liegt usw..

Die Funktion CORRTRAFO schreibt in die so definierten Korrekturlemente die Werte, die über die Vermessung der Maschine bestimmt wurden. Die Modifizierung der Korrekturwerte wird in CORRTRAFO über den Parameter $\langle Corr_Mode \rangle$ definiert.

Kette schließen

Sind Bit 7 oder Bit 8 in der Systemvariablen \$NT_CNTRL[<n>] gesetzt, werden am Ende der Werkstück-Kette (Bit 7) bzw. vor dem Startpunkt der Werkzeug-Kette (Bit 8) intern automatisch zusätzliche konstante Kettenelemente eingefügt, die eine Verbindung vom Endpunkt der Kette zum Maschinennullpunkt herstellen ("Kette schließen").

Diese automatisch eingefügten Elemente können von extern nicht beschrieben, sondern nur gelesen werden (siehe dazu die Systemvariablen \$AC_TRAFO_CORR_ELEM_P/T).

Punkt zum Schließen der Werkzeug-Kette

Ist die Systemvariable \$NT_CLOSE_CHAIN_T nicht leer, wird die Werkzeug-Kette nicht am Endpunkt der Kette, sondern am Endpunkt des bezeichneten Kettenelements geschlossen. Weitere Kettenelemente, die hinter diesem Punkt liegen, führen dann beim Aktivieren der Transformation zu einer entsprechenden Nullpunktverschiebung.

Index einer Orientierungsachse

Außer den konstanten Offsets zwischen den Orientierungsachsen können mit der Funktion CORRTRAFO auch die Richtungsvektoren der Orientierungsachsen beschrieben werden. Unter dem Index einer Orientierungsachse wird hier der Index verstanden, der sich ergibt, wenn man die kinematische Teilkette vom Ursprung bis zu ihrem Ende durchläuft, wobei die Zählung bei null beginnt. Der Index einer Orientierungsachse ist deshalb immer gleich dem Index des davorliegenden Abschnitts.

Der Index einer Orientierungsachse kann auch mit der Systemvariablen \$AC_TRAFO_ORIAX_LOC ermittelt werden.

Maximal zulässige Veränderung eines Kettenelements

Die maximal zulässige Veränderung eines Kettenelements kann durch die beiden Settingdaten SD41610 \$SN_CORR_TRAFO_LIN_MAX für Offsetvektoren und SD41611 \$SN_CORR_TRAFO_DIR_MAX für die Richtungsvektoren von Orientierungsachsen begrenzt werden. SD41610 \$SN_CORR_TRAFO_LIN_MAX gibt an, um welchen Betrag jede einzelne Vektorkomponente gegenüber ihrem Bezugswert maximal verändert werden darf. SD41611 \$SN_CORR_TRAFO_DIR_MAX gibt an, um welchen Winkel sich die Richtung des Achsvektors maximal gegenüber ihrem Bezugswert verändern darf. Der Bezugswert ist dabei immer der entsprechende Wert, der in der beim Aufruf von CORRTRAFO aktiven Transformation wirksam ist. D. h., dass eventuell nach der Aktivierung der Transformation veränderte Inhalte der Kinematikdaten in der Datenhaltung keinen Einfluss auf die Wirkungsweise der Funktion CORRTRAFO haben.

9.4 Beispiele

9.4.1 Vorgaben für TRAORI_DYN

Allgemeines

Anhand einer 5-Achs-Maschine wird beispielhaft das prinzipielle Vorgehen zur Parametrierung einer Transformation mit einer kinematischen Kette über ein Teileprogramm gezeigt. Im Teileprogramm werden alle für die kinematische Kette relevanten Systemvariablen geschrieben:

- Kinematische Ketten mit \$NK_...
- Transformation mit Kinematischer Kette \$NT_...

Options- und Maschinendaten

Für das Beispiel sind folgende Options- und Maschinendaten einzustellen:

- MD18866 \$MN_MM_NUM_KIN_TRAFOS = 2

Transformation mit einer Kinematischen Kette

Für die Beschreibung werden zwei kinematische Ketten definiert:

- Eine kinematische Kette zeigt auf den Werkstückbezugspunkt (Werkstückkoordinatensystem).
- Die zweite kinematische Kette zeigt auf den Werkzeugbezugspunkt.
- Der Transformationstyp ist "TRAORI_DYN".

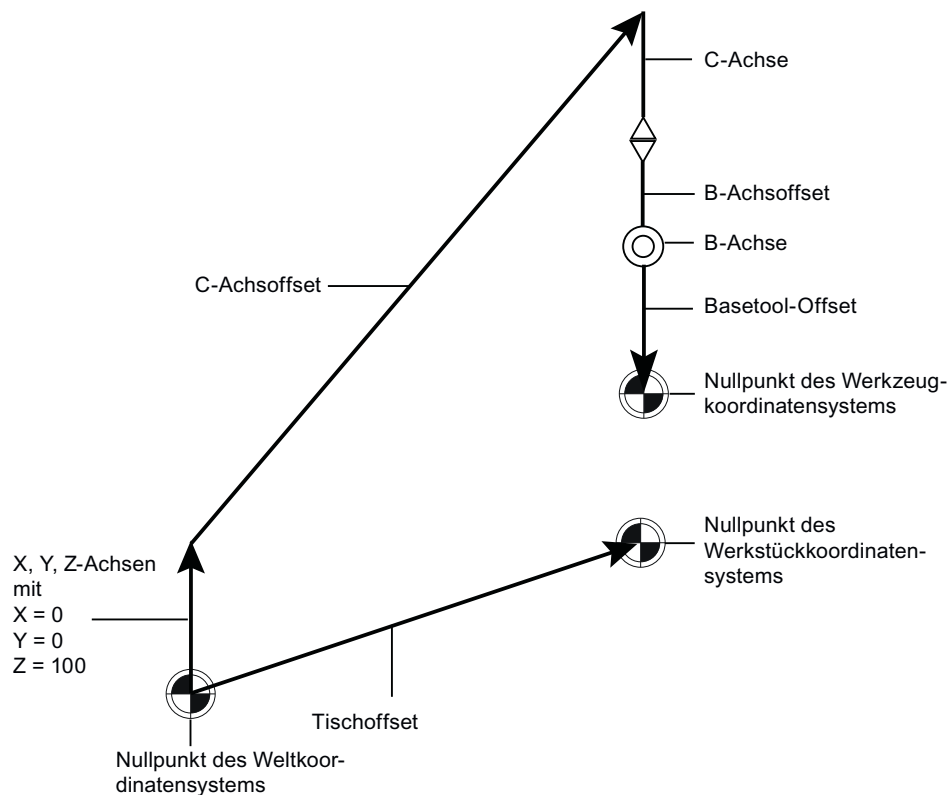


Bild 9-8 Beispielkinematik für Dynamische Orientierungstransformation

Die Kinematik wird in mehreren Schritten erstellt:

- Die beiden Root-Elemente der kinematischen Kette werden definiert:
 - `$NK_NEXT[KIE_CNTR] = „X-Achse“`; Kette zeigt auf den Werkzeugbezugspunkt.
 - `$NK_PARALLEL[KIE_CNTR] = „Tischoffset“`; Kette zeigt auf den Werkstückbezugspunkt.
- Ausgehend vom Nullpunkt des Weltkoordinatensystems werden die drei rechtwinklig aufeinander stehenden Linearachsen X, Y und Z definiert.
 - In den Richtungsvektoren `$NK_OFF_DIR`, die die Achsrichtungen definieren, wird jeweils nur die einzige, von Null verschiedene Komponente beschrieben. Alle Linearachsen haben ihren Ursprung im gleichen Punkt, da zwischen den einzelnen Achsen keine konstanten Elemente definiert sind, und für die Achsen auch keine Nullpunktverschiebungen angegeben wurden.
- Das C-Achsoffset ist ein konstantes Element, das den Abstand zwischen den Linearachsen X, Y, Z und der ersten Rundachse (C-Achse) definiert. Die C-Achse ist eine Rundachse, die in Z-Richtung zeigt.
- Das B-Achsoffset ist ein konstantes Element, an das sich die B-Achse anschließt.
- Den Abschluss bildet das konstante Offset "Basetool".
- Die kinematische Kette zum Werkstückbezugspunkt wird durch das konstante Element "Tischoffset" vom Typ Offset gebildet.

9.4 Beispiele

- Folgende Orientierungsachsen werden definiert:
 - C-Achse
 - B-Achse
- Folgende lineare Ausgleichsachsen werden definiert:
 - X-Achse
 - Y-Achse
 - Z-Achse

9.4.2 Teileprogramm für TRAORI_DYN

Beispielprogramm "5-Achs-Trafo C-B"

Programmcode

```

;=====
; Definitionen
;=====
N10 DEF INT KIE_CNTR = 0 ; Zaehler für Elemente der kin. Ketten
;=====
; Löschen aller Transformationsdatensätze und kinematischen Kettenelemente
;=====
N20 IF (DELOBJ("TRAFO_DATA") < 0)
N30     SETAL(61000)
N40 ENDIF

N50 IF (DELOBJ("KIN_CHAIN_ELEM") < 0)
N60     SETAL(61001)
N70 ENDIF

;=====
; Definition des Root-Elements und der kinematischen Kette zum
Werkzeugbezugspunkt
;=====
N80 $NK_NAME[KIE_CNTR]      = "ROOT"
N90 $NK_TYPE[KIE_CNTR]     = "OFFSET"
N100 $NK_NEXT[KIE_CNTR]    = "X-Achse"
N110 $NK_PARALLEL[KIE_CNTR] = "Tischoffset"
N120 KIE_CNTR = KIE_CNTR + 1
;=====
; Definition der Linearachse in X-Richtung
;=====
N130 $NK_NAME[KIE_CNTR]    = "X-Achse"
N140 $NK_TYPE[KIE_CNTR]   = "AXIS_LIN"
N150 $NK_NEXT[KIE_CNTR]   = "Y-Achse"
N160 $NK_AXIS[KIE_CNTR]   = "X1"
N170 $NK_OFF_DIR[KIE_CNTR,0] = 1.0
N180 KIE_CNTR = KIE_CNTR + 1
;=====
; Definition der Linearachse in Y-Richtung
;=====
N190 $NK_NAME[KIE_CNTR]    = "Y-Achse"
N200 $NK_TYPE[KIE_CNTR]   = "AXIS_LIN"
N210 $NK_NEXT[KIE_CNTR]   = "Z-Achse"
N220 $NK_AXIS[KIE_CNTR]   = "Y1"
N230 $NK_OFF_DIR[KIE_CNTR,1] = 1.0
N240 KIE_CNTR = KIE_CNTR + 1
;=====
; Definition der Linearachse in Z-Richtung
;=====
N250 $NK_NAME[KIE_CNTR]    = "Z-Achse"
N260 $NK_TYPE[KIE_CNTR]   = "AXIS_LIN"
N270 $NK_NEXT[KIE_CNTR]   = "C-Achsoffset"
N280 $NK_AXIS[KIE_CNTR]   = "Z1"
N290 $NK_OFF_DIR[KIE_CNTR,2] = 1.0
N300 KIE_CNTR = KIE_CNTR + 1
;=====
; Definition des Abstands zwischen den Linearachsen und der ersten Rundachse
;=====
N310 $NK_NAME[KIE_CNTR]    = "C-Achsoffset"
N320 $NK_TYPE[KIE_CNTR]   = "OFFSET"

```

Programmcode

```

N330 $NK_NEXT[KIE_CNTR]          = "C-Achse"
N340 $NK_OFF_DIR[KIE_CNTR,0]     = 200.0

N350 $NK_OFF_DIR[KIE_CNTR,2]    = 300.0
N360 KIE_CNTR = KIE_CNTR + 1
;=====
; Definition der C-Achse in Z-Richtung - verweist auf die Achse C1
;=====
N370 $NK_NAME[KIE_CNTR]         = „C-Achse“
N380 $NK_TYPE[KIE_CNTR]        = "AXIS_ROT"
N390 $NK_NEXT[KIE_CNTR]        = "B-Achsoffset"
N400 $NK_AXIS[KIE_CNTR]        = „C1“
N410 $NK_OFF_DIR[KIE_CNTR,2]    = 1.0
N420 KIE_CNTR = KIE_CNTR + 1
;=====
; Definition der B-Achse mit einem Offset zwischen der B- und C-Achse
;=====
N430 $NK_NAME[KIE_CNTR]        = „B-Achsoffset“
N440 $NK_TYPE[KIE_CNTR]        = "OFFSET"
N450 $NK_NEXT[KIE_CNTR]        = "B-Achse"
N460 $NK_OFF_DIR[KIE_CNTR,2]    = -150.0
N470 KIE_CNTR = KIE_CNTR + 1

N480 $NK_NAME[KIE_CNTR]        = „B-Achse“
N490 $NK_TYPE[KIE_CNTR]        = "AXIS_ROT"
N500 $NK_NEXT[KIE_CNTR]        = "Basetool"
N510 $NK_AXIS[KIE_CNTR]        = „B1“
N520 $NK_OFF_DIR[KIE_CNTR,0]    = 1.0
N530 KIE_CNTR = KIE_CNTR + 1
;=====
; Abschluss der kinematischen Kette mit einem Offset
;=====
N540 $NK_NAME[KIE_CNTR]        = "Basetool"
N550 $NK_TYPE[KIE_CNTR]        = "OFFSET"
N560 $NK_NEXT[KIE_CNTR]        = ""
N570 $NK_OFF_DIR[KIE_CNTR,2]    = -50.0
;=====
; Definition der kinematischen Kette zum Tischbezugspunkt
;=====
N580 KIE_CNTR = KIE_CNTR + 1
N590 $NK_NAME[KIE_CNTR]        = „Tischoffset“
N600 $NK_TYPE[KIE_CNTR]        = "OFFSET"
N610 $NK_OFF_DIR[KIE_CNTR,0]    = 200.0
N620 $NK_OFF_DIR[KIE_CNTR,2]    = 100.0
N630 KIE_CNTR = KIE_CNTR + 1
;=====
; Definition der Transformation (TRAORI)
;=====
N640 $NT_NAME[1]                = „5-Achs-Trafo C-B“
N650 $NT_T_CHAIN_LAST_ELEM[1]  = "Basetool"
N660 $NT_P_CHAIN_LAST_ELEM[1]  = "Tischoffset"
N670 $NT_TRAFO_TYPE[1]         = "TRAORI_DYN"
;=====
; Definition der Orientierungs- und Geometrieachsen
;=====
N680 $NT_ROT_AX_NAME[1,0]      = "C-Achse"

```

9.4 Beispiele

Programmcode

```
N690 $NT_ROT_AX_NAME[1,1] = „B-Achse“
N700 $NT_GEO_AX_NAME[1,0] = „X-Achse“
N710 $NT_GEO_AX_NAME[1,1] = „Y-Achse“
N720 $NT_GEO_AX_NAME[1,2] = „Z-Achse“
;===== ENDE =====
```

9.4.3 Teileprogramm für TRANSMIT

Beispielprogramm "5-Achs-Trafo C-B"

Programmcode

```

;=====
; Einfaches Beispiel für TRANSMIT mit kinematischer Kette:
;*****
N10     DEF INT _KIE_CNTR
N20     DEF INT _TRA_CNTR
N30     R2 = DELOBJ("TRAFO_DATA")
N40     R2 = DELOBJ("KIN_CHAIN_ELEM")
N50     _KIE_CNTR = 0
N60     _TRA_CNTR = 1
; Definition der kinematischen Kette
;*****
N70     $NK_NAME[_KIE_CNTR]           = "ROOT"
N80     $NK_TYPE[_KIE_CNTR]          = "OFFSET"
N90     $NK_NEXT[_KIE_CNTR]          = "X-Axis"
N100    $NK_PARALLEL[_KIE_CNTR]      = "C-Axis"
N110    _KIE_CNTR = _KIE_CNTR + 1
N120    $NK_NAME[_KIE_CNTR]          = "X-Axis"
N130    $NK_TYPE[_KIE_CNTR]          = "AXIS_LIN"
N140    $NK_NEXT[_KIE_CNTR]          = "Z-Axis"
N150    $NK_AXIS[_KIE_CNTR]          = "X1"
N160    $NK_OFF_DIR[_KIE_CNTR,0]     = 1.0
N170    _KIE_CNTR = _KIE_CNTR + 1
N180    $NK_NAME[_KIE_CNTR]          = "Z-Axis"
N190    $NK_TYPE[_KIE_CNTR]          = "AXIS_LIN"
N200    $NK_NEXT[_KIE_CNTR]          = "Y-Axis"
N210    $NK_AXIS[_KIE_CNTR]          = "Z1"
N220    $NK_OFF_DIR[_KIE_CNTR,2]     = 1.0
N230    _KIE_CNTR = _KIE_CNTR + 1
N240    $NK_NAME[_KIE_CNTR]          = "Y-Axis"
N250    $NK_TYPE[_KIE_CNTR]          = "AXIS_LIN"
N260    $NK_NEXT[_KIE_CNTR]          = ""
N270    $NK_AXIS[_KIE_CNTR]          = "Y1"
N280    $NK_OFF_DIR[_KIE_CNTR,1]     = 1.0
N290    _KIE_CNTR = _KIE_CNTR + 1
N300    $NK_NAME[_KIE_CNTR]          = "C-Axis"
N310    $NK_TYPE[_KIE_CNTR]          = "AXIS_ROT"
N320    $NK_NEXT[_KIE_CNTR]          = ""
N330    $NK_AXIS[_KIE_CNTR]          = "C1"
N340    $NK_OFF_DIR[_KIE_CNTR,2]     = -1.0
N350    _KIE_CNTR = _KIE_CNTR + 1

; Definition der kinematischen Transformation:
;*****
; 1. TRANSMIT 256
;*****
N360    $NT_NAME[_TRA_CNTR]           = "Trafo Transmit_1"
N370    $NT_TRAFO_TYPE[_TRA_CNTR]    = "TRANSMIT_K"
N380    $NT_P_CHAIN_LAST_ELEM[_TRA_CNTR] = "C-Axis"
N390    $NT_T_CHAIN_LAST_ELEM[_TRA_CNTR] = "Z-Axis"
N400    $NT_GEO_AX_NAME[_TRA_CNTR,0]  = "X-Axis"
N410    $NT_GEO_AX_NAME[_TRA_CNTR,1]  = ""
N420    $NT_GEO_AX_NAME[_TRA_CNTR,2]  = "Z-Axis"
N430    $NT_ROT_AX_NAME[_TRA_CNTR,0]  = ""
N440    $NT_ROT_AX_NAME[_TRA_CNTR,1]  = "C-Axis"

```

Programmcode

```

N450   $NT_ROT_AX_NAME[_TRA_CNTR,2]           = ""
N460   $NT_ROT_OFFSET_FROM_FRAME[_TRA_CNTR]  = 1
N470   $NT_CNTRL[_TRA_CNTR]                  = 'H0'
N480   _TRA_CNTR = _TRA_CNTR + 1
; 2. TRANSMIT 257
;*****
N490   $NT_NAME[_TRA_CNTR]                    = "Trafo Transmit_2"
N500   $NT_TRAFO_TYPE[_TRA_CNTR]              = "TRANSMIT_K"
N510   $NT_P_CHAIN_LAST_ELEM[_TRA_CNTR]      = "C-Axis"
N520   $NT_T_CHAIN_LAST_ELEM[_TRA_CNTR]      = "Y-Axis"
N530   $NT_GEO_AX_NAME[_TRA_CNTR,0]          = "X-Axis"
N540   $NT_GEO_AX_NAME[_TRA_CNTR,1]          = "Y-Axis"
N550   $NT_GEO_AX_NAME[_TRA_CNTR,2]          = "Z-Axis"
N560   $NT_ROT_AX_NAME[_TRA_CNTR,0]          = ""
N570   $NT_ROT_AX_NAME[_TRA_CNTR,1]          = "C-Axis"
N580   $NT_ROT_AX_NAME[_TRA_CNTR,2]          = ""
N590   $NT_ROT_OFFSET_FROM_FRAME[_TRA_CNTR]  = 1
N600   $NT_CNTRL[_TRA_CNTR]                  = 'H200' ; TRANSMIT 257
N610   _TRA_CNTR = _TRA_CNTR + 1

;*****
; Aktivieren der kinematischen Kette und Transformation:
;*****
N620   NEWCONF
N630   STOPRE
;*****
*****

;===== ENDE =====

```

9.4.4 Teileprogramm für TRACYL

Beispielprogramm "TRACYL"

Programmcode

```

;=====
; Einfaches Beispiel für TRACYL mit kinematischer Kette:
;*****
N640  $NK_NAME[_KIE_CNTR]      = "ROOT"
N650  $NK_TYPE[_KIE_CNTR]     = "OFFSET"
N660  $NK_NEXT[_KIE_CNTR]    = "Z-Axis"
N670  $NK_PARALLEL[_KIE_CNTR] = "C-Axis"
N680  _KIE_CNTR = _KIE_CNTR + 1
N690  $NK_NAME[_KIE_CNTR]     = "Z-Axis"
N700  $NK_TYPE[_KIE_CNTR]     = "AXIS_LIN"
N710  $NK_NEXT[_KIE_CNTR]    = "X-Axis"
N720  $NK_AXIS[_KIE_CNTR]    = "Z1"
N730  $NK_OFF_DIR[_KIE_CNTR,2] = 1.0
N740  _KIE_CNTR = _KIE_CNTR + 1
N750  $NK_NAME[_KIE_CNTR]     = "X-Axis"
N760  $NK_TYPE[_KIE_CNTR]     = "AXIS_LIN"
N770  $NK_NEXT[_KIE_CNTR]    = "Y-Axis"
N780  $NK_AXIS[_KIE_CNTR]    = "X1"
N790  $NK_OFF_DIR[_KIE_CNTR,0] = 1.0
N800  _KIE_CNTR = _KIE_CNTR + 1
N810  $NK_NAME[_KIE_CNTR]     = "Y-Axis"
N820  $NK_TYPE[_KIE_CNTR]     = "AXIS_LIN"
N830  $NK_NEXT[_KIE_CNTR]    = ""
N840  $NK_AXIS[_KIE_CNTR]    = "Y1"
N850  $NK_OFF_DIR[_KIE_CNTR,1] = 1.0
N860  _KIE_CNTR = _KIE_CNTR + 1
N870  $NK_NAME[_KIE_CNTR]     = "C-Axis"
N880  $NK_TYPE[_KIE_CNTR]     = "AXIS_ROT"
N890  $NK_NEXT[_KIE_CNTR]    = ""
N900  $NK_AXIS[_KIE_CNTR]    = "C1"
N910  $NK_OFF_DIR[_KIE_CNTR,2] = -1.0

; Definition der kinematischen Transformation:
;*****
; 1. TRACYL 512
;*****
N920  $NT_NAME[_TRA_CNTR]      = "Trafo Tracyl 512"
N930  $NT_TRAFO_TYPE[_TRA_CNTR] = "TRACYL_K"
N940  $NT_T_CHAIN_LAST_ELEM[_TRA_CNTR] = "X-Axis"
N950  $NT_P_CHAIN_LAST_ELEM[_TRA_CNTR] = "C-Axis"
N960  $NT_GEO_AX_NAME[_TRA_CNTR,0] = "X-Axis"
N970  $NT_GEO_AX_NAME[_TRA_CNTR,1] = ""
N980  $NT_GEO_AX_NAME[_TRA_CNTR,2] = "Z-Axis"
N990  $NT_ROT_AX_NAME[_TRA_CNTR,0] = ""
N1000 $NT_ROT_AX_NAME[_TRA_CNTR,1] = "C-Axis"
N1010 $NT_ROT_AX_NAME[_TRA_CNTR,2] = ""
N1020 $NT_ROT_OFFSET_FROM_FRAME[_TRA_CNTR] = 1
N1030 $NT_CNTRL[_TRA_CNTR]     = 'H0'
N1040 _TRA_CNTR = _TRA_CNTR + 1
; 2. TRACYL 513
;*****
N1050 $NT_NAME[_TRA_CNTR]      = "Trafo Tracyl 513"
N1060 $NT_TRAFO_TYPE[_TRA_CNTR] = "TRACYL_K"
N1070 $NT_T_CHAIN_LAST_ELEM[_TRA_CNTR] = "Y-Axis"

```

Programmcode

```

N1080 $NT_P_CHAIN_LAST_ELEM[_TRA_CNTR] = "C-Axis"
N1090 $NT_GEO_AX_NAME[_TRA_CNTR,0] = "X-Axis"
N1100 $NT_GEO_AX_NAME[_TRA_CNTR,1] = "Y-Axis"
N1110 $NT_GEO_AX_NAME[_TRA_CNTR,2] = "Z-Axis"
N1120 $NT_ROT_AX_NAME[_TRA_CNTR,0] = ""
N1130 $NT_ROT_AX_NAME[_TRA_CNTR,1] = "C-Axis"
N1140 $NT_ROT_AX_NAME[_TRA_CNTR,2] = ""
N1150 $NT_ROT_OFFSET_FROM_FRAME[_TRA_CNTR] = 1
N1160 $NT_CNTRL[_TRA_CNTR] = 'H200' ; TRACYL 513
N1170 _TRA_CNTR = _TRA_CNTR + 1
; 3. TRACYL 514
;*****
N1180 $NT_NAME[_TRA_CNTR] = "Trafo Tracyl 514"
N1190 $NT_TRAFO_TYPE[_TRA_CNTR] = "TRACYL_K"
N1200 $NT_T_CHAIN_LAST_ELEM[_TRA_CNTR] = "Y-Axis"
N1210 $NT_P_CHAIN_LAST_ELEM[_TRA_CNTR] = "C-Axis"
N1220 $NT_GEO_AX_NAME[_TRA_CNTR,0] = "X-Axis"
N1230 $NT_GEO_AX_NAME[_TRA_CNTR,1] = "Y-Axis"
N1240 $NT_GEO_AX_NAME[_TRA_CNTR,2] = "Z-Axis"
N1250 $NT_ROT_AX_NAME[_TRA_CNTR,0] = ""
N1260 $NT_ROT_AX_NAME[_TRA_CNTR,1] = "C-Axis"
N1270 $NT_ROT_AX_NAME[_TRA_CNTR,2] = ""
N1280 $NT_ROT_OFFSET_FROM_FRAME[_TRA_CNTR] = 1
N1290 $NT_CNTRL[_TRA_CNTR] = 'H400' ; TRACYL 514
N1300 _TRA_CNTR = _TRA_CNTR + 1

;*****
;Aktivieren der kinematischen Kette und Transformation:
;*****
N1310 NEWCONF
N1320 STOPRE
;*****
*****
;===== ENDE =====

```

9.4.5 Teileprogramm für TRAANG

Beispielprogramm "TRAANG"

Programmcode

```

;=====
; Einfaches Beispiel für TRAANG mit kinematischer Kette
;*****
N2000 $NK_NAME[_KIE_CNTR] = "ROOT"
N2010 $NK_TYPE[_KIE_CNTR] = "OFFSET"
N2020 $NK_NEXT[_KIE_CNTR] = "X-Axis"
N2030 $NK_PARALLEL[_KIE_CNTR] = ""
N2040 _KIE_CNTR = _KIE_CNTR + 1
N2050 $NK_NAME[_KIE_CNTR] = "X-Axis"
N2060 $NK_TYPE[_KIE_CNTR] = "AXIS_LIN"
N2070 $NK_NEXT[_KIE_CNTR] = "Z-Axis"
N2080 $NK_AXIS[_KIE_CNTR] = "X1"
N2090 $NK_OFF_DIR[_KIE_CNTR,0] = COS(20)
N2100 $NK_OFF_DIR[_KIE_CNTR,2] = SIN(20)
N2110 _KIE_CNTR = _KIE_CNTR + 1
N2120 $NK_NAME[_KIE_CNTR] = "Z-Axis"
N2130 $NK_TYPE[_KIE_CNTR] = "AXIS_LIN"
N2140 $NK_NEXT[_KIE_CNTR] = "Y-Axis"
N2150 $NK_AXIS[_KIE_CNTR] = "Z1"
N2160 $NK_OFF_DIR[_KIE_CNTR,2] = 1.0
N2170 _KIE_CNTR = _KIE_CNTR + 1
N2180 $NK_NAME[_KIE_CNTR] = "Y-Axis"
N2190 $NK_TYPE[_KIE_CNTR] = "AXIS_LIN"
N2200 $NK_NEXT[_KIE_CNTR] = ""
N2210 $NK_AXIS[_KIE_CNTR] = "Y1"
N2220 $NK_OFF_DIR[_KIE_CNTR,1] = 1.0
N2230 _KIE_CNTR = _KIE_CNTR + 1

; Definition der kinematischen Transformation:
;*****
; TRRANG
;*****
N2240 $NT_NAME[_TRA_CNTR] = "Trafo Traang"
N2250 $NT_TRAFO_TYPE[_TRA_CNTR] = "TRAANG_K"
N2260 $NT_T_CHAIN_LAST_ELEM[_TRA_CNTR] = "Y-Axis"
N2270 $NT_P_CHAIN_LAST_ELEM[_TRA_CNTR] = ""
N2280 $NT_GEO_AX_NAME[_TRA_CNTR,0] = "X-Axis"
N2290 $NT_GEO_AX_NAME[_TRA_CNTR,2] = "Y-Axis"
N2300 $NT_GEO_AX_NAME[_TRA_CNTR,1] = "Z-Axis"
N2310 $NT_ROT_OFFSET_FROM_FRAME[_TRA_CNTR] = 1
N2320 _TRA_CNTR = _TRA_CNTR + 1

;*****
; Aktivieren der kinematischen Kette und Transformation:
;*****
N2330 NEWCONF
N2340 STOPRE
;*****
*****

```

Programmcode

;===== ENDE =====

M3: Achskopplungen

10.1 Mitschleppen

10.1.1 Kurzbeschreibung

10.1.1.1 Funktion

Die Funktion "Mitschleppen" ermöglicht die Definition einfacher Achskopplungen von einer Leitachse auf eine Folgeachse unter Berücksichtigung eines Koppelfaktors.

Die Funktion Mitschleppen besitzt folgende Eigenschaften:

- Jede Achse der NC kann als Leitachse definiert werden.
- Jede Achse der NC kann als Mitschleppachse mit einem spezifischen Koppelfaktor definiert werden.
- Leitachse und Mitschleppachse(n) bilden einen Mitschleppverband.
- Einer Leitachse können beliebig viele Mitschleppachsen zugeordnet werden.
- Einer Mitschleppachse können maximal 2 Leitachsen zugeordnet sein.
- Eine Mitschleppachse kann Leitachse eines weiteren Mitschleppverbands sein.
- Verfahrbewegungen der Leitachse werden, bewertet mit dem Koppelfaktor, synchron von allen Folgeachsen ausgeführt.
- Mitschleppachsen können auch bei aktiver Kopplung Verfahrbewegungen unabhängig von der Leitachse ausführen (überlagerte Verfahrbewegungen).
- Die Definition der Leit- und Mitschleppachse(n) eines Mitschleppverbandes sowie das Ein-/Ausschalten der Kopplung erfolgt durch Teileprogrammanweisungen im Teileprogramm oder Synchronaktion.
- Mitschleppen ist auch in den Handbetriebsarten JOG, JOG REF, JOG INC etc. möglich.

10.1.1.2 Voraussetzungen

Funktion "Mitschleppen"

Die Funktion "Mitschleppen" ist fester Bestandteil der NC-Software.

Generische Kopplung

Die Mitschleppen-Funktionalität steht auch in der Generischen Kopplung zur Verfügung.

Für die **Grundaufführung** der Generischen Kopplung gelten allerdings folgende Einschränkungen:

- Die maximale Anzahl von Mitschleppverbänden ist auf 4 begrenzt.
- Einer Mitschleppachse kann nur 1 Leitachse zugeordnet sein.
- Eine Kaskadierung ist nicht möglich.

Hinweis

Diese Einschränkungen gelten nicht, wenn die NC-Software mit der entsprechenden optionalen Ausführung der Generischen Kopplung ausgestattet ist (siehe Thema "Voraussetzungen (Seite 484)" in der "Kurzbeschreibung" zur Generischen Kopplung).

10.1.2 Allgemeine Funktionalität

Die Funktion "Mitschleppen" ermöglicht die Definition einfacher Achskopplungen. Die Kopplung erfolgt von einer Leitachse auf eine oder mehrere Folgeachsen, den so genannten Mitschleppachsen. Für jede Mitschleppachse kann ein eigener Koppelfaktor vorgegeben werden.

Mitschleppverband

Die Leitachse und alle ihr zugeordneten Mitschleppachsen bilden zusammen einen Mitschleppverband. Wird die Leitachse verfahren, verfahren alle Mitschleppachsen entsprechend ihrer Koppelfaktoren.

Ein Mitschleppverband kann aus einer beliebigen Kombination von Linear- und Rundachsen bestehen.

Leitachsen

Jede Achse der NC, einschließlich simulierten Achsen, kann als Leitachse verwendet werden.

Mitschleppachsen

Jede Achse der NC kann als Mitschleppachse verwendet werden.

Koppelfaktor

Über den Koppelfaktor wird das Verhältnis angegeben, in dem sich die Mitschleppachse in Bezug auf die Leitachse bewegt:

Koppelfaktor $K = \text{Bewegung der Mitschleppachse} / \text{Bewegung der Leitachse}$

Es sind auch negative Koppelfaktoren (gegensinniges Verfahren der Mitschleppachse) zulässig.

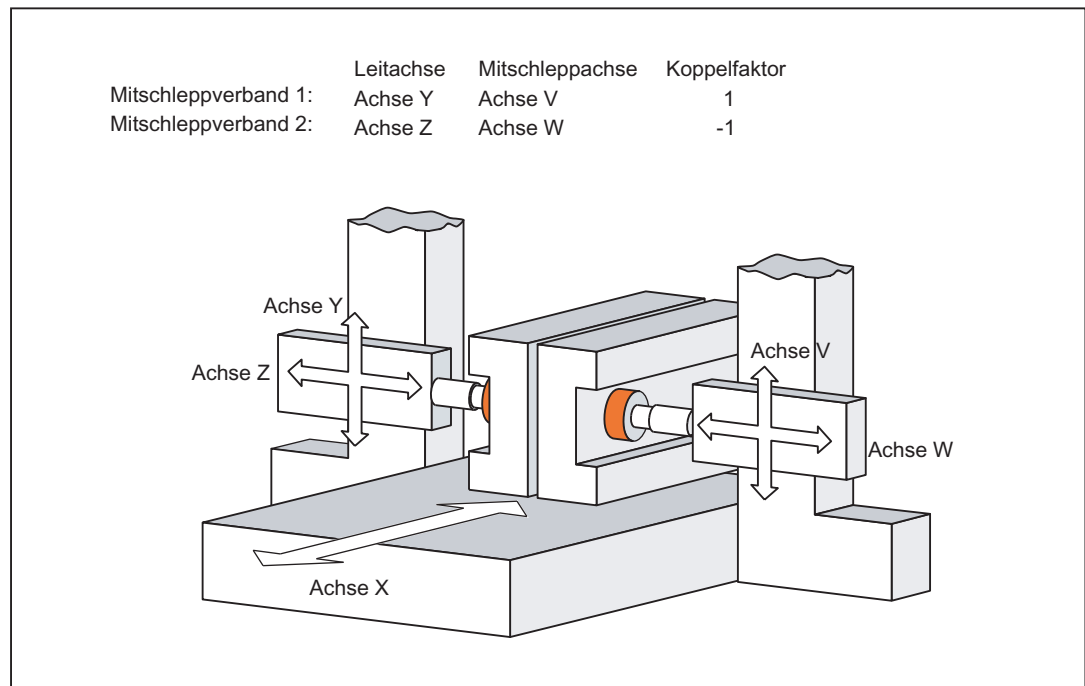


Bild 10-1 Anwendungsbeispiel: Zweiseitenbearbeitung

Mehrfach-Kopplungen

Einer Mitschleppachse können bis zu 2 Leitachsen zugeordnet werden. Die Verfahrbewegung der Mitschleppachse ergibt sich dabei aus der Summe der Verfahrbewegungen der Leitachsen.

Abhängige Mitschleppachse

Eine Mitschleppachse ist "abhängige Mitschleppachse", wenn sie aufgrund einer Leitachsbewegung verfährt.

Unabhängige Mitschleppachse

Eine Mitschleppachse ist "unabhängige Mitschleppachse", wenn sie aufgrund einer direkten Verfahrweisung verfahren wird. Die resultierende Verfahrbewegung der Mitschleppachse ergibt sich dann aus der Summe der Verfahrbewegungen als "abhängige" und "unabhängige" Mitschleppachse.

Mitschleppachse als Leitachse

Eine Mitschleppachse kann gleichzeitig Leitachse eines anderen Mitschleppverbandes sein.

Koordinatensystem

Das Mitschleppen erfolgt immer im Basiskoordinatensystem (BKS).

Ein-/Ausschalten

Mitschleppen kann gleichermaßen über Teileprogramme und Synchronaktionen ein- und ausgeschaltet werden. Dabei ist zu beachten, dass das Ein- und Ausschalten über die gleiche Programmierung erfolgt:

- Einschalten: Teileprogramm → Ausschalten: Teileprogramm
- Einschalten: Synchronaktion → Ausschalten: Synchronaktion

Fliegende Synchronisation

Erfolgt das Einschalten während die Leitachse in Bewegung ist, wird die Mitschleppachse zunächst auf die der Kopplung entsprechende Geschwindigkeit beschleunigt. Die Position der Leitachse zum Zeitpunkt der Synchronisation der Geschwindigkeiten von Leit- und Mitschleppachse gilt dann als Startposition für das weitere Mitschleppen.

Betriebsarten

Mitschleppen ist in den Betriebsarten AUTOMATIK, MDA und JOG wirksam.

Referenzpunktfahren

Beim Referenzpunktfahren von Achsen eines Mitschleppverbandes gilt:

- Leitachsen
Beim Referenzpunktfahren der Leitachse eines Mitschleppverbandes bleibt die Kopplung zu allen Mitschleppachsen erhalten. Die Mitschleppachsen verfahren entsprechend ihrer Koppelfaktoren synchron zur Leitachse.
- Mitschleppachse: Betriebsart JOG/REF
Beim Referenzpunktfahren einer Mitschleppachse eines Mitschleppverbandes wird die Kopplung zur Leitachse aufgehoben. Das Aufheben der Kopplung wird über folgenden Alarm angezeigt:
Alarm 16772 "Kanal Kanal-Nr. Satz Satz-Nr. Achse Achse-Nr. ist Folgeachse, Kopplung wird geöffnet."
Die Kopplung wird erst mit Verlassen der Betriebsart JOG/REF wieder aktiv.
Die Anzeige des Alarms kann über folgendes Maschinendatum unterdrückt werden:
MD11410 \$MN_SUPPRESS_ALARM_MASK, Bit 29 = 1 (Maske zur Unterstützung spezieller Alarmausgaben)

 VORSICHT

Keine Kopplung

Mit dem Referenzpunktfahren der Mitschleppachse wird die Kopplung mit der Leitachse aufgehoben. Wird jetzt sofort, d. h. ohne Wechsel der Betriebsart JOG/REF, ein Referenzpunktfahren mit der Leitachse durchgeführt, verfährt die Mitschleppachse nicht mit der Leitachse.
--

- Mitschleppachse: Teileprogrammanweisung G74
Referenzpunktfahren einer Mitschleppachse eines Mitschleppverbandes per Teileprogrammanweisung G74 ist nicht möglich.

Restweg: Mitschleppachse

Der Restweg einer Mitschleppachse bezieht sich auf den gesamten zu verfahrenen Restweg aus abhängiger und unabhängiger Verfahrbewegung.

Restweglöschen: Mitschleppachse

Restweglöschen für eine Mitschleppachse bewirkt nur den Abbruch der von der Leitachse unabhängigen Verfahrbewegung.

Verhalten bei NC-Start

Das Verhalten von Mitschleppverbänden bei NC-Start ist abhängig von der Einstellung im Maschinendatum:

MD20112 \$MC_START_MODE_MASK (Festlegung der Steuerungs-Grundstellung bei NC-START)

Bit	Wert	Bedeutung
8	0	Mitschleppverbände bleiben bei NC-Start erhalten.
	1	Mitschleppverbände werden bei NC-Start aufgelöst.

Verhalten bei RESET/Teileprogrammende

Das Verhalten von Mitschleppverbänden bei RESET/Teileprogrammende ist abhängig von der Einstellung im Maschinendatum:

MD20110 \$MC_RESET_MODE_MASK (Festlegung der Steuerungs-Grundstellung nach RESET/TP-Ende)

Bit	Wert	Bedeutung
8	0	Mitschleppverbände werden bei RESET/Teileprogrammende aufgelöst.
	1	Mitschleppverbände bleiben über RESET/Teileprogrammende hinweg aktiv.

Hinweis

Wird bei NC-RESET bzw. Teileprogrammende in einem Kanal die Leitachse nicht mit stillgesetzt (kanalübergreifende Kopplung, Kommandoachse, PLC-Achse, etc.), kann der angeforderte RESET nicht abgeschlossen werden.

Aufgrund der Verfahrbewegungen der Leitachse ist für den Kanal, in dem der RESET angefordert ist, die Mitschleppachse weiterhin aktiv. Durch geeignete Maßnahmen (NC-RESET im Kanal der Leitachse, Stillsetzen der Kommando- oder PLC-Achse) muss parallel zur Mitschleppachse auch die Leitachse stillgesetzt werden.

10.1.3 Programmierung

10.1.3.1 Definition und Einschalten eines Mitschleppverbandes (TRAILON)

Definition und Einschalten eines Mitschleppverbandes erfolgen gleichzeitig mit dem Teileprogrammbehehl TRAILON.

Programmierung

Syntax: TRAILON(<Mitschleppachse>, <Leitachse>, [**<Koppelfaktor>**])

Wirksamkeit: modal

Parameter:

Mitschleppachse: Typ: AXIS
Wertebereich: Alle im Kanal definierten Achs- und Spindelname

Leitachse: Typ: AXIS
Wertebereich: Alle im Kanal definierten Achs- und Spindelname

Koppelfaktor: Über den optionalen Koppelfaktor wird das Verhältnis der Verfahrbewegung der Mitschleppachse zur Leitachse angegeben:
Koppelfaktor = Weg der Mitschleppachse/Weg der Leitachse
Ein negativer Koppelfaktor bewirkt ein gegensinniges Verfahren von Leit- und Mitschleppachse.

Typ: REAL
Wertebereich: ± (2,2 * 10⁻³⁰⁸ ... 1,8 * 10⁺³⁰⁸)
Standardwert: +1.0

Beispiel:

Programmcode	Kommentar
TRAILON(V,Y,2)	; Definition und Einschalten der Kopplung der Mitschleppachse V zur Leitachse Y. Der Koppelfaktor ist 2.

10.1.3.2 Ausschalten (TRAILOF)

Das Ausschalten der Kopplung einer Mitschleppachse zu einer Leitachse erfolgt über den Teileprogrammbehehl TRAILOF.

Programmierung

Syntax: TRAILOF(<Mitschleppachse>, <Leitachse>)
TRAILOF(<Mitschleppachse>)

Wirksamkeit: modal

Parameter:

Mitschleppachse: Typ: AXIS
 Wertebereich: Alle im Kanal definierten Achs- und Spindelname
 Leitachse: Typ: AXIS
 Wertebereich: Alle im Kanal definierten Achs- und Spindelname

Beispiel:

Programmcode	Kommentar
TRAILOF (V, Y)	; Ausschalten der Kopplung der Mitschleppachse
	; V zur Leitachse Y.

10.1.4 Wirksamkeit der PLC-Nahtstellensignale

Unabhängige Mitschleppachse

Für die unabhängige Bewegung einer Mitschleppachse sind alle zugehörigen kanal- und achsspezifischen Nahtstellensignale der Mitschleppachse wirksam, z. B.:

- DB21, ... DBX0.3 (DRF aktivieren)
- DB31, ... DBX0.0 - 0.7 (Vorschub-Override)
- DB31, ... DBX1.3 (Achsensperre)
- DB31, ... DBX2.1 (Reglerfreigabe)
- DB31, ... DBX4.0 - 4.2 (Handrad aktivieren)
- DB31, ... DBX4.3 (Vorschub Halt)
- ...

Für die unabhängige Bewegung einer Mitschleppachse kann dadurch z. B. über die Vorschub-Override die Geschwindigkeit verändert werden oder in den Betriebsarten AUTOMATIK und MDA mit dem Handrad eine DRF-Verschiebung vorgegeben werden.

Abhängige Mitschleppachse

Für die von einer Leitachse abhängige Bewegung einer Mitschleppachse sind nur die Nahtstellensignale der Mitschleppachse wirksam, die zu einem Stopp der Bewegung führen (z. B. achsspezifischer Vorschub Halt, Reglerfreigabe usw.)

Leitachse

Bei einem aktivierten Mitschleppverband wirken die Nahtstellensignale der Leitachse durch die Achskopplung auf die zugehörige Mitschleppachse, d. h.:

- Eine Positionsverschiebung oder Vorschubbeeinflussung der Leitachse bewirkt über den Koppelfaktor eine entsprechende Positionsverschiebung bzw. Vorschubbeeinflussung der Mitschleppachse.
- Ein Stillsetzen der Leitachse durch Nahtstellensignale (z. B. achsspezifischer Vorschub Halt, Reglerfreigabe, usw.) bewirkt ein Stillsetzen der zugehörigen Mitschleppachse.

Lagemesssystem 1/2 (DB31, ... DBX1.5/1.6)

Für einen aktivierten Mitschleppverband ist eine Umschaltung des Lagemesssystems für Leit- und Mitschleppachsen nicht verriegelt. Die Kopplung bleibt dabei erhalten.

Empfehlung: Die Umschaltung bei ausgeschalteter Kopplung vornehmen.

Nachführen (DB31, ... DBX1.4)

Die Aktivierung des Nachführbetriebs für eine Achse erfolgt über das PLC-Anwenderprogramm durch Setzen folgender NC/PLC-Nahtstellensignale:

DB31, ... DBB2.1 = 0 (Reglerfreigabe)

DB31, ... DBX1.4 = 1 (Nachführbetrieb)

Bei der Aktivierung des Nachführbetriebs eines Mitschleppverbandes müssen die genannten NC/PLC-Nahtstellensignale gleichzeitig für alle Achsen (Leit- und Folgeachsen) des Mitschleppverbandes gesetzt werden.

Erfolgt die Aktivierung des Nachführbetriebs nur für die Leitachse, entsteht innerhalb der Kopplung ein permanenter Versatz, der nicht mehr herausgefahren wird.

Ob und welche Achse Leit- oder Folgeachse ist, kann aus folgenden NC/PLC-Nahtstellensignalen und Systemvariablen ermittelt werden:

DB31, ... DBX99.0 (Leitachse/-spindel aktiv)

DB31, ... DBX99.1 (Folgeachse/-spindel aktiv)

\$AA_COUP_ACT [Achsenname] (siehe: Status der Kopplung)

10.1.5 Kopplungsstatus

Der Kopplungsstatus einer Achse kann über die folgenden Systemvariablen ermittelt werden:

\$AA_COUP_ACT [Achsenname]

Wert	Bedeutung
0	Keine Kopplung aktiv
1, 2, 3	Tangentiales Nachführen
4	Synchronspindelkopplung
8	Mitschleppen aktiv
16	Leitwertkopplung
32	Folgeachse des elektronischen Getriebes


Hinweis

Zu einem Zeitpunkt kann immer nur eine Kopplungsart aktiv sein.

10.1.6 Dynamikbegrenzung

Die Dynamikbegrenzung ist abhängig von der Art der Aktivierung des Mitschleppverbandes:

- **Aktivierung im Teileprogramm**
Erfolgt die Aktivierung im Teileprogramm und sind alle Leitachsen als Programmachsen im aktivierenden Kanal, wird beim Verfahren der Leitachsen die Dynamik aller Mitschleppachsen so berücksichtigt, dass keine Mitschleppachse überlastet wird. Erfolgt die Aktivierung im Teileprogramm mit Leitachsen, die nicht als Programmachsen im aktivierenden Kanal aktiv sind ($\$AA_TYP \neq 1$), wird beim Verfahren der Leitachsen die Dynamik der Mitschleppachse nicht berücksichtigt. Dadurch kann es bei Mitschleppachsen mit einer geringeren als der für die Kopplung benötigten Dynamik zu einer Überlastung kommen.
- **Aktivierung in Synchronaktion**
Erfolgt die Aktivierung in einer Synchronaktion, wird beim Verfahren der Leitachsen die Dynamik der Mitschleppachsen nicht berücksichtigt. Dadurch kann es bei Mitschleppachsen mit einer geringeren als der für die Kopplung benötigten Dynamik zu einer Überlastung kommen.

 VORSICHT
<p>Achsüberlastung</p> <p>Wird ein Mitschleppverband</p> <ul style="list-style-type: none"> • in Synchronaktionen • im Teileprogramm mit Leitachsen, die nicht Programmachsen im Kanal der Mitschleppachse sind, <p>aktiviert, dann liegt es in der besonderen Verantwortung des Anwenders/ Maschinenherstellers, geeignete Maßnahmen vorzusehen, damit es durch die Verfahrbewegungen der Leitachse nicht zu einer Überlastung der Mitschleppachsen kommt.</p>

10.1.7 Randbedingungen

Regeldynamik

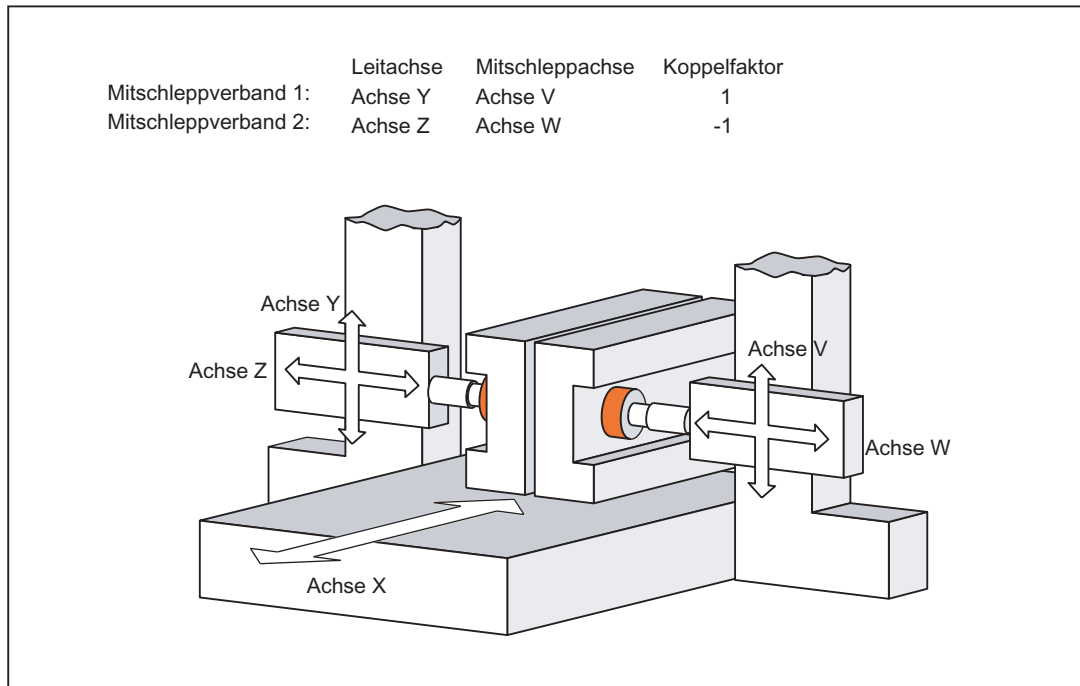
Es wird empfohlen, innerhalb eines Mitschleppverbandes die Lageregler-Parameter von Leitachse und Mitschleppachse aufeinander abzustimmen.

Hinweis

Das Abstimmen der Lageregler-Parameter von Leitachse und Mitschleppachse kann dynamisch über eine Parametersatzumschaltung vorgenommen werden.

10.1.8 Beispiele

Anwendungsbeispiel: Zweiseitenbearbeitung



Beispiel 1

Beispiel für ein NC-Teileprogramm für die im Bild dargestellte Achskonstellation:

Programmcode	Kommentar
TRAILON(V,Y,1)	; Einschalten des 1. Mitschleppverbandes
TRAILON(W,Z,-1)	; Einschalten des 2. Mitschleppverbandes
G0 Z10	; Zustellung der Z- und W-Achse in entgegengesetzter Achsrichtung
G0 Y20	; Zustellung der Y- und V-Achse in gleicher Achsrichtung
G1 Y22 V25	; Überlagerung einer abhängigen und unabhängigen Bewegung der Mitschleppachse "v"
TRAILOF(V,Y)	; Ausschalten des 1. Mitschleppverbandes
TRAILOF(W,Z)	; Ausschalten des 2. Mitschleppverbandes

Beispiel 2

Beim Mitschleppen addieren sich die abhängige und die unabhängige Bewegung einer Mitschleppachse. Der abhängige Anteil kann dabei als eine Koordinatenverschiebung bezüglich der Mitschleppachse betrachtet werden.

Programmcode	Kommentar
N01 G90 G0 X100 U100	;
N02 TRAILON(U,X,1)	; Einschalten Mitschleppverband
N03 G1 F2000 X200	; Abhängige Bewegung von U, Upos=200, UTrail=100
N04 U201	; Unabhängige Bewegung Upos=U201+UTrail=301
N05 X250	; Abhängige Bewegung von U, UTrail=UTrail(100)+50=150, Upos=351
N06 G91 U100	; Unabhängige Bewegung, Upos(351)+U100=451
N07 G90 X0	; Abhängige Bewegung von U, Upos=Upos(451)-UTrail(250)=201
N10 TRAILOF(U,X)	

10.2 Kurventabellen - nur 840D sl**10.2.1 Kurzbeschreibung****10.2.1.1 Funktion**

Mit Hilfe der Funktion "Kurventabellen" ist es möglich, in einer Kurventabelle einen komplexen Bewegungsablauf einer Achse zu definieren.

Dabei kann jede beliebige Achse als Leitachse definiert und unter Berücksichtigung einer Kurventabelle eine Folgeachse bewegt werden.

Führungsgröße bei diesen Bewegungsabläufen ist ein abstrakter Leitwert, der von der Steuerung erzeugt oder von einer externen Größe (z. B. der simulierten Position einer Achse) abgeleitet wird.

Das Anlegen von Kurventabellen erfolgt über Teileprogrammsequenzen.

Die Kurventabellen im statischen NC-Speicher bleiben über das Teileprogramm-Ende und auch über POWER DOWN hinaus erhalten.

Die Kurventabellen können zur schnelleren Abarbeitung im dynamischen NC-Speicher abgelegt werden. Hierbei ist zu beachten, dass nach dem Hochlauf die Tabellen erneut eingelesen werden müssen.

Achsverbände mit Kurventabelle müssen unabhängig vom Speicherort der Kurventabelle nach POWER ON neu aktiviert werden.

Lineare Kurventabellensegmente werden in eigenen Bereichen Speicherplatz-sparend abgelegt.

10.2.1.2 Voraussetzungen

Speicherkonfiguration

Statischer NC-Speicher

Der Speicherplatz für Kurventabellen im statischen NC-Speicher wird definiert mit den Maschinendaten:

MD18400 \$MN_MM_NUM_CURVE_TABS (Anzahl der Kurventabellen)

MD18402 \$MN_MM_NUM_CURVE_SEGMENTS (Anzahl der Kurvensegmente)

MD18403 \$MN_MM_NUM_CURVE_SEG_LIN (Anzahl der linearen Kurvensegmente)

MD18404 \$MN_MM_NUM_CURVE_POLYNOMS (Anzahl der Kurventabellenpolynome)

Dynamischer NC-Speicher

Der Speicherplatz für Kurventabellen im dynamischen NC-Speicher wird definiert mit den Maschinendaten:

MD18406 \$MN_MM_NUM_CURVE_TABS_DRAM (Anzahl der Kurventabellen)

MD18408 \$MN_MM_NUM_CURVE_SEGMENTS_DRAM (Anzahl der Kurvensegmente)

MD18409 \$MN_MM_NUM_CURVE_SEG_LIN_DRAM (Anzahl der linearen Kurvensegmente)

MD18410 \$MN_MM_NUM_CURVE_POLYNOMS_DRAM (Anzahl der Kurventabellenpolynome)

10.2.2 Allgemeine Funktionalität

Kurventabelle

In der Kurventabelle wird ein funktionaler Zusammenhang zwischen einer Führungsgröße "Leitwert" und einem abstrakten Folgewert beschrieben.

In einem definierten Wertebereich des Leitwerts kann eindeutig zu jedem Leitwert die Folgegröße zugeordnet werden.

Kurvensegment

Der funktionale Zusammenhang kann in einzelne Abschnitte der Leitwertachse, so genannte Kurvensegmente, unterteilt werden.

Innerhalb eines Kurvensegments wird der Zusammenhang zwischen Leit- und Folgewert im Allgemeinen durch ein Polynom bis zu 3. Grades beschrieben. Es sind auch Polynome bis zu 5. Grades zulässig.

Literatur:

Programmierhandbuch Arbeitsvorbereitung

Kurvensegmente werden gebraucht, wenn:

- Polynome bzw. Kreise programmiert sind
- Spline aktiv ist
- Kompressor aktiv ist
- Polynome bzw. Kreise intern generiert werden (Fasen/Runden, Überschleifen mit G643, WRK usw.)

Werkzeugradiuskorrektur

Es stehen Kurventabellen zur Verfügung, bei denen in der Tabellendefinition Werkzeugradiuskorrektur auch dann angegeben werden kann, wenn in der Kurventabelle Polynomsätze oder Sätze ohne Bewegung für eine Achse bzw. Sprünge für die Folgeachse vorkommen (G41/G42/G40 in der Tabellendefinition).

Die äquidistante Kurve (Werkzeug-Mittelpunktsbahn der Werkzeugradiuskorrektur) einer aus Polynomen bestehenden Kurve kann nicht mehr exakt durch Polynome dargestellt werden. Daher muss in diesem Fall die zugehörige Kurventabelle stückweise durch Polynome angenähert werden. Dies hat zur Folge, dass in diesem Fall die Anzahl der Segmente der Kurventabelle nicht mehr mit der Anzahl der programmierten Segmente übereinstimmt. Die Anzahl der für die Kurventabelle benötigten Segmente ist durch die Krümmung der Kurve festgelegt. Je größer die Krümmung der programmierten Kurve ist, umso mehr Segmente werden für die Kurventabelle benötigt.

Aufgrund von Werkzeugradiuskorrektur bei Kurventabellen kann Speichermehrbedarf entstehen. Durch Auswahlmöglichkeit des Speichertyps muss das aber nicht zur Verknappung des Speicherplatzes im statischen NC-Speicher führen.

Auswahl des Speichertyps

Bei der Definition einer Kurventabelle kann festgelegt werden, ob die Kurventabelle im statischen oder im dynamischen NC-Speicher angelegt wird.

Hinweis

Tabellendefinitionen im statischen NC-Speicher stehen nach Steuerungshochlauf weiter zur Verfügung. Kurventabellen des dynamischen NC-Speichers müssen nach Steuerungshochlauf neu definiert werden.

10.2.3 Speicherorganisation

Speicherkonfiguration

Der für die Kurventabellen im statischen und dynamischen NC-Speicher zur Verfügung stehende Speicherplatz wird bei der Speicherkonfiguration festgelegt (siehe Kapitel "Speicherkonfiguration (Seite 423)").

Speicheroptimierung

Bei einer Kurventabelle mit linearen Segmenten können die linearen Segmente nur dann effizienter im Speicher abgelegt werden, wenn die beiden folgenden Maschinendaten > 0 sind:

MD18403 \$MC_MM_NUM_CURVE_SEG_LIN (Anzahl der linearen Kurvensegmente im statischen NC-Speicher)

MD18409 \$MC_MM_NUM_CURVE_SEG_LIN_DRAM (Anzahl der linearen Kurvensegmente im dynamischen NC-Speicher)

Werden mit diesen Maschinendaten keine Speicherbereiche angelegt, werden lineare Segmente platzverschwendend als Polynom-Segmente gespeichert.

Alarmer bei Speichermangel

Wenn über Maschinendaten sowohl Speicher für Tabellen mit linearen als auch mit Polynom-Segmenten konfiguriert ist und bei der Erzeugung einer linearen Tabelle der Speicher für lineare Segmente ausgeht, dann wird für die linearen Segmente der Speicher für Polynom-Segmente verwendet (sofern verfügbar). Es wird in diesem Fall Speicher "vergeudet", da ein Polynom-Segment zur Speicherung eines linearen Segments unnötig viel Speicher benötigt. Dieser Umstand wird durch einen Alarm mitgeteilt, bei dem auch die Anzahl der unnötig verbrauchten Polynom-Segmente angegeben wird. Der Alarm stellt nur eine **Warnung** dar und führt nicht zur Unterbrechung des Programms bzw. der Erzeugung der Kurventabelle.

Besteht eine Kurventabelle sowohl aus linearen Segmenten als auch aus Polynomen höheren Grades, wird zur Speicherung der Kurventabelle sowohl ein Speicherbereich für lineare Segmente als auch ein Speicherbereich für Polynom-Segmente benötigt. Bei Speichermangel in den entsprechenden Bereichen wird ein Alarm ausgegeben. Über die Alarmparameter ist zu erkennen, welche Ressourcen nicht ausreichen.

Zu wenig Speicher

Kann eine Kurventabelle nicht erzeugt werden, weil nicht ausreichend Speicher zur Verfügung steht, dann wird die neu angelegte Tabelle sofort mit Auftreten des Alarms wieder gelöscht.

Ist der Speicher zu klein, müssen zuerst eine oder mehrere nicht mehr benötigte Tabelle(n) mit `CTABDEL` gelöscht oder der Speicher über Maschinendaten neu konfiguriert werden.

Temporäre Kurventabelle

Beim Erzeugen einer neuen Kurventabelle wird im Speicher zunächst eine temporäre Kurventabelle angelegt und diese Satz für Satz ergänzt. Zum Schluss (`CTABEND`) wird die Tabelle auf Konsistenz geprüft. Nur wenn die Kurventabelle gültig ist, wird aus der temporären Tabelle eine Tabelle, die in einer Kopplung verwendet werden kann.

Gleiche Tabellen-Nr.

Eine zu erzeugende Kurventabelle kann die Nummer einer bereits existierenden Tabelle haben. Die neue Kurventabelle überschreibt dann die existierende Tabelle gleicher Nummer. Dieses wird nur dann gemacht, wenn die neue Kurventabelle keinen Fehler enthält. Bei einem Fehler in der neuen Kurventabelle bleibt die alte Tabelle erhalten.

Wenn der Anwender von der Möglichkeit Gebrauch machen möchte, eine existierende Kurventabelle zu überschreiben ohne diese vorher zu löschen, dann muss er den Tabellenspeicher so dimensionieren, dass er immer eine Reserve im Umfang der zu überschreibenden Tabelle hat.

Kurventabellen überschreiben

Kurventabellen, die nicht an einer Leitwertkopplung aktiv und mit CTABLOCK() gesperrt sind, können überschrieben werden.

Kurventabellen löschen

Kurventabellen, die nicht an einer Leitwertkopplung aktiv und mit CTABLOCK() gesperrt sind, können gelöscht werden.

10.2.4 Inbetriebnahme

10.2.4.1 Speicherkonfiguration

Für die Kurventabellen steht im statischen und dynamischen NC-Speicher ein definierter Speicherplatz zur Verfügung, der durch die folgenden Maschinendaten festgelegt wird:

Statischer NC-Speicher	
MD18400 \$MN_MM_NUM_CURVE_TABS	Legt die Anzahl der Kurventabellen fest, die im statischen NC-Speicher angelegt werden können.
MD18402 \$MN_MM_NUM_CURVE_SEGMENTS	Legt die Anzahl der Kurventabellen-Segmente fest, die im statischen NC-Speicher angelegt werden können.
MD18403 \$MN_MM_NUM_CURVE_SEG_LIN	Legt die maximale Anzahl der linearen Segmente im statischen NC-Speicher fest.
MD18404 \$MN_MM_NUM_CURVE_POLYNOMS	Legt die Anzahl der Kurventabellen-Polynome fest, die im statischen NC-Speicher angelegt werden können.

Dynamischer NC-Speicher	
MD18406 \$MN_MM_NUM_CURVE_TABS_DRAM	Legt die Anzahl der Kurventabellen fest, die im dynamischen NC-Speicher angelegt werden können.
MD18408 \$MN_MM_NUM_CURVE_SEGMENTS_DRAM	Legt die Anzahl der Kurventabellen-Segmente fest, die im dynamischen NC-Speicher angelegt werden können.
MD18409 \$MN_MM_NUM_CURVE_SEG_LIN_DRAM	Legt die maximale Anzahl der linearen Segmente im dynamischen NC-Speicher fest.
MD18410 \$MN_MM_NUM_CURVE_POLYNOMS_DRAM	Legt die Anzahl der Kurventabellen-Polynome fest, die im dynamischen NC-Speicher angelegt werden können.

Hinweis

Eine Kurventabelle mit linearen Segmenten kann nur dann effizienter im Speicher abgelegt werden, wenn:

MD18403 > 0 bzw. MD18409 > 0

Werden mit diesen Maschinendaten keine Speicherbereiche angelegt, werden lineare Segmente platzverschwendend als Polynom-Segmente gespeichert.

10.2.4.2 Werkzeugradiuskorrektur

MD20900

Bei der Werkzeugradiuskorrektur können Segmente entstehen, bei denen entweder die Folgeachse oder die Leitachse keine Bewegung hat. Eine fehlende Bewegung der Folgeachse stellt in der Regel kein Problem dar. Eine fehlende Bewegung der Leitachse hingegen erfordert eine Angabe, wie derartige Unstetigkeiten behandelt werden sollen, d. h. ob in diesen Fällen eine Kurventabelle erstellt werden soll oder nicht. Diese Angabe erfolgt mit der Einstellung im Maschinendatum:

MD20900 \$MC_CTAB_ENABLE_NO_LEADMOTION (Kurventabellen mit Sprung der Folgeachse)

Wert	Bedeutung
0	Es werden keine Kurventabellen erzeugt, die einen Sprung der Folgeachse enthalten. Es wird der Alarm 10949 ausgegeben und die Programmverarbeitung abgebrochen.
1	Kurventabellen mit einem Sprung der Folgeachse können erzeugt werden. Falls in einem Segment ein Sprung der Folgeachse auftritt, wird der Alarm 10955 ausgegeben, die Programmverarbeitung wird jedoch fortgesetzt.
2	Kurventabellen mit einem Sprung der Folgeachse können angelegt werden, ohne dass ein Alarm ausgegeben wird.

Hinweis

Bei einer Kurventabelle, die Segmente ohne Leitachsbewegung enthält (dies bedeutet, dass die Folgeachse an dieser Stelle einen Sprung macht), kann die Folgeachse den Sprung nur im Rahmen ihrer Dynamik-Grenzen (max. Geschwindigkeit und max. Beschleunigung) nachvollziehen. Dies bedeutet, dass in diesem Fall immer eine Abweichung von der programmierten Kurve auftritt.

10.2.4.3 Vorgabe des Speichertyps

MD20905

Wenn bei der Definition oder beim Löschen einer Kurventabelle keine Speicherangabe erfolgt, wird der Speichertyp durch folgendes Maschinendatum festgelegt:

MD20905 \$MC_CTAB_DEFAULT_MEMORY_TYPE (Default-Speichertyp für Kurventabellen)

Wert	Bedeutung
0	Standardmäßig werden Kurventabellen im statischen NC-Speicher angelegt.
1	Standardmäßig werden Kurventabellen im dynamischen NC-Speicher angelegt.

10.2.5 Programmierung

Definition

Folgende modal wirksame Sprachbefehle arbeiten mit Kurventabellen: (Erläuterungen zu den Parametern finden Sie am Ende der Liste der Funktionen.)

- Beginn der Definition einer Kurventabelle:
CTABDEF(Folgeachse, Leitachse, n, applim, memType)
- Ende der Definition einer Kurventabelle:
CTABEND()
- Löschen von Kurventabelle(n):
CTABDEL(n)
; die Kurventabelle n
CTABDEL(n, m)
; [n < m], mehrere im Nummernbereich
; Es wird im statischen "SRAM" und dynamischen "DRAM" NC-Speicher gelöscht.
- CTABDEL(n, m, memType)
; Löschen mit Speichertyp-Angabe:
Es werden im angegebenen Speichertyp die Kurventabellen mit den Nummern aus dem Bereich gelöscht, die im genannten Speichertyp liegen. Die übrigen bleiben bestehen.
Alle Tabellen in einem bestimmten Speichertyp löschen:
CTABDEL(, , "DRAM")
CTABDEL(, , "SRAM")
CTABDEL()
; alle im DRAM oder
; alle im SRAM:
; alle unabhängig vom Speichertyp
- Lesen des Folgewertes zu einem Leitwert
CTAB(Leitwert, n, grad, [Folgeachse, Leitachse])
- Lesen des Leitwertes zu einem Folgewert
CTABINV(Folgewert, approxLeitwert, n, grad, [Folgeachse, Leitachse])

Zugriff auf Kurventabellensegmente

- Lesen des Startwertes (Folgeachswert) eines Tabellensegmentes
CTABSSV(Leitwert, n, grad, [Folgeachse, Leitachse])
- Lesen des Endwertes (Folgeachswert) eines Tabellensegmentes
CTABSEV(Leitwert, n, grad, [Folgeachse, Leitachse])

Hinweis

Werden die Kurventabellenfunktionen, wie CTAB(), CTABINV(), CTABSSV() usw., in **Synchronaktionen** verwendet, so sind für den Rückgabewert und das Argument "grad" der Funktion nur **Hauptlaufvariable**, z. B. \$AC_PARAM[...] oder \$R[...] zulässig.

Beispiel:

```
ID=1 WHEN TRUE DO $R1 = CTABSSV(10, 1, $R2)
```

oder

```
ID=1 WHEN TRUE DO $AC_PARAM[1] = CTABSSV(10, 1, $AC_PARAM[2])
```

Sperrern setzen/aufheben

Die folgenden Funktionen gestatten es Teilprogrammen, Sperrern gegen Löschen und Überschreiben zu setzen oder aufzuheben.

- **Sperrern** gegen Löschen und Überschreiben **setzen**.
Allgemeine Form: CTABLOCK(n, m, memType)
- **Sperrern** gegen Löschen und Überschreiben **aufheben**.
CTABUNLOCK gibt die mit CTABLOCK gesperrten Tabellen wieder frei. Tabellen, die in einer aktiven Kopplung wirken, bleiben weiterhin gesperrt, d. h. sie können weiterhin nicht gelöscht werden. Die Sperre mit CTABLOCK ist allerdings aufgehoben, d. h. sobald die Sperrung durch die aktive Kopplung durch Deaktivierung der Kopplung aufgehoben wird, kann diese Tabelle gelöscht werden. Ein nochmaliger CTABUNLOCK Aufruf ist nicht notwendig.
Allgemeine Form: CTABUNLOCK(n, m, memType)

Anwendungen in den Formen:

Kurventabelle mit Nummer n

```
CTABLOCK(n)
```

Kurventabellen im Nummernbereich von n bis m

```
CTABLOCK(n, m)
```

Alle Kurventabellen ungeachtet des Speichertyps

```
CTABLOCK()
```

Alle Kurventabellen im angegebenen Speichertyp

```
CTABLOCK(, , memType)
```

Kurventabelle mit Nummer n

```
CTABUNLOCK(n)
```

Kurventabellen im Nummernbereich von n bis m

CTABUNLOCK(n, m)

Alle Kurventabellen ungeachtet des Speichertyps

CTABUNLOCK()

Alle Kurventabellen im angegebenen Speichertyp

CTABUNLOCK(, , memType)

Weitere Befehle zur Ermittlung und Unterscheidung von Kurventabellen für Anwendungen bei einer Diagnose und Optimierung der Ressourcennutzung:

- Anzahl der **definierten** Tabellen **ungeachtet** des Speichertyps
CTABNO()
- Anzahl der **definierten** Tabellen im statischen oder dynamischen NC-Speicher
CTABNOMEM(memType)
- Anzahl der **noch möglichen** Kurventabellen im Speicher memType.
CTABFNO(memType)
- Tabellen-Nummer der **n-ten** Kurventabelle.
Allgemeine Form: CTABID(n, memType)
Liefert die Tabellen-Nummer der n-ten Kurventabelle, im Speichertyp memType.
MitCTABID(1, memType) wird die höchste Kurvennummer (105) vom angegebenen Speichertyp auslesen.
CTABID(n)
Liefert die Tabellen-Nummer der n-ten Kurventabelle im Speicher der durch das folgende Maschinendatum angegeben wird:
MD20905 \$MC_CTAB_DEFAULT_MEMORY_TYPE (Default Speichertyp für Kurventabellen)
CTABID(p)
Liefert die ID (Tabellen-Nummer) der Kurventabelle, die im Speicher als p-te Kurventabelle eingetragen ist.

Hinweis

Wird z. B. zwischen aufeinander folgenden Aufrufen von CTABID() die Reihenfolge geändert, so kann mit CTABID(n, ...) eine **andere** Kurventabellen wie vor der Änderung im Speicher geliefert werden.

- Gibt **den Sperrzustand** der Kurventabelle mit der Nummer n zurück.
CTABISLOCK(n)
- **Prüft** die Kurventabelle mit der Nummer n.
CTABEXISTS(n)
- Gibt den **Speicher zurück**, in dem die Kurventabelle mit der Nummer n angelegt ist.
CTABMEMTYP(n)
- Gibt die **Tabellenperiodizität zurück**.
CTABPERIOD(n)
- Anzahl der bereits verwendeten **Kurvensegmente** im Speicher memType.
CTABSEG(memType, segType)
- Anzahl der von der **Kurventabelle** mit der Nummer n verwendeten Kurvensegmente
CTABSEGID(n, segType)

- Anzahl der **noch** möglichen **Kurvensegmente** im Speicher memType.
CTABFSEG(memType, segType)
- Anzahl der **maximal** möglichen **Kurvensegmente** im Speicher memType.
CTABMSEG(memType, segType)
- Anzahl der bereits **verwendeten Polynome** im Speicher memType.
CTABPOL(memType)
- Anzahl der von der Kurventabelle mit der Nummer n verwendeten **Kurvenpolynome**.
CTABPOLID(n)
- Anzahl der **noch** möglichen **Polynome** im Speicher memType.
CTABFPOL(n)
- Anzahl der **maximal** möglichen **Polynome** im Speicher memType.
CTABMPOL(n)

Randwerte von Kurventabellen

Verhalten der Leitachse bzw. Folgeachsen an den Rändern der Kurventabelle:

- Von einer Folgeachse wird der Wert am **Anfang** der Kurventabelle gelesen.
CTABTSV(n, grad, FAchse), Folgewert am Anfang der Kurventabelle
- Von einer Folgeachse wird der Wert am **Ende** der Kurventabelle gelesen.
CTABTEV(n, grad, FAchse), Folgewert am Ende der Kurventabelle
- Von der Leitachse wird der Wert am **Anfang** der Kurventabelle gelesen.
CTABTSP(n, grad, FAchse), Leitwert am Anfang der Kurventabelle
- Von der Leitachse wird der Wert am **Ende** der Kurventabelle gelesen.
CTABTEP(n, grad, FAchse), Leitwert am Ende der Kurventabelle
- Wertebereich des Folgewertes bestimmen.
CTABTMIN(n, FAchse), Minimaler Folgewert der Kurventabelle
CTABTMAX(n, FAchse), Maximaler Folgewert der Kurventabelle

Parameter

- Folgeachse:
Name der Achse, durch welche die Folgeachse in der Definition programmiert wird.
- Leitachse:
Name der Achse, durch welchen der Leitwert programmiert wird.
- n, m
Nummern der Kurventabellen.
Die Nummern der Kurventabellen können beliebig vergeben werden. Sie dienen ausschließlich zur eindeutigen Identifizierung einer Kurventabelle.
Für das Löschen eines Bereiches von Kurventabellen mit dem Befehl CTABDEL(n, m) muss m größer als n sein.
- p
Eintragsplatz (im Speicherbereich memType)

- applim:
Verhalten an den Rändern der Kurventabelle.
 - 0 nicht periodisch (Tabelle wird nur einmal abgearbeitet, auch bei Rundachsen.)
 - 1 periodisch, modulo (Der Modulo-Wert richtet sich nach den LA Tabellenwerten.)
 - 2 periodisch, modulo (LA- und FA- sind periodisch.)
- Leitwert
Positionswert, zu welchem ein Folgewert bestimmt werden soll.
- Folgewert
Positionswert, zu welchem ein Leitwert berechnet werden soll.
- aproxLeitwert
Positionswert, mit welchem im Falle der mehrdeutigen Umkehrfunktion der Kurventabelle ein eindeutiger Leitwert bestimmt werden kann.
- grad
Parameter, in welchem die Steigung der Tabellenfunktion zurückgegeben wird.
- memType
Optionaler Parameter zur Angabe des Speichertyps, in dem die Kurventabelle angelegt werden soll.
Mögliche Werte:
"SRAM" Kurventabelle wird im statischen NC-Speicher angelegt.
"DRAM" Kurventabelle wird im dynamischen NC-Speicher angelegt.
Wird ein ungültiger memType eingegeben, wird der Wert -2 zurückgegeben.
Wird der Parameter weggelassen, dann wird der durch das folgende Maschinendatum eingestellte Speichertyp wirksam:
MD20905 \$MC_CTAB_DEFAULT_MEMORY_TYPE (Default Speichertyp für Kurventabellen)
- segType
Optionaler Parameter zur Angabe der Segmentart
Mögliche Werte:
segType "L" lineare Segmente
segType "P" Polynomsegmente

Literatur:

Programmierhandbuch Arbeitsvorbereitung; Achskopplungen, Kapitel: Kurventabellen (CTAB)

Einschränkungen

Für die Programmierung gelten folgende Einschränkungen:

- Der NC-Satz darf keinen Vorlaufstopp erzeugen.
- In der Bewegung der Leitachse dürfen keine Sprünge auftreten.
- Jeder Satz, der eine Bewegungsanweisung für die Folgeachse enthält, muss auch eine Bewegung für die Leitachse haben.
- Die Bewegungsrichtung der Leitachse darf sich an keiner Stelle des Bewegungsgesetzes umkehren, d. h. die Position der Leitachse muss innerhalb des Bewegungsablaufs immer eindeutig sein.
Die programmierte Kontur darf sich nicht senkrecht zur Leitachse bewegen.

- Achsnamen aus Gantry-Achsverbänden können nicht zur Tabellendefinition benutzt werden (nur Leitachse geht).
- Abhängig vom folgenden Maschinendatum können Sprünge der Folgeachse bei fehlender Bewegung der Leitachse toleriert werden:
MD20900 \$MC_CTAB_ENABLE_NO_LEADMOTION (Kurventabellen mit Sprung der Folgeachse)
Die übrigen oben genannten Einschränkungen gelten weiterhin.

Achszuordnung

Erfolgt erst bei Aktivierung einer Kopplung mit der Kurventabelle.

Hinweis

Eine Prüfung dynamischer Grenzwerte der Bewegungsanweisungen für eine Kurventabelle erfolgt erst bei Aktivierung bzw. bei der Interpolation.

Startwert

Die erste Bewegungsanweisung innerhalb der Definition einer Kurventabelle legt den Startwert für den Leit- und Folgewert fest.

Es müssen alle Anweisungen, die zu einem Vorlaufstopp führen, ausgeblendet werden.

Beispiel 1

Ohne Werkzeugradiuskorrektur, ohne Speichertyp

Programmcode	Kommentar
N100 CTABDEF (AX2, AX1, 3,0)	; Beginn der Definition für die nicht- ; periodische Kurventabelle Nummer 3
N110 AX1=0 AX2=0	; 1. Bewegungsanweisung legt den Startwert fest ; Leitwert: 0, Folgewert:0
N110 AX1=20 AX2=0	; 1. Kurvensegment: Leitwert: 0...20, ; Folgewert: Startwert ...0
N120 AX1=100 AX2=6	; 2. Kurvensegment: Leitwert: 20...100, ; Folgewert: 0...6
N130 AX1=150 AX2=6	; 3. Kurvensegment: Leitwert: 100...150, ; Folgewert 6
N130 AX1=180 AX2=0	; 4. Kurvensegment: Leitwert: 150...180, ; Folgewert: 6...0
N200 CTABEND	; Ende der Definition, die Kurventabelle ; wird in ihrer internen Darstellung erzeugt. ; Der Vorlauf reorganisiert auf den Zustand ; zu Beginn von N100

Beispiel 2

Beispiel einer Kurventabelle mit aktiver Werkzeugradiuskorrektur:

Vor der Definition einer Kurventabelle mit CTABDEF() darf die Werkzeugradiuskorrektur nicht aktiv sein, sonst wird der Alarm 10942 ausgegeben. Das heißt, die Werkzeugradiuskorrektur **muss innerhalb** der Definition der Kurventabelle aktiviert werden. Ebenso muss diese vor dem Ende der Kurventabellendefinition mit CTABEND wieder deaktiviert werden.

Programmcode	Kommentar
N10 CTABDEF(Y, X, 1, 0)	; Beginn der Definition für die nicht- ; periodische Kurventabelle Nummer 1
N20 X0 Y0	
N30 G41 X10 Y0	; WZR-Korrektur ein
N40 X20 Y20	
N50 X40 Y0	
N60 X60 Y20	
N70 X80 Y0	
N80 G40 X90 Y0	; WZR-Korrektur aus
N90 CTABEND	

Im Satz N30 wird die Werkzeugradiuskorrektur aktiviert, dadurch wird in diesem Satz die Anfahrbewegung der Radiuskorrektur durchgeführt. Ebenso erfolgt im Satz N80 bei der Deaktivierung der Radiuskorrektur die Abfahrbewegung.

Hinweis

Zwischen CTABDEF und CTABEND müssen Wertepaare für genau die Achsnamen angegeben werden, die in CTABDEF als Leitachsname und Folgeachsname angegeben wurden. Im Fehlerfall können Alarme oder unerwünschte Konturen erzeugt werden.

10.2.6 Zugriff auf Tabellenpositionen und Tabellensegmente

Lesen von Tabellenpositionen

Mit den Programmbefehlen CTAB und CTABINV kann vom Teileprogramm und von Synchronaktionen aus direkt der Folgewert zu einem Leitwert (CTAB) oder die Umkehrung der Kurventabelle, das heißt der Leitwert zu einem Folgewert gelesen werden. Mit Hilfe des Steigungswertes kann der Anwender die Geschwindigkeit der Folge- oder der Leitachse an der gewünschten Tabellenposition berechnen.

Lesen von Segmentpositionen

Die Segmentpositionen einer Kurventabelle bezüglich des Folgeachswertes können mit den Aufrufen CTABSSV und CTABSEV gelesen werden.

Die Sprachbefehle CTABSSV und CTABSEV liefern generell die Start- bzw. Endwerte der internen Segmente der Kurventabelle für die Folgeachse. Diese Werte stimmen nur dann mit den programmierten Werte der Kurventabelle überein, wenn sich die programmierten

Segmente 1:1 in die internen Segmente der Kurventabelle umwandeln lassen. Dies ist auf jeden Fall gegeben, falls bei der Definition der Kurventabellen nur G1-Sätze bzw. Achspolynome verwendet werden, ohne dass weitere Funktionen aktiv sind.

Die programmierten Segmente werden u. U. **nicht** unverändert in die internen Segmente der Kurventabelle umgewandelt, wenn:

1. Kreise oder Evolventen programmiert sind
2. Fasen bzw. Runden ist aktiv (CHF, RND)
3. Überschleifen mit G643 aktiv ist
4. Kompressor aktiv ist (COMPON, COMPCURV, COMPCAD)
5. Werkzeugradiuskorrektur bei Polynominterpolation aktiv ist.

In diesen Fällen können die Sprachbefehle CTABSSV und CTABSEV nicht dazu verwendet werden, die Start- bzw. Endpunkte der programmierten Segmente abzufragen.

CTABINV

Bei der Verwendung der Umkehrfunktion der Kurventabellen CTABINV muss beachtet werden, dass die Abbildung vom Folgewert auf den Leitwert ggf. nicht eindeutig ist.

Innerhalb einer Kurventabelle kann der Folgewert für (beliebig) viele Leitwertpositionen den gleichen Wert haben. Um diese Mehrdeutigkeit auflösen zu können benötigt der Programmbefehl CTABINV neben dem Folgewert einen weiteren Parameter, mit welchem der "richtige" der möglichen Leitwerte selektiert werden kann. CTABINV gibt als Ergebnis immer den Leitwert zurück, der diesem Hilfsparameter am nächsten liegt. Dieser Hilfswert kann zum Beispiel der Leitwert aus dem vorhergehenden Interpolatortakt sein.

Hinweis

Obwohl mit dem Hilfsparameter die Umkehrfunktion der Kurventabelle ein eindeutiges Ergebnis berechnen kann, muss darauf hingewiesen werden, dass Konturen erzeugt werden können, bei denen die Umkehrfunktion auf Grund von numerischen Ungenauigkeiten nicht das Ergebnis liefert, welches sich bei einer Rechnung mit unbegrenzter Genauigkeit ergeben würde.

Optionale Parameter

Die Funktionen CTAB, CTABINV, CTABSSV und CTABSEV haben optionale Parameter Leit- oder Folgeachse. Wird einer dieser Parameter programmiert, dann werden der Leitwert und der Folgewert mit den Skalierungsfaktoren der betreffenden Achsen modifiziert.

Dies ist insbesondere wichtig, wenn Achsen in unterschiedlichen Längeneinheiten (inch/metrisch) projiziert sind. Werden die optionalen Parameter nicht programmiert, dann werden der Leit- und der Folgewert bei der Wandlung von der externen zur internen Darstellung wie Bahnpositionen behandelt. Das heißt, die Werte werden entsprechend der projizierten Auflösung (Nachkommastellen) multipliziert und die verbliebenen Nachkommastellen werden abgeschnitten.

Bestimmung des zum Leitwert X gehörigen Segments

Beispiel zum Lesen der Segmentanfangs- und Segmentendwerte zur Bestimmung des zu dem Leitwert $X = 30$ gehörigen Kurvensegments unter Verwendung von CTABSSV und CTABSEV:

Programmcode	Kommentar
N10 DEF REAL STARTPOS	; Beginn der Definition für die Start- und Anfangsposition der Kurventabelle
N20 DEF REAL ENDPOS	
N30 DEF REAL GRADIENT	
N100 CTABDEF(Y,X,1,0)	; Beginn der Tabellendefinition
N110 X0 Y0	; Startposition, 1.Tabellensegment
N120 X20 Y10	; Endposition, 1. Tabellensegment = ; Startposition 2. Tabellensegment
N130 X40 Y40	
N140 X60 Y10	; 3. Kurvensegment
N150 X80 Y0	; 4. Kurvensegment
N160 CTABEND	; Ende der Tabellendefinition
N200 STARTPOS=CTABSSV(30.0,1,GRADIENT)	; Startposition Y in Segment 2 = 10.
N210 ENDPOS=	; Endposition Y in Segment 2 = 40
CTABSEV(30.0,1,GRADIENT)	; Zu LW X = 30.0 gehört Segment 2

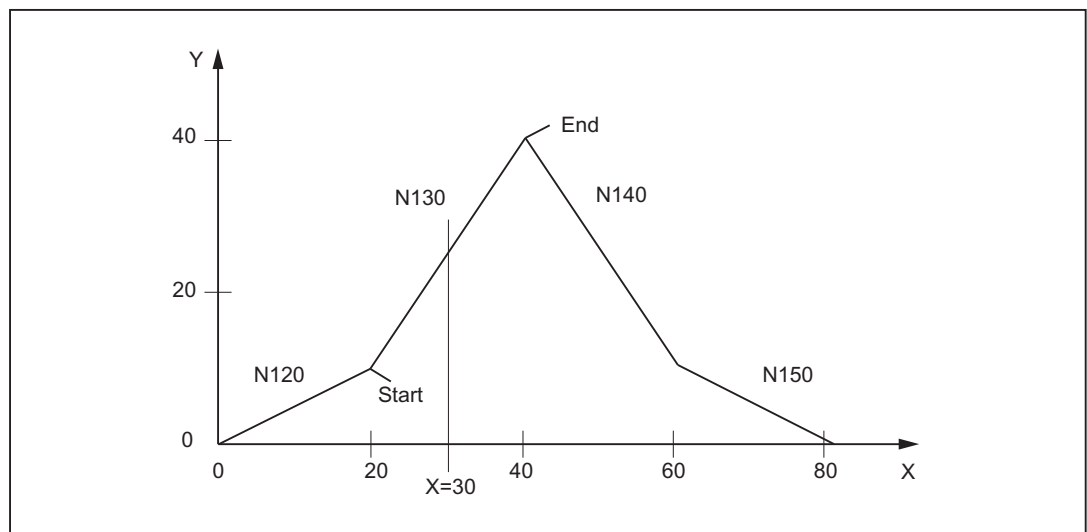


Bild 10-2 Bestimmung des zu dem Leitwert $X = 30$ gehörenden Kurvensegments

Werte am Anfang und Ende lesen

Die Werte der Folgeachsen und auch der Leitachse am Anfang und am Ende einer Kurventabelle können gelesen werden mit den Aufrufen:

$R10 = \text{CTABTSV}(n. \text{ grad}, \text{F Achse}, \text{Folgewert am Anfang der Tabelle})$

$R10 = \text{CTABTEV}(n. \text{ grad}, \text{F Achse}), \text{Folgewert am Ende der Tabelle}$

$R10 = \text{CTABTSP}(n. \text{ grad}, \text{L Achse}, \text{Leitwert am Anfang der Tabelle})$

R10 =CTABTEP(n, grad, LAchse, Leitwert am Ende der Tabelle

Wertebereich des Folgewerts

Die Bestimmung des minimalen und maximalen Werts der Tabelle unter Verwendung von CTABTMIN und CTABTMAX zeigt folgendes Beispiel:

Programmcode	Kommentar
N10 DEF REAL STARTVAL	; Beginn der Definition für die Start- und- Anfangswerte der Kurventabelle
N20 DEF REAL ENDVAL	
N30 DEF REAL STARTPARA	
N40 DEF REAL ENDPARA	
N50 DEF REAL U_MINVAL	
N60 DEF REAL U_MAXVAL	
N70 DEF REAL GRADIENT	
...	
N100 CTABDEF(Y,X,1,0)	; Beginn der Tabellendefinition
N110 X0 Y10	; Startwert, 1. Tabellensegment
N120 X30 Y40	; Endposition, 1. Tabellensegment = ; Startposition 2. Tabellensegment
N130 X60 Y5	; Endposition, 2. Tabellensegment ...
N140 X70 Y30	; Endposition, 3. Kurvensegment ...
N150 X80 Y20	; Endposition, 4. Kurvensegment ...
N160 CTABEND	; Ende der Tabellendefinition
...	
N200 STARTPOS=CTABTSV(1,GRADIENT)	; STARTPOS = 10 ; Startposition der Tabelle sowie
N210 ENDPOS=CTABTEV(1,GRADIENT)	; ENDPOS = 20 ; Endposition der Tabelle
N220 STARTPARA=CTABTSP(1,GRADIENT)	; STARTPARA = 0 ; Leitwert am Anfang der Kurventabelle
N230 ENDPARA=CTABTEP(1,GRADIENT)	; ENDPARA = 80 ; Leitwert am Ende der Kurventabelle vom Wer- tebereich der Folgeachse lesen
N240 U_MINVAL=CTABTMIN(1)	; Minimalwert bei Y = 5 und
N250 U_MAXVAL=CTABTMAX(1)	; Maximalwert bei Y = 40

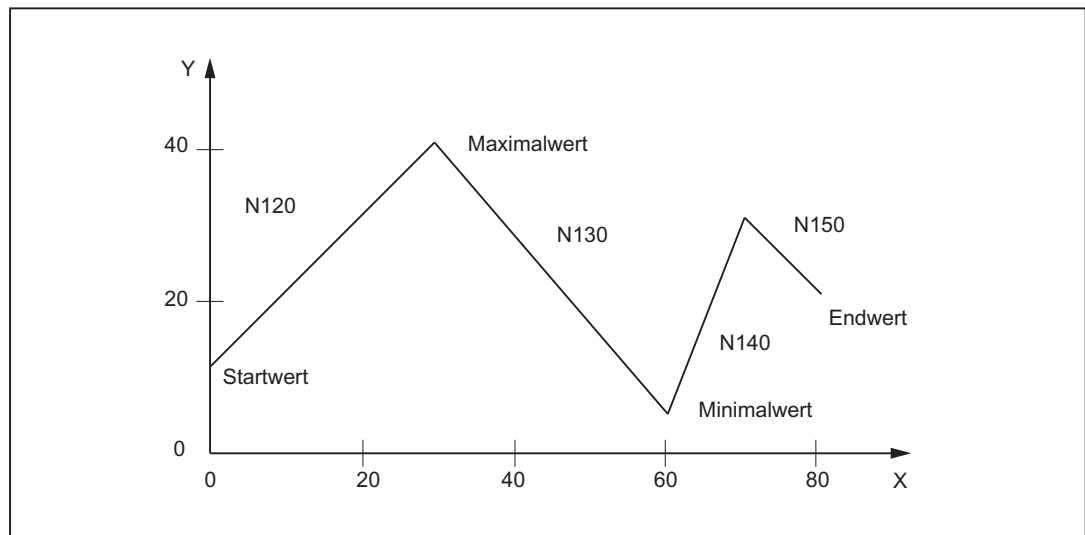


Bild 10-3 Bestimmung des minimalen und maximalen Werts der Tabelle

10.2.7 Aktivierung/Deaktivierung

Aktivierung

Das Ankoppeln realer Achsen an eine Kurventabelle wird aktiviert mit dem Befehl:

```
LEADON (<Folgeachse>, <Leitachse>, <n>)
```

mit <n> = Nummer der Kurventabelle

Die Aktivierung ist möglich:

- im Teileprogramm
- bei der Definition einer Synchronaktion

Beispiel:

```
...
N1000 LEADON(A,X,3)      ; die Achse A folgt dem Leitwert X nach den in der Kur-
                          ; ventabelle Nummer 3 definierten Bewegungsgesetzen
...
```

Deaktivierung

Das Ausschalten der Kopplung an eine Kurventabelle erfolgt über den Befehl:

```
LEADOF (<Folgeachse>, <Leitachse>)
```

Die Deaktivierung ist möglich:

- im Teileprogramm
- in Synchronaktionen

Hinweis

Bei der Programmierung von LEADOF ist auch die verkürzte Form ohne Angabe der Leitachse möglich.

Beispiel:

```
...  
N1010 LEADOF(A,X)          ; die Kopplung der Achse A an ihren Leitwert wird aufge-  
                           hoben  
...
```

Mehrfachnutzung

Eine Kurventabelle kann mehrfach in einem Teileprogramm und für die Kopplung unterschiedlicher Kanalachsen verwendet werden.

10.2.8 Sonderfall Modulo-Leitachse

Position wirkt absolut

Bei einer axialen Leitwertkopplung ist die Position der Folgeachse über eine Kurventabelle eindeutig, d. h. absolut der Leitachse zugeordnet.

Dies bedeutet, dass auch im Falle einer Modulo-Rundachse als Leitachse die Position der Leitachse absolut eingeht. D. h. die in die Kurventabelle eingehende Position der Modulo-Rundachse wird nicht modulo-reduziert, sie wirkt absolut.

Beispiel

Die Position der Modulo-Rundachse bei LEADON sei 210°. Die Position 210° dient als Eingangswert in die Kurventabelle. Nach einer Umdrehung der Modulo-Rundachse wird als Achsposition wieder 210° angezeigt. Als Eingangswert in die Kurventabelle wird aber die Absolutposition 570° genommen:

$$210^\circ + 1 \text{ Rund } (360^\circ) = 570^\circ$$

10.2.9 Verhalten in den Betriebsarten AUTOMATIK, MDA, JOG

Wirksamkeit

Eine eingeschaltete Kurventabelle ist in den Betriebsarten AUTOMATIK, MDA und JOG aktiv.

Grundstellung nach Hochlauf

Nach Hochlauf sind keine Kurventabellen aktiv.

10.2.10 Wirksamkeit der PLC-Nahtstellensignale

Abhängige Folgeachse

Für die von einer Leitachse abhängige Bewegung einer Folgeachse sind nur die Nahtstellensignale der Folgeachse wirksam, die zu einem Stopp der Bewegung führen (z. B. achsspezifischer Vorschub Halt, Achsensperre, Reglerfreigabe, usw.).

Leitachse

Bei einem aktivierten Achsverband wirken die Nahtstellensignale der Leitachse durch die Achskopplung auf die zugehörige Folgeachse, d. h.:

- eine Vorschubbeeinflussung der Leitachse bewirkt eine entsprechende Vorschubbeeinflussung der Folgeachse.
- ein Stillsetzen der Leitachse durch NST (z. B. achsspezifischer Vorschub Halt, Achsensperre, Reglerfreigabe, usw.) bewirkt ein Stillsetzen der zugehörigen Folgeachse.

Die Wirkung der Achsensperre der Leitachse auf die Folgeachse kann durch folgende MD-Einstellung verhindert werden:

MD37160 \$MA_LEAD_FUNCTION_MASK, Bit 1 = 1

Lagemesssystem 1/2 (DB31, ... DBX1.5/1.6)

Für einen aktivierten Achsverband ist eine Umschaltung des Lagemesssystems für Leit- und Folgeachsen nicht verriegelt. Die Kopplung bleibt dabei erhalten.

Empfehlung: Die Umschaltung bei ausgeschalteter Kopplung vornehmen.

10.2.11 Diagnose und Optimierung der Ressourcennutzung

Die folgenden Funktionen gestatten es **Teileprogrammen** sich aktuell über die Belegung der Ressourcen für Kurventabellen, Tabellensegmente und Polynome zu informieren.

Als Folge der Ergebnisse aus den Diagnosefunktionen können die noch verfügbaren Ressourcen **dynamisch** mit den Funktionen unter flexibel benutzt werden ggf. ohne den

absoluten Speicherbedarf zu vergrößern. Die Erklärung der Parameter im Kap. "Programmierung Kurventabelle" gilt auch für die folgenden Funktionen.

a) Kurventabellen

- Gesamtanzahl der definierten Tabellen feststellen.
Die Bestimmung erfolgt über alle Speichertypen (siehe auch CTABNOMEM)
CTABNO ()
- Anzahl der definierten Tabellen im statischen oder dynamischen NC-Speicher feststellen.
CTABNOMEM (memType)
Ist memType nicht angegeben, gilt der im folgenden Maschinendatum gesetzte Speichertyp:
MD20905 \$MC_CTAB_DEFAULT_MEMORY_TYPE (Default Speichertyp für Kurventabellen)
Ergebnis:
>= 0: Anzahl der definierten Tabellen
-2: Speichertyp ungültig
- Anzahl der noch möglichen Kurventabellen im Speichertyp feststellen.
CTABFNO (memType)
Ist memType nicht angegeben, gilt der im folgenden Maschinendatum gesetzte Speichertyp:
MD220905 \$MC_CTAB_DEFAULT_MEMORY_TYPE
Ergebnis:
>= 0: Anzahl der noch möglichen Tabellen
-2: Speichertyp ungültig
- Tabellen-Nummer der p-ten Tabelle im optional angegebenen Speichertyp feststellen
CTABID (p, memType)
Ist memType nicht angegeben, gilt der im folgenden Maschinendatum gesetzte Speichertyp:
MD20905 \$MC_CTAB_DEFAULT_MEMORY_TYPE
Ergebnis:
Tabellenummer oder
Alarm bei ungültigem p oder memType

Bei der Verwendung der Funktion CTABID (p, memType) sollten keine Annahmen über die Reihenfolge im Speicher der Kurventabellen getroffen werden. Die Funktion CTABID (p, . . .) liefert die ID (Tabellen-Nummer) der Kurventabelle, die im Speicher als p-te Kurventabelle eingetragen ist.

Wird zwischen aufeinander folgenden Aufrufen von CTABID () die Reihenfolge der Kurventabellen im Speicher geändert, z. B. durch Löschen von Kurventabellen mit CTABDEL (), kann die Funktion CTABID (p, . . .) mit derselben Nummer eine andere Kurventabelle liefern als vorher.

Soll dies verhindert werden, können die betroffenen Kurventabellen gesperrt werden durch Verwendung des Sprachbefehls `CTABLOCK (. . .)`. In diesem Fall ist darauf zu achten, dass die betroffenen Kurventabellen mit `CTABUNLOCK ()` wieder freigegeben werden.

- **Sperrzustand abfragen**
Tabelle n
`CTABISLOCK (n)`
Ergebnis:
> 0: Tabelle ist gesperrt
Sperrgrund:
1: durch `CTABLOCK()`
2: durch aktive Kopplung
3: durch `CTABLOCK ()` und aktive Kopplung
= 0: Tabelle ist nicht gesperrt
- 1: Tabelle existiert nicht
- **Prüfen, ob Kurventabelle existiert**
`CTABEXISTS (n)`
Ergebnis:
1: Tabelle existiert
0: Tabelle existiert nicht
- **Speichertyp einer Kurventabelle feststellen**
`CTABMEMTYP (n)`
Ergebnis:
0: Tabelle im statischen NC-Speicher "SRAM"
1: Tabelle im dynamischen NC-Speicher "DRAM"
-1: Tabelle existiert nicht
- **Feststellen, ob Tabelle als periodisch definiert ist**
`CTABPERIOD (n)`
Ergebnis:
0: Tabelle ist nicht periodisch
1: Tabelle ist periodisch in der Leitachse
2: Tabelle ist periodisch in der Leit- und Folgeachse
-1: Tabelle existiert nicht

b) Kurventabellensegmente

- Anzahl der **verwendeten** Kurvensegmente vom Typ `segType` im Speicherbereich `memType` feststellen.
- `CTABSEG (memType, segType)`
- Ist `memType` nicht angegeben, gilt der im folgenden Maschinendatum gesetzte Speichertyp:
`MD20905 $MC_CTAB_DEFAULT_MEMORY_TYPE`
Ergebnis:
>= 0: Anzahl der Kurvensegmente
-2: Speichertyp ungültig
Ist `segType` nicht angegeben, dann wird die Summe über Linear- und Polynom-Segmente im Speichertyp ausgegeben.
-2: `segType` ungleich "L" oder "P"

- Anzahl der verwendeten Kurvensegmente vom Typ segType einer bestimmte Kurventabelle feststellen
CTABSEGID(n, segType)
Ergebnis:
>= 0: Anzahl der Kurvensegmente
-1: Kurventabelle mit der Nummer n existiert nicht
-2: segType ungleich "L" oder "P"
- Anzahl der **freien** Kurvensegmente vom Typ segType des Speichertyps feststellen
CTABFSEG(memType, segType)
Ist memType nicht angegeben, gilt der im folgenden Maschinendatum gesetzte Speichertyp:
MD20905 \$MC_CTAB_DEFAULT_MEMORY_TYPE
Ergebnis:
>= 0: Anzahl der noch freien Kurvensegmente
-2: Speichertyp ungültig, segType ungleich "L" oder "P"
- Anzahl der **maximal** möglichen Kurvensegmente vom Typ segType des Speichertyps feststellen
CTABMSEG(memType, segType)
Ist memType nicht angegeben, gilt der im folgenden Maschinendatum gesetzte Speichertyp:
MD20905 \$MC_CTAB_DEFAULT_MEMORY_TYPE
Ergebnis:
>= 0: Anzahl der maximal möglichen Kurvensegmente
-2: Speichertyp ungültig, segType ungleich "L" oder "P"

c) Polynome

- Anzahl der **verwendeten** Polynome des Speichertyps feststellen
CTABPOL(memType)
Ist memType nicht angegeben, gilt der im folgenden Maschinendatum gesetzte Speichertyp:
MD20905 \$MC_CTAB_DEFAULT_MEMORY_TYPE
Ergebnis:
>= 0: Anzahl der verwendeten Polynome im Speichertyp
-2: Speichertyp ungültig
- Anzahl der von einer Kurventabelle verwendeten Kurvenpolynome feststellen
CTABPOLID(n)
Ergebnis:
>=0: Anzahl der verwendeten Kurvenpolynome
-1: Kurventabelle mit Nummer n existiert nicht

- Anzahl der noch **freien** Polynome des Speichertyps feststellen
CTABFPOL (memType)
Ist memType nicht angegeben, gilt der im folgenden Maschinendatum gesetzte Speichertyp:
MD20905 \$MC_CTAB_DEFAULT_MEMORY_TYPE
Ergebnis:
>= 0: Anzahl der noch freien Kurvenpolynome
-2: Speichertyp ungültig
- Anzahl der **maximal** möglichen Polynome des Speichertyps feststellen
CTABMPOL (memType)
Ist memType nicht angegeben, gilt der im folgenden Maschinendatum gesetzte Speichertyp:
MD20905 \$MC_CTAB_DEFAULT_MEMORY_TYPE
Ergebnis:
>= 0: Anzahl der maximal möglichen Kurvenpolynome
-2: Speichertyp ungültig

10.2.12 Randbedingungen

Transformationen

Transformationen sind in Kurventabellen nicht zulässig. Ausgenommen ist TRAANG.

TRAANG

Wenn TRAANG programmiert ist, wird das im Basiskoordinatensystem programmierte Bewegungsgesetz in das zugehörige Maschinenkoordinatensystem transformiert. Auf diese Weise kann eine Kurventabelle für eine Maschine mit schrägen Linearachsen kartesisch programmiert werden.

Die Bedingung "die Bewegungsrichtung der Leitachse darf sich an keiner Stelle des Bewegungsgesetzes umkehren" muss dann im Maschinenkoordinatensystem erfüllt sein. Es ist zu beachten, dass diese Bedingung im Basiskoordinatensystem nicht das gleiche bedeutet wie im Maschinenkoordinatensystem, da die Konturtangenten durch die Transformation verändert werden.

10.2.13 Beispiele

Definition einer Kurventabelle mit Linearsätzen

```
%_N_TAB_1_NOTPERI_MPF
; $PATH=/_N_WKS_DIR/_N_KURVENTABELLEN_WPD
; Def.TAB1 0-100mm Kue1/1 notperio.
```

```

%_N_TAB_1_NOTPERI_MPF
N10 CTABDEF(YGEO,XGEO,1,0) ; FA=Y LA=X Kurvennr.=1 Nicht
period.
N1000 XGEO=0 YGEO=0 ; Startwerte
N1010 XGEO=100 YGEO=100
CTABEND
M30

```

Definition einer Kurventabelle mit Polynomsätzen

```

%_N_TAB_1_NOTPERI_MPF
; $PATH=/_N_WKS_DIR/_N_KURVENTABELLEN_WPD
; Def.TAB1 0-100mm Kuel/1 notperio.
N10 CTABDEF(Y,X,1,0) ; FA=Y LA=X Kurven-
Nr.=1 Nicht period.
N16 G1 X0.000 Y0.000
N17 POLY PO[X]=(31.734,0.352,-0.412) PO[Y]=(3.200,2.383,0.401)
N18 PO[X]=(49.711,-0.297,0.169) PO[Y]=(7.457,1.202,-0.643)
N19 PO[X]=(105.941,1.961,-0.938) PO[Y]=(11.708,-6.820,-1.718)
N20 PO[X]=(132.644,-0.196,-0.053) PO[Y]=(6.815,-2.743,0.724)
N21 PO[X]=(147.754,-0.116,0.103) PO[Y]=(3.359,-0.188,0.277)
N22 PO[X]=(174.441,0.578,-0.206) PO[Y]=(0.123,1.925,0.188)
N23 PO[X]=(185.598,-0.007,0.005) PO[Y]=(-0.123,0.430,-0.287)
N24 PO[X]=(212.285,0.040,-0.206) PO[Y]=(-3.362,-2.491,0.190)
N25 PO[X]=(227.395,-0.193,0.103) PO[Y]=(-6.818,-0.641,0.276)
N26 PO[X]=(254.098,0.355,-0.053) PO[Y]=(-11.710,0.573,0.723)
N26 PO[X]=(254.098,0.355,-0.053) PO[Y]=(-11.710,0.573,0.723)
N27 PO[X]=(310.324,0.852,-0.937) PO[Y]=(-7.454,11.975,-1.720)
N28 PO[X]=(328.299,-0.209,0.169) PO[Y]=(-3.197,0.726,-0.643)
N29 PO[X]=(360.031,0.885,-0.413) PO[Y]=(0.000,-3.588,0.403)
CTABEND
N30 M30

```

Definition einer periodischen Kurventabelle

Tabellennummer: 2

Leitwertbereich: 0 - 360

Die Folgeachse führt von N70 bis N90 eine Bewegung von 0 nach 45 und wieder zurück nach 0 aus.

```

N10 DEF REAL DEPPOS
N20 DEF REAL GRADIENT
N30 CTABDEF(Y,X,2,1)
N40 G1 X=0 Y=0
N50 POLY

```

```

N10 DEF REAL DEPPPOS
N60 PO[X]=(45.0)
N70 PO[X]=(90.0) PO[Y]=(45.0,135.0,-90)
N80 PO[X]=(270.0)
N90 PO[X]=(315.0) PO[Y]=(0.0,-135.0,90)
N100 PO[X]=(360.0)
N110 CTABEND
N130 G1 F1000 X0 ; Test der Kurve durch eine Kopplung
                  ; von Y an X
N140 LEADON(Y,X,2)
N150 X360
N160 X0
N170 LEADOF(Y,X)
N180 DEPPPOS = CTAB(75.0,2,GRADIENT) ; Lesen der Tabellenposition beim
                  ; Leitwert 75.0 aus der Kurventabelle
                  ; mit der Tabellennummer 2
N190 G0 X75 Y=DEPPPOS ; Positionieren von Leit- und Folge-
                  ; achse
N200 LEADON(Y,X,2) ; nach dem Einschalten der Kopplung
                  ; ist kein Synchronisieren der Folge-
                  ; achse nötig
N210 G1 X110 F1000
N220 LEADOF(Y,X)
N230 M30

```

10.3 Leitwertkopplung - nur 840D sl

10.3.1 Kurzbeschreibung

10.3.1.1 Funktion

Mit Hilfe der Funktion "Axiale Leitwertkopplung" ist die zyklische Bearbeitung von kurzen Programmen mit enger Kopplung der Achsen untereinander und an einen Leitwert, der intern erzeugt oder auch extern vorgegeben werden kann, möglich.

Der Leitwert kann sich z. B. von einem Transportband oder einer Königswelle ableiten.

Die axiale Leitwertkopplung kann im NC-Teileprogramm oder per Synchronaktion ein- und ausgeschaltet werden.

10.3.1.2 Voraussetzungen

Für die Nutzung der Funktion wird die Option "Leitwertkopplung und Kurventabellen-Interpolation" oder die entsprechende optionale Ausführung der Generischen Kopplung (siehe Thema "Voraussetzungen (Seite 484)" in der "Kurzbeschreibung" zur Generischen Kopplung) benötigt.

10.3.2 Allgemeine Funktionalität

Kurventabelle

Bei der axialen Leitwertkopplung werden eine Leit- und eine Folgeachse synchron verfahren. Dabei ist die jeweilige Position der Folgeachse über eine Kurventabelle bzw. ein daraus berechnetes Polynom eindeutig einer - ggf. simulierten - Position der Leitachse zugeordnet.

Leitwertobjekt

Das Leitwertobjekt ist die Eingangsgröße für die Kurventabelle.

Als Position des Leitwertobjekts kann definiert sein:

- die Achs-Istposition (Istwert vom Geber gemessen)
oder
- die Sollposition (vom Interpolator berechnet) (Standardeinstellung)

Wenn die Leitachse von der gleichen NCU interpoliert wird, liefert die Sollwertkopplung ein besseres Folgeverhalten als bei Istwertkopplung möglich (im selben IPO-Takt).

Virtuelle Leitachse / Simulierter Leitwert

Wenn die Leitachse nicht von der gleichen NCU interpoliert wird, kann der in der NCU für diese Leitachse existierende Interpolator zur Leitwertsimulation benutzt werden. Dazu sind folgende Maschinendaten-Einstellungen vorzunehmen:

MD30132 \$MA_IS_VIRTUAL_AX[n] = 1 (Achse ist virtuelle Achse)

MD30130 \$MA_CTRLOUT_TYPE[n] = 0 (Simulation als Ausgabeart des Sollwerts)

Eigenschaften der Leitwertsimulation:

- Trennung von IPO und Servo.
- Istwerte der Achse werden erfasst.
- Sollwerte werden von IPO erzeugt aber nicht an den Servo übergeben.
- Beim Umschalten auf simulierten Leitwert kann die Simulation abhängig vom zuletzt gelesenen Istwert programmiert werden, während der Verlauf des Istwerts in der Regel außerhalb der Kontrolle der NCU liegt.
- Wird zwecks Leitwertsimulation das Leitwertobjekt von Istwertkopplung auf Sollwertkopplung umgeschaltet und wird im gleichen Interpolatorakt ein Fahrbefehl für die Leitachse abgesetzt, so wird vom NC der Interpolator für die Achse so initialisiert, dass ein in der ersten Ableitung stetiger Verlauf des Leitwerts resultiert.

Hinweis

Virtuelle Achsen, die einem realen Antrieb zugeordnet werden, müssen unverriegelt bleiben.

Verschiebung und Skalierung

Der Sollwert für die Folgeachse kann verschoben und skaliert werden. Dazu dienen folgende Settingdaten:

SD43102 `$$SA_LEAD_OFFSET_IN_POS` (Verschiebung des Leitwerts bei Kopplung zu dieser Achse)

SD43104 `$$SA_LEAD_SCALE_IN_POS` (Skalierung des Leitwerts bei Kopplung zu dieser Achse)

SD43106 `$$SA_LEAD_OFFSET_OUT_POS` (Verschiebung des Funktionswerts der Kurventabelle)

SD43108 `$$SA_LEAD_SCALE_OUT_POS` (Skalierung des Funktionswerts der Kurventabelle)

Wenn (x) eine periodische Kurventabelle ist und man diese als Schwingung betrachtet, können die Verschiebung und Skalierung auch so interpretiert werden:

SD43102 `$$SA_LEAD_OFFSET_IN_POS[Y]` verschiebt die Phase der Schwingung.

SD43106 `$$SA_LEAD_SCALE_OUT_POS[Y]` wirkt auf die Amplitude.

SD43108 `$$SA_LEAD_OFFSET_OUT_POS[Y]` verschiebt den Mittelpunkt der Schwingung

Wenn die Kopplung eingeschaltet und synchron ist, bewirkt Schreiben auf diese Settingdaten sofortiges Herausfahren der neuen Sollposition.

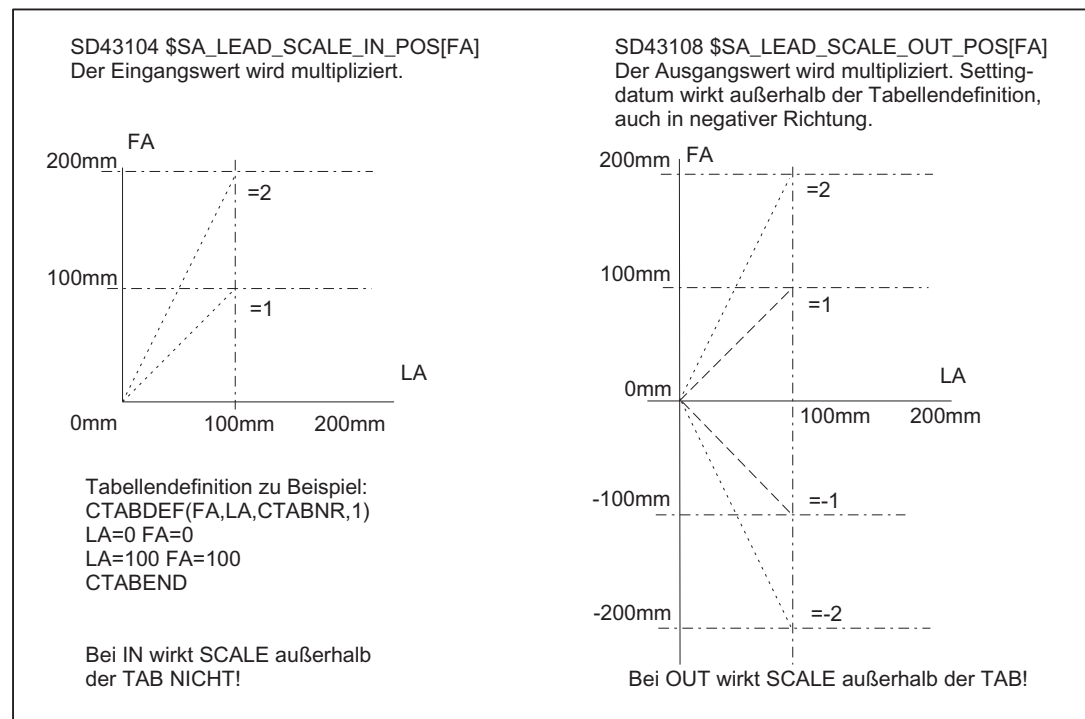


Bild 10-4 Leitwertkopplung Verschieben und Skalieren (multipliziert)

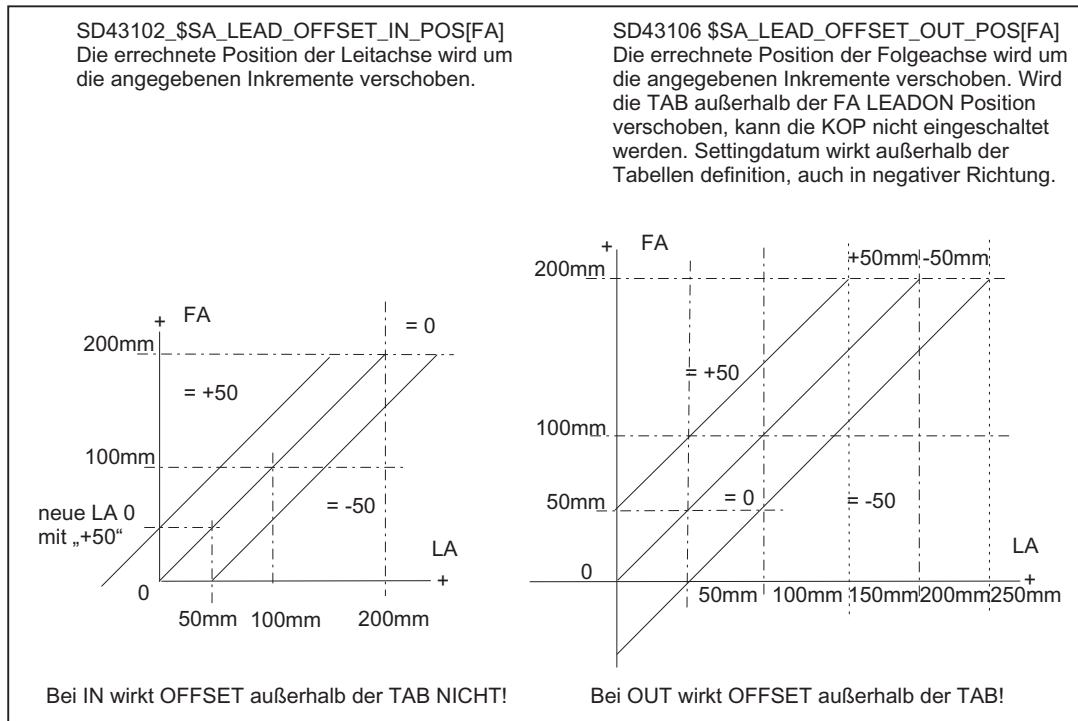


Bild 10-5 Leitwertkopplung Verschieben und Skalieren (mit Inkremente verschieben)

Reaktion auf Stopp

Alle Leitwert-gekoppelten Folgeachsen reagieren auf Kanal-Stopp und BAG-Stopp.

Auf Stopp wegen Programmende (M30, M02) reagieren Leitwert-gekoppelte Folgeachsen, die nicht durch statische Synchronaktion (IDS= . . .) eingeschaltet worden sind. In diesem Zusammenhang sind folgende Maschinendaten zu beachten:

MD20110 \$MC_RESET_MODE_MASK (Festlegung der Steuerungs-Grundstellung nach Reset/TP-Ende)

MD20112 \$MC_START_MODE_MASK (Festlegung der Steuerungs-Grundstellung bei NC-START)

Leitachse und Folgeachse müssen immer im selben Kanal interpolieren. Eine Folgeachse, die sich im anderen Kanal befindet, kann nicht gekoppelt werden (Achstausch).

Mit START und mit Wechsel der Betriebsart wird eine gestoppte Folgeachse der Leitwertkopplung wieder freigegeben.

Mit RESET wird eine gestoppte Folgeachse der Leitwertkopplung ebenfalls wieder freigegeben. Wenn diese Freigabe unerwünscht oder gefährlich ist (z. B. weil die Folgeachse an einen externen Leitwert gekoppelt ist, der nicht von der NC kontrolliert wird), dann soll MD20110 so programmiert werden, dass Leitwertkopplungen bei RESET ausgeschaltet werden (=2001H, d. h. Bit 13 auf 1 setzen).

Axiale Funktionen

Bei Istwertkopplung entsteht ein Positionsversatz zwischen der Leit- und Folgeachse. Ursache hierfür ist die IPO-Takt-bedingte Totzeit im Lageregler, die zwischen dem Istwert der Leitachse und der Folgeachse liegt.

Standardmäßig wird der Positionsversatz und der Schleppabstand durch eine lineare Extrapolation des Leitwerts um diese Totzeit kompensiert, d. h. die Totzeitkompensation bei Leitwertkopplung ist aktiv. Zum Ausschalten der Totzeitkompensation wird folgende MD-Einstellung gesetzt:

MD37160 \$MA_LEAD_FUNCTION_MASK Bit 0 = 0

Schnittstelle zum Achstausch

Eine Leitwert-gekoppelte Folgeachse erhält ihre Sollwerte aus Kurventabellen. **Eine überlagerte Programmierung dieser Achse im Teileprogramm ist nicht möglich.** Damit wird die Leitwert-gekoppelte Folgeachse genauso wie beim Achstausch aus dem Kanal abgemeldet. Das erfolgt automatisch, wenn die Kopplung im Teileprogramm eingeschaltet wird.

Wenn die Kopplung mit Synchronaktion eingeschaltet werden soll, muss sie zuvor mit `RELEASE` dafür vorbereitet werden.

Nachdem eine Leitwertkopplung ausgeschaltet worden ist, kann die ehemalige Folgeachse wieder im Teileprogramm programmiert werden.

Spindeln in der Leitwertkopplung

Als Leitwert-gekoppelte Folgeachse kann eine Spindel nur dann verwendet werden, wenn sie zuvor in den Achsbetrieb geschaltet worden ist. Es gilt dann der Maschinendaten-Parametersatz des Achsbetriebs.

Beispiel: Einschalten aus Synchronaktion

Programmcode	Kommentar
SPOS=0	
B=IC(0)	; Spindel in Achsbetrieb schalten.
RELEASE(Y)	; Für Synchronaktion freischalten.
ID=1 WHEN (\$AA_IM[X]<-50) DO LEADON(B,X,2)	; Y wird mittels Kurventabelle Nr.2 an X gekoppelt.

10.3.3 Programmierung

Definition und Einschalten

Die Definition und das Einschalten einer Achs-Leitwertkopplung erfolgt gleichzeitig mit dem modal wirksamen Sprachbefehl `LEADON`.

Syntax:

`LEADON (<FA>, <LA>, <CTABn>)`

Bedeutung:

- <FA> Folgeachse als Geometrie-, Kanal- oder Maschinenachsname (X,Y,Z,...)
- <LA> Leitachse als Geometrie-, Kanal- oder Maschinenachsname (X,Y,Z,...)
Auch Softwareachse möglich:
MD30130 \$MA_CTRLOUT_TYPE=0 (Ausgabeart des Sollwerts)
- <CTABn> Nummer der Kurventabelle
Wertebereich: 1 bis 999

Beispiel:

Programmcode	Kommentar
LEADON(Y,X,1)	; Definition und Einschalten einer Leitwertkopplung zwischen der Leitachse X und der Folgeachse Y. Zur Berechnung des Folgewerts soll die Kurventabelle Nr.1 verwendet werden.

Randbedingungen:

- Zum Einschalten der Kopplung sind keine Referenzpunkte nötig.
- Eine definierte Folgeachse kann nicht in der Betriebsart JOG verfahren werden (auch dann nicht, wenn die Nahtstellensignale "Synchronlauf fein" bzw. "Synchronlauf grob" noch nicht anstehen).
- Eine eingeschaltete Kopplung muss erst mit `LEADOF` wieder ausgeschaltet werden, bevor diese wieder mit `LEADON` eingeschalten werden kann. Diesbezüglich sind die Einstellungen in den folgenden Maschinendaten zu beachten:
`MD20112 MC_START_MODE_MASK` (Festlegung der Steuerungs-Grundstellung bei NC-START)
`MD20110 $MC_RESET_MODE_MASK` (Festlegung der Steuerungsgrundstellung nach Reset/TP-Ende)

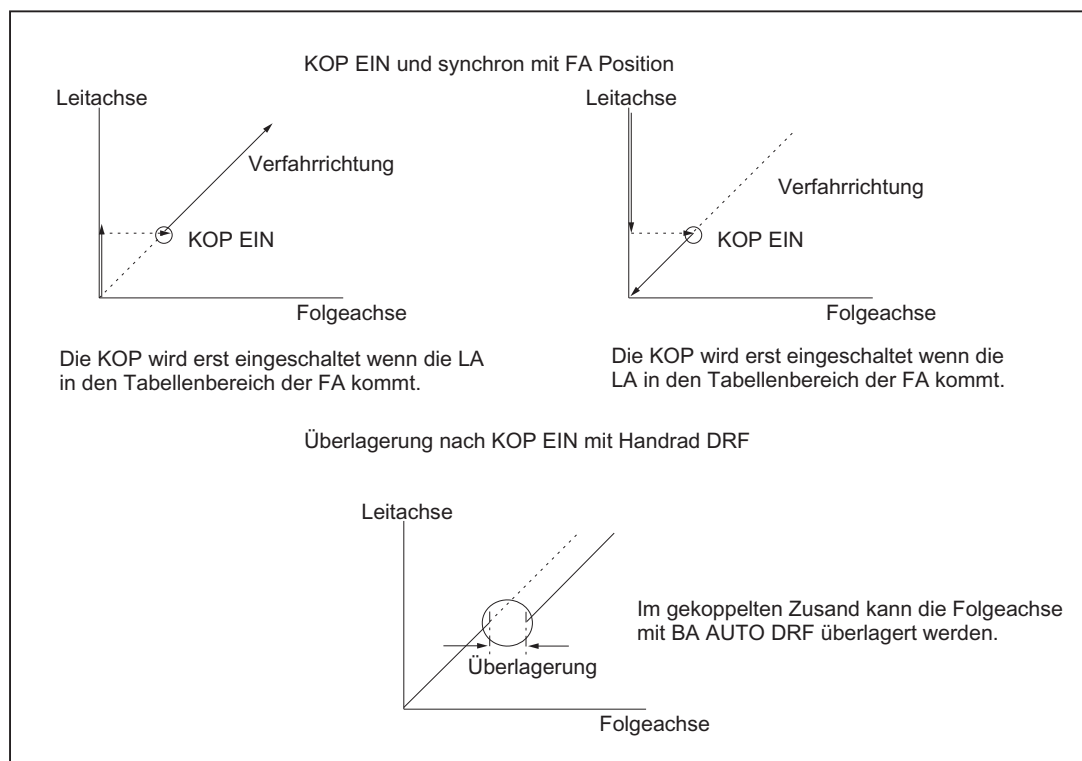


Bild 10-6 Aktivierung Leitwertkopplung

Ausschalten

Das Ausschalten einer Achs-Leitwertkopplung erfolgt mit dem Sprachbefehl `LEADOF`.

Beim Ausschalten der Achs-Leitwertkopplung wird die Folgeachse zur Kommandoachse und es wird implizit ein Stopp-Kommando für die Folgeachse erzeugt. Das Stopp-Kommando kann per Synchronaktion durch ein anderes Kommando überschrieben werden.

Syntax:

`LEADON (<FA>, <LA>)`

Bedeutung:

- <FA> Folgeachse als Geometrie-, Kanal- oder Maschinenachsname (X,Y,Z,...)
 - <LA> Leitachse als Geometrie-, Kanal- oder Maschinenachsname (X,Y,Z,...)
- Auch Softwareachse möglich:
 MD30130 \$MA_CTRLOUT_TYPE=0 (Ausgabeart des Sollwerts)

Beispiel:

Programmcode	Kommentar
LEADOF(Y,X,1)	; Ausschalten der Leitwertkopplung zwischen der Leitachse X und der Folgeachse Y.

Hinweis

Ein-/Ausschalten der Achs-Leitwertkopplung mit LEADON / LEADOF ist sowohl im Teileprogramm als auch in Synchronaktion zulässig.

Literatur:

Funktionshandbuch Synchronaktionen

Koppelart

Die Koppelart wird festgelegt mit dem achsspezifischen Settingdatum:

SD43100 \$SA_LEAD_TYPE[<LA>] (Art des Leitwerts)

<LA>: Leitachse als Geometrieachsname, Kanalachsname oder Maschinenachsname (X,Y,Z,...)

Wert	Bedeutung
0	Istwertkopplung (bei externen Leitachsen ist nur diese Koppelart möglich)
1	Sollwertkopplung (Standardeinstellung)
2	Simulierter Leitwert (Beachte virtuelle Achse, für FA nicht bewertet)

Ein Umschalten zwischen Ist- und Sollwertkopplung ist jederzeit möglich (bevorzugt in der Rastphase).

Systemvariablen des Leitwerts

Die folgenden Leitwert-Systemvariablen können vom Teileprogramm und aus Synchronaktionen nur gelesen werden:

Systemvariable	Bedeutung
\$AA_LEAD_V[ax]	Geschwindigkeit der Leitachse
\$AA_LEAD_P[ax]	Position der Leitachse
\$AA_LEAD_P_TURN	Leitwert-Position Anteil, der bei Moduloreaktion abgezogen wird. Die tatsächliche (nicht moduloreduzierte) Position des Leitwerts ist: $\$AA_LEAD_P_TURN + \AA_LEAD_P

Die Geschwindigkeiten und Positionen simulierter Leitwerte (bei \$SA_LEAD_TYPE[ax]=2) können im Teileprogramm und aus Synchronaktionen gelesen **und** geschrieben werden.

Systemvariable	Bedeutung
\$AA_LEAD_SV[ax]	Simulierte Leitwert-Geschwindigkeit pro IPO-Takt
\$AA_LEAD_SP[ax]	Simulierte Position in MKS

Systemvariablen der Folgeachse

Die folgenden Systemvariablen können für die Folgeachse im Teileprogramm und Synchronaktion gelesen werden:

Systemvariable	Bedeutung
\$AA_SYNC[ax]	Zustand der Kopplung zwischen Folge- und Leitachsen
	Wert Bedeutung
	0 nicht synchronisiert
	1 "Synchronlauf grob" ¹⁾
	2 "Synchronlauf fein" ²⁾
	3 "Synchronlauf grob" ¹⁾ UND "Synchronlauf fein" ²⁾
	Korrespondiert mit:
1) • MD37200 \$MA_COUPLE_POS_TOL_COARSE • DB31, ... DBX98.1 (Synchronlauf grob)	
2) • MD37210 \$MA_COUPLE_POS_TOL_FINE • DB31, ... DBX98.0 (Synchronlauf fein)	
\$AA_IN_SYNC[ax]	Zustand der Synchronisation zwischen Folge- und Leitachsen
	Wert Bedeutung
	0 Synchronisation ist nicht gestartet oder beendet.
	1 Synchronisation läuft, d. h. die Folgeachse wird aufsynchronisiert.
	\$AA_IN_SYNC[ax] entspricht dem NC/PLC-Nahtstellensignal: DB31, ... DBX99.4 (Synchronisation läuft)

Hinweis

Wenn die Folgeachse keine Fahrfreigabe hat, wird sie angehalten und ist nicht mehr synchron.

10.3.4 Verhalten in den Betriebsarten AUTOMATIK, MDA, JOG

Wirksamkeit

Eine Leitwertkopplung ist in Abhängigkeit von den Einstellungen im Teileprogramm und folgenden Maschinendaten aktiv:

MD20110 \$MC_RESET_MODE_MASK (Festlegung der Steuerungs-Grundstellung nach Reset/TP-Ende)

MD20112 \$MC_START_MODE_MASK (Festlegung der Steuerungs-Grundstellung bei NC-START)

Handbetrieb

Bei einer aktivierten Achs-Leitwertkopplung bewirkt ein Verfahren der Leitachse (z. B. mit Eilgang oder Schrittmaß INC1 ... INC10000) ein Verfahren der Folgeachse mit Berücksichtigung der Definition der Kurventabelle.

Referenzieren

Eine Leitwertgekoppelte Folgeachse soll vor Einschaltung der Kopplung referenziert sein. Bei eingeschalteter Kopplung kann die Folgeachse nicht referenziert werden.

Restweglöschen

Restweglöschen einer Leitachse bewirkt ein Stillsetzen aller Achsen der zugehörigen aktivierten Leitwertkopplung.

Grundstellung nach Power On

Nach Hochlauf sind keine Leitwertkopplungen aktiv. (Möglichkeiten durch Asup).

Verhalten bei NC-Start/RESET

Folgendes Verhalten ergibt sich abhängig von der Einstellung im Maschinendatum:

MD20110 \$MC_RESET_MODE_MASK (Bit 13) (Festlegung der Steuerungs-Grundstellung nach Reset/TP-Ende)

MD20112 \$MC_START_MODE_MASK (Bit 13) (Festlegung der Steuerungs-Grundstellung bei NC-START)

- MD20110 \$MC_RESET_MODE_MASK=2001H
&&
MD20112 \$MC_START_MODE_MASK=0H
→ Leitwertkopplung bleibt über `RESET` und über `START` erhalten
- MD20110 \$MC_RESET_MODE_MASK=2001H
&&
MD20112 \$MC_START_MODE_MASK=2000H
→ Leitwertkopplung bleibt über `RESET` erhalten und wird bei `START` aufgelöst. Per `IDS=...` eingeschaltete Leitwertkopplung bleibt jedoch erhalten.
- MD20110 \$MC_RESET_MODE_MASK=1H
→ Leitwertkopplung wird bei `RESET` aufgelöst und zwar unabhängig vom Maschinendatum:
MD20112 \$MC_START_MODE_MASK
Per `IDS=...` eingeschaltete Leitwertkopplung wird nur bei Bedientafelfront-Reset ausgeschaltet und bleibt bei Programmende/Reset (M30, M02) erhalten.
- MD20110 \$MC_RESET_MODE_MASK=0H
→ Leitwertkopplung bleibt über `RESET` erhalten und wird bei `START` aufgelöst, und zwar unabhängig vom Maschinendatum:
MD20112 \$MC_START_MODE_MASK
Per `IDS=...` eingeschaltete Leitwertkopplung bleibt jedoch erhalten.

Literatur:

Funktionshandbuch Grundfunktionen; Koordinatensysteme, Achstypen, Achskonfigurationen, ... (K2)

Ein-, Ausschalten

Leitwertkopplungen, die durch eine statische Synchronaktion (`IDS=...`) eingeschaltet worden sind, werden:

- bei Programmstart nicht ausgeschaltet, unabhängig vom Wert der Maschinendaten:
MD20110 \$MC_RESET_MODE_MASK (Festlegung der Steuerungs-Grundstellung nach Reset/TP-Ende)
und
MD20112 \$MC_START_MODE_MASK (Festlegung der Steuerungs-Grundstellung bei NC-START)
- bei Programmende-Reset (M30, M02) nicht ausgeschaltet, unabhängig vom Wert des Maschinendatums:
MD20110 \$MC_RESET_MODE_MASK

10.3.5 Wirksamkeit der PLC-Nahtstellensignale

Leitachse

Bei einer aktivierten Leitwertkopplung wirken die NST der Leitachse durch die Achskopplung auf die zugehörige Folgeachse. D. h.:

- eine Vorschubbeeinflussung der Leitachse bewirkt über die Leitwertkopplung eine entsprechende Vorschubbeeinflussung der Folgeachse.
- ein Stillsetzen der Leitachse durch NST (z. B. achsspezifischer Vorschub Halt, Achsensperre, Reglerfreigabe usw.) bewirkt ein Stillsetzen der zugehörigen Folgeachse.

Lagemesssystem 1/2 (DB31, ... DBX1.5/1.6)

Für einen aktivierten Achsverband ist eine Umschaltung des Lagemesssystems für Leit- und Folgeachsen nicht verriegelt. Die Kopplung bleibt dabei erhalten.

Empfehlung: Die Umschaltung bei ausgeschalteter Kopplung vornehmen.

10.3.6 Besonderheiten der Funktion Achs-Leitwertkopplung

Regeldynamik

Für einen Achsverband ist es je nach Anwendung sinnvoll die Lageregler-Parameter von Leit- und Folgeachse (z. B. K_V -Faktor) aufeinander abzugleichen. Gegebenenfalls sind für die Folgeachse andere Parametersätze zu aktivieren. Die Folgeachse sollte die gleiche oder bessere Dynamik als die Leitachse haben.

Status der Kopplung

siehe Kapitel "Kopplungsstatus (Seite 416)"

Istwertanzeige

Die Anzeige der Istposition wird für alle Achsen eines leitwertgekoppelten Achsverbandes (nur reale Achsen) aktualisiert.

Interpolation

Bei der Interpolation der in der Kurventabelle definierten Bewegung wird zu einem Leitwert und dessen Geschwindigkeit eine Achsposition und Achsgeschwindigkeit berechnet.

Archivierung

Die durch die Definition von Bewegungsabläufen erzeugten Kurventabellen werden im gepufferten Speicher abgelegt.

Kurventabellen bleiben nach dem Ausschalten der Steuerung erhalten.

Diese Funktionen spielen zum einen bei zyklischen Maschinen keine Rolle, weil sie ohne Bedieneingriffe betrieben werden; zum anderen macht es bei externen Leitwerten keinen Sinn, automatische (Re-)Positionierungen durch die NC vorzunehmen.

10.3.7 Randbedingungen

Externe Leitwertachsen

Bei Verwendung der Teileprogrammanweisungen `REPOS` bzw. `REPOSA` ist im Zusammenhang mit externen Leitwertachsen darauf zu achten, dass diese zuvor durch die Teileprogrammanweisungen `RELEASE` vom Kanal freigegeben bzw. in den "neutralen Zustand" versetzt werden.

Ohne Freigabe der Achse wird beim Versuch des Repositionierens die Meldung "Warten: Vorschub Stopp aktiv" angezeigt und die Bearbeitung des Teileprogramms nicht fortgesetzt.

10.4 Elektronisches Getriebe (EG)

10.4.1 Kurzbeschreibung

10.4.1.1 Funktion

Allgemein

Mit Hilfe der Funktion "Elektronisches Getriebe" ist es möglich, die Bewegung einer Folgeachse abhängig von bis zu fünf Leitachsen zu steuern. Die Zusammenhänge zwischen den Leitachsen und der Folgeachse sind je Leitachse durch den Koppelfaktor definiert. Die aus den einzelnen Leitachsen-Bewegungsanteilen errechneten Folgeachs-Bewegungsanteile wirken additiv.

Die Kopplung kann bezogen werden auf:

- Istwert der Leitachse
- Sollwert der Leitachse

Folgende Funktionen eines Getriebeverbandes sind über Teileprogrammanweisungen programmierbar:

- Definieren
- Einschalten
- Ausschalten
- Löschen

Kurventabellen

Zwischen Leit- und Folgeachsen können über Kurventabellen auch nichtlineare Zusammenhänge realisiert werden.

Kaskadierung

Elektronische Getriebe können kaskadiert werden, d. h. die Folgeachse eines Elektronischen Getriebes kann Leitachse für ein weiteres Elektronisches Getriebe sein.

Synchronposition

Mit einer zusätzlichen Funktion kann bei der Synchronisation der Folgeachse eine Synchronposition gewählt werden:

- Nächste Teilung (Zahnlücke) zeitoptimal anfahren
- Nächste Teilung (Zahnlücke) wegoptimal anfahren
- Anfahren in positiver Achsdrehrichtung absolut
- Anfahren in negativer Achsdrehrichtung absolut
- Zeitoptimiert zur programmierten Synchronposition fahren
- Wegoptimiert zur programmierten Synchronposition fahren

Anwendungsbeispiele:

- Werkzeugmaschinen für die Zahnradfertigung
- Getriebezüge für Produktionsmaschinen

10.4.1.2 Voraussetzungen

Für die Nutzung der Funktion wird die Option "Elektronisches Getriebe" oder die entsprechende optionale Ausführung der Generischen Kopplung (siehe Thema "Voraussetzungen (Seite 484)" in der "Kurzbeschreibung" zur Generischen Kopplung) benötigt.

10.4.2 Elektronisches Getriebe (EG)

Funktion

Mit Hilfe der Funktion "Elektronisches Getriebe" kann die Bewegung einer **Folgeachse FA** abhängig von bis zu fünf **Leitachsen LA** interpoliert werden. Die Zusammenhänge zwischen den Leitachsen und der Folgeachse sind je Leitachse durch einen Koppelfaktor definiert. Die damit berechneten Folgeachs-Bewegungsanteile aus den einzelnen Leitachsen-Bewegungsanteilen wirken additiv.

$$FA_{\text{Soll}} = \text{SynPosFA} + (LA_1 \cdot \text{SynPosLA}_1) * KF_1 + \dots + (LA_5 \cdot \text{SynPosLA}_5) * KF_5$$

mit:

SynPosFA, SynPosLA_i: aus Aufruf EGONSYN (s. unten)

FA_{Soll}: Teilsollwert der Folgeachse

LA_i: Sollwert oder Istwert der i-ten Leitachse (abhängig vom Kopplungstyp (s. u.))

KF_i: Koppelfaktor der i-ten Leitachse (s. u.)

Alle Wege beziehen sich auf das Basiskoordinatensystem **BKS**.

Bei der Aktivierung eines EG-Achsverbundes kann die Synchronisation von Leitachsen und Folgeachse auf eine definierte Ausgangsposition veranlasst werden.

Ein Getriebeverband kann aus dem Teileprogramm:

- definiert,
- eingeschaltet,
- ausgeschaltet,
- gelöscht werden.

Erweiterungen


Der Einfluss jeder der bis zu 5 Leitachsen kann alternativ zu einem Übersetzungsverhältnis (KF=Zähler/Nenner) durch eine **Kurventabelle** angegeben werden.

Für jede Kurve (ungleich Sonderfall Gerade) ist damit die Möglichkeit vorhanden, in **nicht linearer** Weise auf die Folgeachse von der Leitachse einzuwirken. Die Funktion kann nur mit EGONSYN benutzt werden.

Die Funktion EG ist mit Kurventabellen mit EGON aktivierbar.

Die Funktion EGONSYNE zum Anfahren der Synchronposition der Folgeachse unter Vorgabe von Anfahrmodus ist verfügbar.

Für besondere Anwendungen kann es sinnvoll sein, den Lageregler als **PI-Regler** zu projektieren.

 VORSICHT
Anwendungsfehler
Für die Nutzung dieser Funktion sind regeltechnische Fachkenntnisse und Messungen mit Servo Trace unumgänglich.

Literatur:

- Inbetriebnahmehandbuch CNC: NC, PLC, Antrieb
- Funktionshandbuch Grundfunktionen; Geschwindigkeiten, Soll-/ Istwertsysteme, Regelung (G2)

Kopplungstyp

Die Folgeachsbeziehung kann wahlweise abgeleitet werden von den:

- Sollwerten der Leitachsen
- Istwerten der Leitachsen

Die Festlegung erfolgt mit dem Definitionsauftrag für den EG-Achsverband:

EGDEF

(siehe Kapitel "Definition eines EG-Achsverbandes (Seite 464)")

Koppelfaktor

Der Koppelfaktor wird je Leitachse des Koppelverbandes benötigt. Er ist definiert durch Zähler/ Nenner.

Die Koppelfaktorwerte Zähler und Nenner werden je Leitachse in folgenden Aktivierungsaufträgen angegeben:

EGON

EGONSYN

EGONSYNE

(siehe Kapitel "Einschalten eines EG-Achsverbandes (Seite 465)")

Anzahl der EG-Achsverbände

Zu einem Zeitpunkt können mehrere EG-Achsverbände definiert werden. Die maximale Anzahl der EG-Achsverbände wird durch das folgende Maschinendatum festgelegt:

MD11660 \$MN_NUM_EG

Es sind maximal 31 EG-Achsverbände zulässig.

Hinweis

Die Option muss freigeschaltet sein.

EG-Kaskadierung

Die Folgeachse eines EGs kann Leitachse eines anderen EGs sein. Ein ausführlicheres Beispiel hierzu finden Sie im Kapitel "Beispiele".

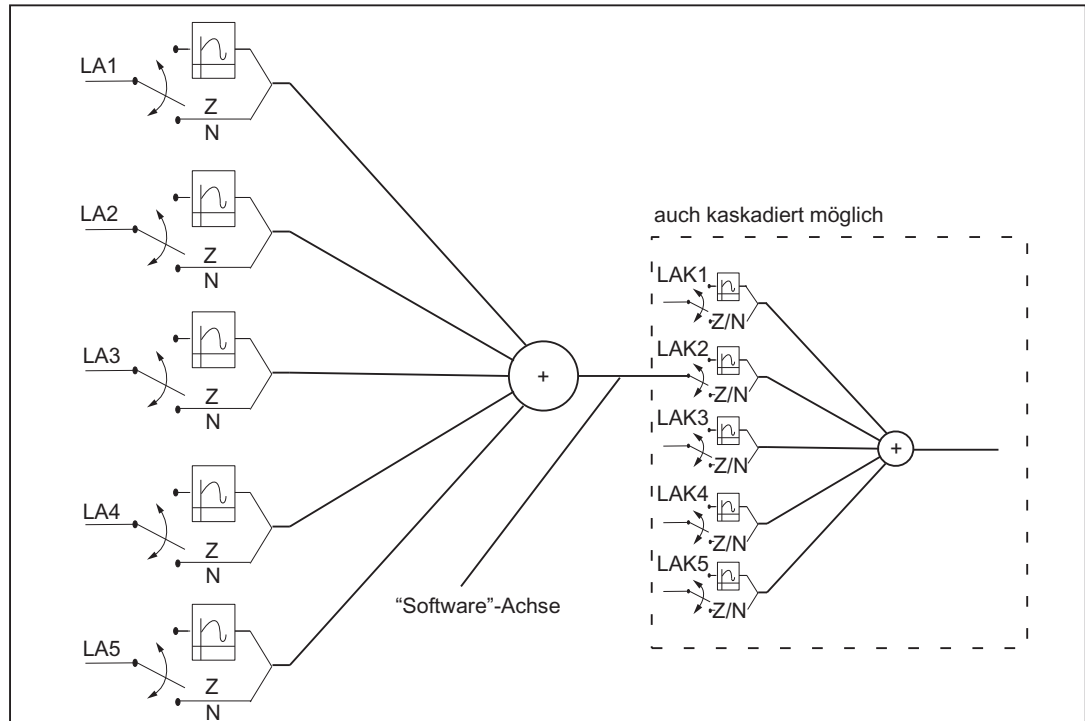


Bild 10-7 Blockschaltbild eines Elektronischen Getriebes

Synchronpositionen

Für den Anlauf des EG-Achsverbandes kann für die Folgeachse zunächst das Anfahren definierter Positionen angefordert werden.

Die Synchronpositionen werden festgelegt mit:

EGONSYN (Details siehe unten)

EGONSYNE (Erweiterter Aufruf EGONSYN)

Synchronisierung

Wenn ein Getriebe mit EGON() oder EGONSYN() bzw. EGONSYNE() s. u. eingeschaltet wird, dann ist nur in dem Fall, dass der Teileprogrammentwickler dafür sorgt, zu diesem Zeitpunkt die Ist-Position der Folgeachse genau identisch ihrer durch das Bewegungsgesetz des Getriebes aus den Positionen der Leitachsen vorgegebenen Sollposition. Die Steuerung sorgt dann durch die Bewegung der Folgeachse dafür, dass bei weiterer Bewegung der Leitachsen schnellstmöglich eine Übereinstimmung von Soll- und Ist-Position der Folgeachse erreicht wird. Dieser Vorgang wird Synchronisation genannt. Nach dem Synchronisieren der Folgeachse spricht man von einem **synchronen** Getriebe.

Einschaltvarianten

Ein elektronisches Getriebe kann auf zwei Arten aktiviert werden:

1. Aus den Achspositionen heraus, die im bisherigen Bearbeitungsverlauf erreicht wurden, wird der Befehl zur Aktivierung des EG Achsverbandes ohne Angabe von Synchronisierpositionen für die einzelnen Achsen gegeben.
EGON (siehe Kapitel "Einschalten eines EG-Achsverbandes (Seite 465)")
2. Mit dem Befehl zur Aktivierung des EG Achsverbandes werden je Achse Synchronpositionen vorgegeben. Ab dem Zeitpunkt, an dem diese Positionen erreicht sind, soll das EG synchron sein.
EGONSYN (siehe Kapitel "Einschalten eines EG-Achsverbandes (Seite 465)")
3. Mit dem Befehl zur Aktivierung des EG Achsverbandes werden je Achse Synchronpositionen und der Anfahrmodus vorgegeben. Ab dem Zeitpunkt, an dem diese Positionen erreicht sind, soll das EG synchron sein.
EGONSYNE (siehe Kapitel "Einschalten eines EG-Achsverbandes (Seite 465)")

Synchronisation bei EGON

Bei EGON() werden keine Vorgaben für die Positionen gemacht, an denen die Folgeachse synchron sein soll. Die Steuerung schaltet das EG ein und meldet "Synchronposition erreicht".

Synchronisation bei EGONSYN

1. Bei EGONSYN() werden die Positionen der Leitachsen und die Synchronisationsposition der Folgeachse durch den Befehl vorgegeben.
 - Die Steuerung fährt dann die Folgeachse mit der gerade ausreichenden Beschleunigung und Geschwindigkeit an die vorgegebene Synchronisationsposition, so dass die Folgeachse mit den Leitachsen an ihrer Synchronisationsposition ist.
 - Wenn die Folgeachse **steht**: Ist für die Folgeachse das NST "Vorschub Halt/Spindel Halt" DB31, ... DBX4.3 gesetzt, dann wird die Folgeachse auch **nicht** mit EGON oder EGONSYN in Bewegung gesetzt. Es wird ein Fahrbefehl für die Folgeachse abgesetzt und der Satzwechsel ist so lange blockiert, bis der achsspezifische Vorschub freigegeben wird. EGONSYN wird durch RESET abgebrochen und in EGON umgewandelt. Die programmierten Synchronpositionen gehen damit verloren.
 - Wenn die Folgeachse **nicht steht**: Das NST "Vorschub Halt/Spindel Halt" DB31, ... DBX4.3 hat keinen unmittelbaren Einfluss auf das Elektronische Getriebe. Es wirkt wie bisher indirekt auf die Leitachsen, soweit diese sich im gleichen Kanal befinden.
 - Für die **kanalspezifische** Vorschubfreigabe und für den Override wird nichts implementiert. Der Override hat weiterhin keinen unmittelbaren Einfluss auf das Elektronische Getriebe. Es wird die **achsspezifische** Vorschubfreigabe abhängig von der aktuellen Override-Stellung gesetzt.

Synchronisation bei EGONSYNE

Bei EGONSYNE() werden die Positionen der Leitachsen und die Synchronisationsposition der Folgeachse durch den Befehl vorgegeben.

Die Steuerung fährt die Folgeachse abhängig vom programmierten Anfahrmodus auf die Synchronposition.

Abbruch der Synchronisation bei EGONSYN und EGONSYNE

1. Unter folgenden Bedingungen wird der EGONSYN/EGONSYNE-Befehl abgebrochen und in einen EGON-Befehl gewandelt:

- RESET
- Achse geht ins Nachführen

Die definierten Synchronisationspositionen werden nicht mehr beachtet. Die Synchronlaufüberwachung berücksichtigt die Synchronisationspositionen weiterhin.

Der Abbruch der Positionssynchronisation verursacht den Alarm 16774.

Der Alarm selbst kann mit folgendem Maschinendatum unterdrückt werden:

MD11410 \$MN_SUPPRESS_ALARM_MASK Bit31 = 1

Synchronlauf-Überwachung

In jedem Interpolatortakt wird die Synchronität des Getriebes anhand der Istwerte der Folgeachse und der Leitachsen überwacht. Dazu wird das Bewegungsgesetz der Kopplung mit den Istwerten nachgerechnet. Die **Synchronlaufdifferenz** ist die Differenz des Istwerts der Folgeachse zu dem Wert, der sich nach dem Bewegungsgesetz der Kopplung aus den Istwerten der Leitachsen ergäbe. Die Synchronlaufdifferenz kann vom Teileprogramm aus abgefragt werden (siehe unten).

Änderungen der Synchronlaufdifferenz

Aufgrund von Massenträgheit der Achssysteme beim Beschleunigen kann es dynamisch zu Schwankungen in der Synchronlaufdifferenz kommen. Die Synchronlaufdifferenz wird ständig überprüft und führt über Toleranzwerte in Maschinendaten zu Nahtstellensignalen.

Die Synchronlaufdifferenz wird verglichen mit folgenden Maschinendaten:

MD37200 \$MA_COUPLE_POS_TOL_COARSE

MD37210 \$MA_COUPLE_POS_TOL_FINE

Abhängig vom Ergebnis dieses Vergleichs werden folgende Signale gesetzt:

NST "Synchronlauf fein" DB31, ... DBX98.0

NST "Synchronlauf grob" DB31, ... DBX98.1

Differenz > .. TOL_COARSE

Solange die Synchronlaufdifferenz größer als das folgende Maschinendatum ist, ist das Getriebe nicht synchron und in der Nahtstelle steht weder das NST "Synchronlauf grob" DB31, ... DBX98.1 noch das NST "Synchronlauf fein" DB31, ... DBX98.0 an:

MD37200 \$MN_COUPLE_POS_TOL_COARSE

Stattdessen wird das folgende Nahtstellensignal angezeigt:

NST "Synchronisation läuft" DB31, ... DBX99.4

Differenz < .. TOL_COARSE

Wenn die Synchronlaufdifferenz kleiner als das folgende Maschinendatum dann steht in der Nahtstelle das NST "Synchronlauf Grob" "DB31, ... DBX98.1 an und das NST "Synchronisation läuft" DB31, ... DBX99.4 wird gelöscht:

MD37200 \$MN_COUPLE_POS_TOL_COARSE

Differenz > .. TOL_FINE

Wenn die Synchronlaufdifferenz kleiner als das folgende Maschinendatum ist, dann steht in der Nahtstelle das NST "Synchronlauf fein" DB31, ... DBX98.0 an:

MD37210 \$MA COUPLE_POS_TOL_FINE

Synchronlaufdifferenz bei EG-Kaskaden

Synchronlaufdifferenz bei EG-Kaskaden ist die Abweichung der Folgeachs-Istposition von der aus dem Bewegungsgesetz resultierenden Sollposition in Abhängigkeit von den beteiligten realen Achsen.

Beispiel:

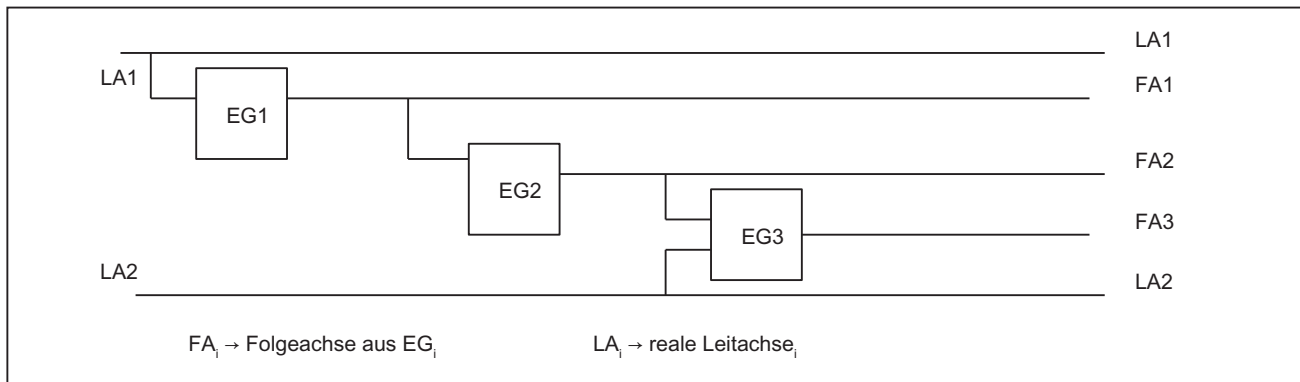


Bild 10-8 Dreistufige EG-Kaskade

Die Synchronlaufdifferenz von Folgeachse FA3 des vorstehenden Beispiels wird nach der vorn gegebenen Definition bestimmt durch den Folgeachsenistwert $FA3_{Ist}$ und die Leitachsenistwerte $FA2_{Ist}$ und $LA2_{Ist}$, nicht jedoch durch $LA1_{Ist}$ und $FA1_{Ist}$.

Wenn FA2 keine reale Achse ist, dann steht der Istwert $FA2_{Ist}$ nicht zur Verfügung. In diesem Fall muss anstatt des Istwertes der **Sollwert** der Achse, der in diesem Fall allein aus dem Leitachsenistwert $FA1_{Ist}$ abgeleitet werden kann, benutzt werden.

Weitere Signale

Beim Eintreten eines EGON(), EGONSYN() oder EGONSYNE() Satzes in den Hauptlauf wird das Signal "Kopplung Aktiv" für die Folgeachse gesetzt. Wird die Folgeachse nur überlagert, so werden die Signale "Kopplung Aktiv" und "Achse wird überlagert" gesetzt. Ist ein EGON(), EGONSYN() oder ein EGONSYNE() aktiv und die Folgeachse wird zusätzlich überlagert, so werden ebenfalls die Signale "Kopplung Aktiv" und "Achse wird überlagert" gesetzt.

NST "Folgespindel aktiv" DB31, ... DBX 99.1: Kopplung Aktiv,

NST "überlagerte Bewegung" DB31, ... DBX98.4: Achse wird überlagert,

NST "Freigabe Folgeachsüberlagerung" DB31, ... DBX26.4

Bei den Befehlen EGONSYN(), EGONSYNE() muss das Signal "Freigabe Folgeachsenüberlagerung" vorliegen, damit das Getriebe sich gemäß der vorgegebenen Folgeachsensynchronisationsposition synchronisieren darf. Liegt es nicht vor, so kommt es zum Alarm 16771 "Überlagerte Bewegung nicht freigegeben". Liegt das Signal an, so fährt die Folgeachse nach ihrer berechneten Beschleunigung und Geschwindigkeit ggf. entsprechend Anfahrmodus die Synchronposition an.

Weitere Überwachungssignale

Mit dem Maschinendatum MD37550 \$MA_EG_VEL_WARNING kann ein %-Satz der Geschwindigkeiten und Beschleunigungen in den Maschinendaten MD32000 \$MA_MAX_AX_VELO und MD32300 \$MA_MAX_AX_ACCEL bezogen auf die Folgeachse angegeben werden, bei dem die folgenden Nahtstellensignale erzeugt werden:

NST "Geschwindigkeits-Warnschwelle" DB31, ... DBX98.5

NST "Beschleunigungs-Warnschwelle" DB31, ... DBX98.6

Die Überwachungssignale können als Auslösekriterien für Notrückzug verwendet werden (siehe Kapitel "Auslösequellen (Seite 584)").

Im Maschinendatum MD37560 \$MA_EG_ACC_TOL kann ein %-Satz bezogen auf das Maschinendatum MD32300 \$MA_MAX_AX_ACCEL der Folgeachse definiert werden, bei dessen Überschreitung das Signal NST "Achse beschleunigt" DB31, ... DBX99.3 erzeugt wird.

Synchronlaufdifferenz abfragen

1. Das Rechenergebnis der Synchronlaufdifferenz kann mit der Systemvariablen \$VA_EG_SYNCDIFF als Betrag im Teileprogramm gelesen werden. Der entsprechende Wert mit Vorzeichen steht in der Systemvariablen \$VA_EG_SYNCDIFF_S zur Verfügung. Es bedeutet:

- Negativer Wert (bei Leitachse und Folgeachse in positiver Bewegungsrichtung): Die Folgeachse bleibt hinter ihrer errechneten Sollposition zurück.
- Positiver Wert (bei Leitachse und Folgeachse in positiver Bewegungsrichtung): Die Folgeachse ist vor ihrer errechneten Sollposition (Überschwinger).

Der Betrag aus der Synchronlaufdifferenz mit Vorzeichen entspricht der Systemvariablen ohne Vorzeichen aus \$VA_EG_SYNCDIFF.

$\$VA_EG_SYNCDIFF[ax] = ABS(\$VA_EG_SYNCDIFF_S[ax])$

Satzwechselmodus

1. Beim Einschalten eines EG-Koppelverbandes kann angegeben werden, unter welchen Bedingungen der Satzwechsel in der Abarbeitung des Teileprogramms ausgelöst werden soll:
2. Die Angaben erfolgen durch Stringparameter mit folgenden Bedeutungen:
3. "NOC": Satzwechsel sofort
4. "FINE": Satzwechsel, wenn "Synchronlauf fein" vorliegt
5. "COARSE": Satzwechsel, wenn "Synchronlauf grob" vorliegt
6. "IPOSTOP": Satzwechsel, wenn "sollwertseitiger Synchronlauf" vorliegt

Hinweis

Bei der Programmierung in den Aktivierungsaufrufen EGON, EGONSYN, EGONSYNE können die angegebenen Strings auf die jeweils ersten beiden Zeichen verkürzt angegeben werden.

War für den EG-Achsverband noch kein Satzwechselmodus definiert und ist auch aktuell keiner angegeben, so gilt "FINE".

10.4.3 Definition eines EG-Achsverbandes

Hinweis

Die folgenden Definitionsbefehle und Schaltenweisungsbefehle des Elektronischen Getriebes müssen in einem Teileprogramm **allein in einem Satz stehen**.

Alle Befehle des Elektronischen Getriebes lösen **Vorlaufstopp** aus, mit Ausnahme der Aktivierungsbefehle:

- EGON
 - EGONSYN
 - EGONSYNE
-

Definition und Aktivierung

Die nachfolgend beschriebene Definition und die Aktivierung sind eigenständige Vorgänge. Eine Aktivierung ohne vorherige Definition ist nicht möglich.

Definition eines EG-Achsverbandes

Ein EG-Achsverband wird durch die Angabe der Folgeachse und mindestens einer, höchstens fünf Leitachsen mit dem jeweiligen Kopplungstyp festgelegt:

EGDEF(Folgeachse, Leitachse1, Kopplungstyp1, Leitachse2, Kopplungstyp2,..)

Der Kopplungstyp muss nicht für alle Leitachsen gleich sein und ist daher für jede Leitachse einzeln anzugeben.

Kopplungstyp:

Istwert der Leitachse auswerten: 0

Sollwert der Leitachse auswerten: 1

Die Koppelfaktoren werden bei der Definition des EG-Kopplungsverbandes mit Null vorbesetzt. Damit tritt bis zur Aktivierung noch keine Einwirkung auf die Folgeachse auf. (Siehe EGON, EGONSYN, EGONSYNE).

Voraussetzung für eine EG-Achsverbands-Definition:

Für die Folgeachse darf noch keine Achskopplung definiert sein. (Ggf. muss eine bestehende vorher mit EGDEL gelöscht werden.)

EGDEF löst Vorlaufstopp mit Fehlermeldung aus.

Ein Beispiel zur Anwendung des EG-Getriebes beim Wälzfräsen finden Sie im Kapitel "Beispiele", "Elektronische Getriebe für Wälzfräsen".

EGDEF

Die Getriebedefinition mit EGDEF ist auch dann unverändert zu verwenden, wenn eine oder mehrere Leitachsen über Kurventabelle auf die Folgeachse einwirken.

Die um nichtlineare Kopplung mittels Kurventabellen erweiterte Variante finden Sie in einem erweiterten Beispiel im Kapitel "Erweitertes Beispiel mit nicht linearen Anteilen".

10.4.4 Einschalten eines EG-Achsverbandes

Ohne Synchronisation

Der EG-Koppelpverband wird **ohne Synchronisationselektiv** eingeschaltet mit:

EGON(FA, Satzwechselmodus, LA1, Z1, N1, LA2 , Z2, N2,..LA5, Z5, N5.)

Die Kopplung wird sofort aktiviert.

Mit:

FA: Folgeachse

Je nach Satzwechselmodus wird der nächste Satz weitergeschaltet:

"NOC": Satzwechsel erfolgt sofort

"FINE": Satzwechsel erfolgt bei "Synchronlauf fein"

"COARSE": Satzwechsel erfolgt bei "Synchronlauf grob"

"IPOSTOP": Satzwechsel erfolgt bei sollwertseitigem Synchronlauf

LA_i: Achsname der Leitachse i

Z_i: Zähler für den Koppelfaktor der Leitachse i

N_i : Nenner für den Koppelfaktor der Leitachse i

Es dürfen nur die Leitachsen programmiert werden, die zuvor mit EGDEF spezifiziert worden sind. Es muss mindestens eine Leitachse programmiert werden.

Die Positionen der Leitachsen sowie der Folgeachse zum Zeitpunkt des Einschaltens werden gespeichert als "Synchronpositionen". Die "Synchronpositionen" können mit den Systemvariablen \$AA_EG_SYN gelesen werden.

Mit Synchronisation

Der EG-Kopplverband wird mit **Synchronisation selektiv** eingeschaltet mit:

1. EGONSYN

EGONSYN(FA, Satzwechselmodus, SynPosFA, LA_i , SynPos LA_i , Z_ LA_i , N_ LA_i)

Mit:

FA: Folgeachse

Satzwechselmodus:

"NOC": Satzwechsel erfolgt sofort

"FINE": Satzwechsel erfolgt bei "Synchronlauf fein"

"COARSE": Satzwechsel erfolgt bei "Synchronlauf grob"

"IPOSTOP": Satzwechsel erfolgt bei sollwertseitigem Synchronlauf

SynPosFA: Synchronposition der Folgeachse

LA_i : Achsname der Leitachse i

SynPos LA_i : Synchronposition der Leitachse i

Z i : Zähler für den Koppelfaktor der Leitachse i

N i : Nenner für den Koppelfaktor der Leitachse i

Hinweis

Die mit i indizierten Parameter müssen mindestens für eine Leitachse und dürfen höchstens für fünf Leitachsen angegeben werden.

Es dürfen nur Leitachsen programmiert werden, die zuvor mit EGDEF spezifiziert worden sind.

Durch die programmierten "Synchronpositionen" für die Folgeachse (SynPosFA) und für die Leitachsen (SynPosLA) werden Positionen definiert, in denen der Koppelverband als *synchron* gilt. Sofern sich das elektronische Getriebe beim Einschalten nicht in synchronem Zustand befindet, fährt die **Folgeachse** auf ihre definierte Synchronposition.

Die Positionsangabe der Synchronpositionen erfolgt unabhängig von der programmierbaren Maßangabe (G70/G71) im eingestellten Grundsystem.

Wenn Moduloachsen im Koppelverband sind, werden ihre Positionswerte im modulo reduziert. Damit wird zuverlässig die schnellstmögliche Synchronposition angefahren, z. B. den nächsten

Zahn aus dem Zahnabstand ($360 \text{ Grad} * Zi / Ni$) und den zugehörigen Synchronpositionen.
(Sog. *relative Synchronisation*.)

erreicht, dass die schnellstmögliche Synchronposition angefahren wird. (Sog. *relative Synchronisation*, z. B. die nächste Zahnücke nach "Einmitten".)

Wenn für die Folgeachse nicht das folgende Nahtstellensignal gegeben ist, wird nicht auf die Synchronposition gefahren:

DB31, ... DBX26.4 (Freigabe Folgeachsüberlagerung)

Stattdessen wird das Programm beim EGONSYN-Satz angehalten und es wird der selbstlöschende Alarm 16771 so lange gemeldet, bis das o. g. Signal gesetzt wird.

2. EGONSYNE

EGONSYNE(FA, Satzwechselmodus, SynPosFA, Anfahrmodus, LA_i, SynPosLA_i, Z_LA_i, N_LA_i)

mit:

"FA": Folgeachse

Satzwechselmodus:

"NOC": Satzwechsel erfolgt sofort

"FINE": Satzwechsel erfolgt bei "Synchronlauf fein"

"COARSE": Satzwechsel erfolgt bei "Synchronlauf grob"

"IPOSTOP": Satzwechsel erfolgt bei sollwertseitigem Synchronlauf

SynPosFA: Synchronposition der Folgeachse

Anfahrmodus:

"NTGT": NextToothGapTime optimized, nächste Zahnücke wird zeitoptimiert angefahren (Vorbesetzungswert wirkt, wenn nichts angegeben ist.)

"NTGP": NextToothGapPath optimized, nächste Zahnücke wird wegoptimiert angefahren

"ACN": AbsolutCoordinateNegativ, Absolutmaßangabe, Rundachse verfährt in negativer Achsdrehrichtung

"ACP": AbsulteCoordinatePositiv, Absolutmaßangabe, Rundachse verfährt in positiver Achsdrehrichtung

"DCT": DirectCoordinateTime optimized, Absolutmaßangabe, Rundachse verfährt zeitoptimiert zur programmierten Synchronposition

"DCP": DirectCoordinatePath optimized, Absolutmaßangabe, Rundachse verfährt wegoptimiert zur programmierten Synchronposition

LA_i: Achsname der Leitachse i

SynPosLA_i: Synchronposition der Leitachse i

Zi: Zähler für den Koppelfaktor der Leitachse i

Ni: Nenner für den Koppelfaktor der Leitachse i

Hinweis

Die mit i indizierten Parameter müssen mindestens für eine Leitachse und dürfen höchstens für fünf Leitachsen angegeben werden.

Die Funktion wirkt nur auf Modulo-Folgeachsen, die an Modulo-Leitachsen gekoppelt sind.

Zahnabstand

Der Zahnabstand ergibt sich aus $360 \text{ Grad} * Zi / Ni$

Beispiel:

EGONSYNE(A, "FINE", FASynPos, "Verfahrmodus", B, LASynPos, 2, 10)

Zahnabstand: $360 * 2 / 10 = 72 \text{ (Grad)}$

Anfahrverhalten bei stehender FA

In diesem Fall sind die Verfahrmodi zeitoptimiert und wegoptimiert identisch.

Die folgende Tabelle zeigt die Zielpositionen und gefahrenen Wege mit Richtungskennzeichnung (in Klammern) abhängig vom Anfahrmodus:

Programmierte Synchronposition FaSy-sPos	Position der Folgeachse vor EGONSYNE	Verfahrmodus NTGT/NTGP	Verfahrmodus DCT/DCP	Verfahrmodus ACP	Verfahrmodus ACN
110	150	182 (+32)	110 (-40)	110 (+320)	110 (-40)
110	350	326 (-24)	110 (+120)	110 (+120)	110 (-240)
130	0	346 (-14)	130 (+130)	130 (+130)	130 (-230)
130	30	58 (+28)	130 (+100)	130 (+100)	130 (-260)
130	190	202 (+12)	130 (-60)	130 (+300)	130 (-60)
190	0	334 (-26)	190 (-170)	190 (+190)	190 (-170)
230	0	14 (+14)	230 (-130)	230 (+230)	230 (-130)

Anfahrverhalten bei fahrender FA

Die Folgeachse bewege sich bei Aktivierung der Kopplung durch EGONSYNE annähernd mit ihrer maximalen Achsgeschwindigkeit in positiver Richtung. Die programmierte Synchronposition der Folgeachse sei 110, die aktuelle Position 150. Damit ergeben sich die zwei alternativen Synchronpositionen 110 und 182 (siehe obige Tabelle).

Mit dem Verfahrmodus NTGP (wegoptimiert), wird unabhängig von der aktuellen Geschwindigkeit die Synchronposition 182 ausgewählt. Diese hat den kürzesten Abstand zur aktuellen Position der Folgeachse. Mit Verfahrmodus NTGT (zeitoptimiert) wird die aktuelle Geschwindigkeit der Folgeachse berücksichtigt und aufgrund der Begrenzung durch die maximale Geschwindigkeit der Achse, ein Abbremsen realisiert, um in möglichst kürzester Zeit Synchronlauf zu erreichen (siehe Bild).

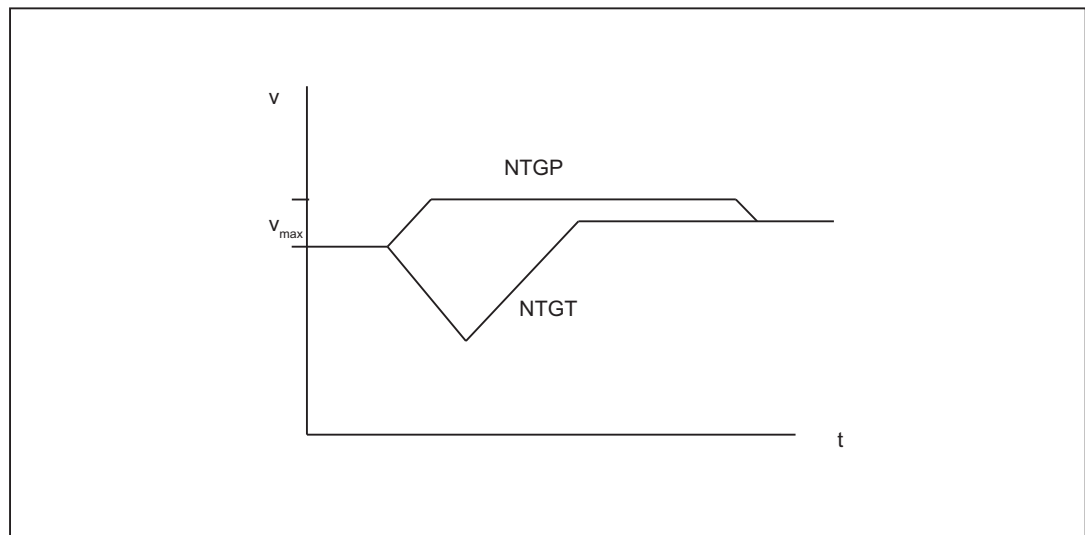


Bild 10-9 Erreichen der nächsten Zahnücke FA wegoptimiert (oben) vs. zeitoptimiert (unten)

Beispielnotationen

EGONSYNE(A, "FINE", 110, "NTGT", B, 0, 2, 10)

kopple A an B, Synchronposition A = 110, B = 0, Kopplungsfaktor 2/10, Anfahrmodus = NTGT

EGONSYNE(A, "FINE", 110, "DCT", B, 0, 2, 10)

kopple A an B, Synchronposition A = 110, B = 0, Kopplungsfaktor 2/10, Anfahrmodus = DCT

EGONSYNE(A, "FINE", 110, "NTGT", B, 0, 2, 10, Y, 15, 1, 3)

kopple A an B und Y, Synchronposition A = 110, B = 0, Y = 15,

Kopplungsfaktor zu B = 2/10, Kopplungsfaktor zu Y = 1/3,

Anfahrmodus = NTGT

Mit Synchronisation

Es gilt die oben angegebene Syntax mit folgenden Bedeutungsabweichungen:

Wird für eine der Leitachsen eine **Kurventabelle** verwendet, so muss der Nenner des Koppelfaktors linearer Kopplungen (N_i) auf 0 gesetzt werden (Nenner 0 wäre für lineare Kopplungen unzulässig).

Nenner Null ist für die Steuerung das Kennzeichen, dass der Zähler für den Koppelfaktor (Z_i) als Nummer der zu verwendenden Kurventabelle interpretiert werden soll. Die Kurventabelle mit der angegebenen Nummer muss zum Einschaltzeitpunkt bereits definiert sein (gemäß Kapitel "Kurventabellen").

Die Angabe der Leitachse (LA_i) entspricht der Leitachsangabe bei Kopplung über Koppelfaktor (lineare Kopplung).

10.4.5 Ausschalten eines EG-Achsverbandes

Variante 1

Für das Ausschalten eines aktiven EG-Achsverbandes gibt es verschiedene Möglichkeiten.

EGOFS(Folgeachse)

Das elektronische Getriebe wird ausgeschaltet. Die Folgeachse wird zum Stillstand abgebremst. Der Aufruf löst Vorlaufstopp aus.

Variante 2

Die folgende Parametrierung des Befehls erlaubt **selektiv** den Einfluss einzelner Leitachsen auf die Bewegung der Folgeachse zu unterbinden.

EGOFS(Folgeachse, Leitachse 1, ... Leitachse 5)

Hinweis

Es muss wenigstens eine Leitachse angegeben werden.

Der Einfluss der angegebenen Leitachsen auf die Folgeachse wird gezielt ausgeschaltet. Der Aufruf löst Vorlaufstopp aus.

Verbleiben noch aktive Leitachsen, so läuft die Folgeachse unter deren Einfluss weiter. Sind alle Leitachseneinflüsse auf diese Weise ausgeschaltet, so wird die Folgeachse zum Stillstand abgebremst.

Wird der Befehl EGONSYN selektiv ausgeschaltet, wird keine Achsbewegung ausgeführt.

Variante 3

EGOFC(Folgespindel)

Das elektronische Getriebe wird ausgeschaltet. Die Folgespindel läuft mit der zum Ausschaltzeitpunkt aktuellen Drehzahl/Geschwindigkeit weiter. Der Aufruf löst Vorlaufstopp aus.

Hinweis

Aufruf nur für Folge-**Spindeln** verfügbar. Bei EGOFC muss ein Spindelname programmiert werden.

10.4.6 Löschen eines EG-Achsverbandes

Ein EG-Achsverband muss ausgeschaltet sein gemäß Kapitel "Ausschalten eines EG-Achsverbandes", bevor seine Definition gelöscht werden kann.

EGDEL(Folgeachse)

Die Kopplungsdefinition des Achsverbandes wird gelöscht. Es wird bis zum Erreichen der maximalen Anzahl von gleichzeitig aktivierten Achsverbänden wieder möglich, weitere Achsverbände mit EGDEF neu zu definieren.

Der Aufruf löst Vorlaufstopp aus.

10.4.7 Zusammenwirken des Umdrehungsvorschubs (G95) mit dem Elektronischen Getriebe

Mit dem Teileprogrammbefehl FPR() kann auch die Folgeachse eines Elektronischen Getriebes als vorschubbestimmende Achse des Umdrehungsvorschubes angegeben werden. Für diesen Fall gilt folgendes Verhalten:

- Der Vorschub ist abhängig von der Sollgeschwindigkeit der Folgeachse des Elektronischen Getriebes.
- Die Sollgeschwindigkeit wird berechnet aus den Geschwindigkeiten der Leitspindeln und Modulo-Leitachsen (die nicht Bahnachsen sind) und deren zugeordneten Koppelfaktoren.
- Geschwindigkeitsanteile von anderen Leitachsen und überlagerte Bewegungen der Folgeachse werden nicht berücksichtigt.

Literatur: /V1/, Vorschübe

10.4.8 Verhalten bei POWER ON, RESET, BA-Wechsel, Suchlauf

Funktion	Verhalten bezüglich Elekt. Getriebe	
	Kopplungszustand	Konfiguration
Betriebsartenwechsel	Bleibt erhalten	Bleibt erhalten
Teileprogrammende	Bleibt erhalten	Bleibt erhalten
Reset	Bleibt erhalten	Bleibt erhalten
Power On ¹⁾	Bleibt nicht erhalten	Bleibt nicht erhalten

1) Nach Power On ist **keine** Kopplung aktiv.

Satzsuchlauf

Unter folgenden Randbedingungen ist ein Satzsuchlauf bei aktiver Kopplung (EG) möglich:

- Es wird ausschließlich mit Sollwertkopplung simuliert.
- Alle kanalübergreifenden Leitachsen dürfen keiner Sperre unterliegen.
- Achsbewegungen von denen alle realen Positionen der NC bekannt sind.

Ist es technologisch nicht sinnvoll bzw. nicht möglich den Zielsatz für einen Satzsuchlauf mit Berechnung bzw. SERUPRO innerhalb eines Teileprogrammabschnittes mit aktiver Kopplung zuzulassen, kann dieser Abschnitt für das Wiederaufsetzen gesperrt werden.

Programmierung

- IPTRLOCK()
- IPTRUNLOCK()

Literatur

(K1) BAG, Kanal, Programmbetrieb, Reset-Verhalten, Kapitel "Satzsuchlauf Typ 5 SERUPRO" > "Programmabschnitt für Wiederaufsetzen sperren"

10.4.9 Systemvariablen für Elektronisches Getriebe

Anwendung

Mit Hilfe der folgenden Systemvariablen kann das Teileprogramm aktuelle Zustände eines EG-Achsverbandes ermitteln und ggf. darauf reagieren:

Tabelle 10-1 Systemvariablen, R bedeutet: Lesezugriff möglich

Name	Typ	Zugriff		Vorlaufstopp		Bedeutung, Wert	Bed. Index
		Teileprog.	Sync Akt.	Teileprog.	Sync Akt.		
\$AA_EG_TYPE[a,b]	INT	R		R		Art der Kopplung: 0: Istwertkopplung 1: Sollwertkopplung	Achsname a: Folgeachse b: Leitachse
\$AA_EG_NUMERA[a,b]	REAL	R		R		Zähler des Koppelfaktors KF KF = Zähler/Nenner Voreinstellung: 0 Nummer der Kurventabelle, wenn \$AA_EG_DENOM[a,b] 0 ist.	Achsname a: Folgeachse b: Leitachse
\$AA_EG_DENOM[a,b]	REAL	R		R		Nenner des Koppelfakt. KF KF = Zähler/Nenner Voreinstellung: 1 Nenner muss positiv sein. Nenner ist 0, wenn statt Zähler \$AA_EG_NUMERA[a,b] die Nummer einer Kurventabelle angegeben ist.	Achsname a: Folgeachse b: Leitachse
\$AA_EG_SYN[a,b]	REAL	R		R		Synchronposition für die angegebene Leitachse Voreinstellung: 0	Achsname a: Folgeachse b: Leitachse
\$AA_EG_SYNFA[a]	REAL	R		R		Synchronposition für die angegebene Folgeachse Voreinstellung: 0	Achsname a: Folgeachse

Name	Typ	Zugriff		Vorlaufstopp		Bedeutung, Wert	Bed. Index
		Teile- prog.	Sync Akt.	Teile- prog.	Sync Akt.		
\$P_EG_BC[a]	STRING	R		R		Satzwechselkriterium für EG-Aktivierungsaufrufe: EGON, EGONSYN: "NOC": sofort "FINE": Synchronlauf fein "COARSE": Synchronlauf grob "IPOSTOP": sollwertseitiger Synchronlauf	Achsname a: Folgeachse
\$AA_EG_NUM_LA[a]	INT	R		R		Anzahl der mit EGDEF definierten Leitachsen. 0, wenn keine Achse mit EGDEF als Folgeachse festgelegt wurde.	Achsname a: Folgeachse
\$AA_EG_AX[n,a]	AXIS	R		R		Achsname der Leitachse, deren Index n angegeben wurde.	Achsname n: Index der Leitachse im EG-Verband 0 ... 4 a: Folgeachse
\$AA_EG_ACTIVE[a,b]	BOOL	R		R		Einschaltzustand einer Leitachse feststellen: 0: ausgeschaltet 1: eingeschaltet	Achsname a: Folgeachse b: Leitachse
\$VA_EG_SYNCDIFF[a]	REAL	R	R	R		Istwert der Synchronlaufdifferenz. Der Vergleich mit den Maschinendaten MD37200 \$MA_COUPLE_POS_TOL_COARSE und MD37210 \$MA_COUPLE_POS_TOL_FINE liefert Nahtstellensignale.	Achsname a: Folgeachse

10.4.10 Beispiele

10.4.10.1 Beispiel mit linearen Kopplungen

Benutzung der Achsen

Im folgenden Bild ist die Konfiguration einer typischen Wälzfräsmaschine dargestellt. Die Maschine umfasst fünf numerisch geregelte Achsen und eine gesteuerte Hauptspindel. Es sind dies:

- die Drehbewegung von Werkstücktisch (C) und Wälzfräser (B)
- die Axialachse (Z) zur Erzeugung der Vorschubbewegung über die gesamte Werkstückbreite
- die Tangentialachse (Y) zur Verschiebung des Wälzfräasers längs seiner Achse

- die Radialachse (X) für die Zustellung des Fräasers auf Zahntiefe
- die Fräuserschwenkachse (A) für die Einstellung des Wälzfräasers zum Werkstück abhängig vom Fräsersteigungswinkel und dem Zahnschrägungswinkel

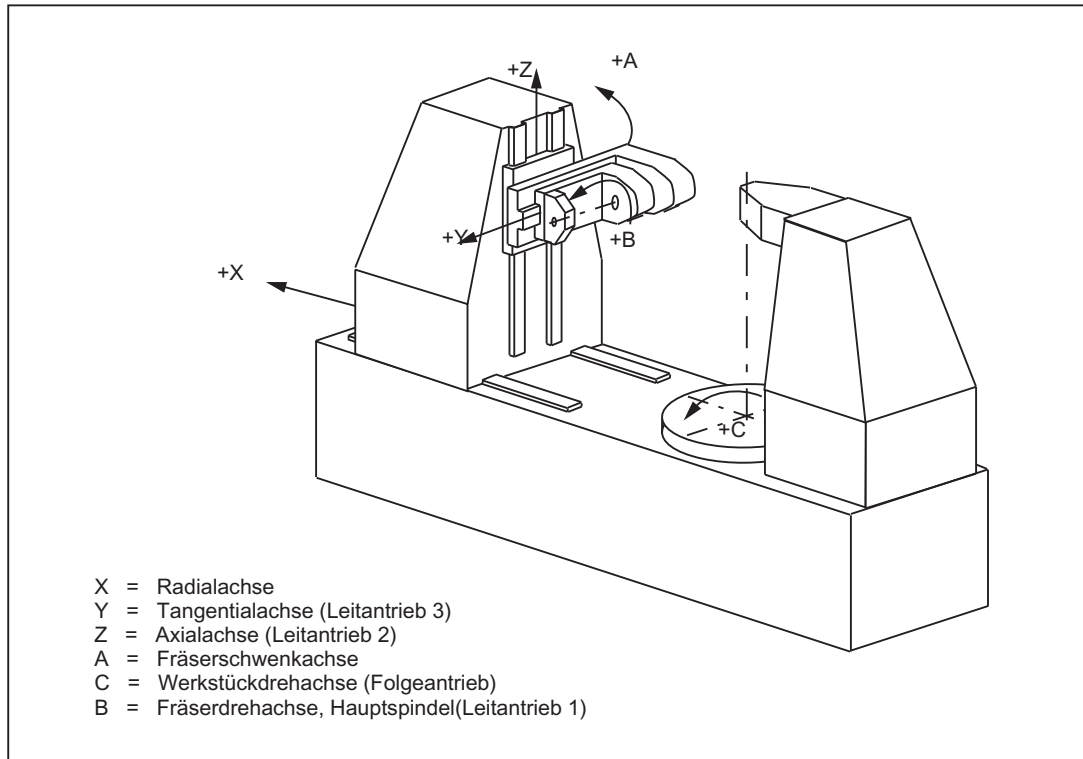
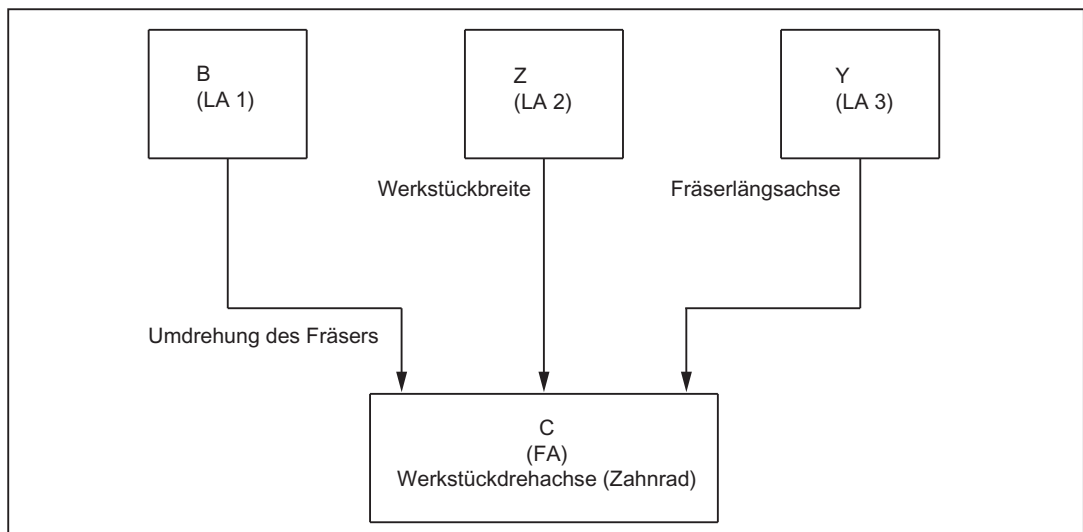


Bild 10-10 Achsendefinition einer Wälzfräsmaschine (Beispiel)

Für die Wälzfräsmaschine ergibt sich folgender Funktionszusammenhang:



Die Werkstücktischachse (C) ist hierbei die Folgeachse, die in diesem Beispiel von drei Leitantrieben beeinflusst wird.

Der Sollwert der Folgeachse wird zyklisch mit folgender Verknüpfungsgleichung ermittelt:

$$n_c = n_b * (z_0 / z_2) + v_z * (u_{dz} / z_2) + v_y * (u_{dy} / z_2)$$

mit:	n_c	= Drehzahl der Werkstückachse (C)
	n_b	= Drehzahl der Fräerspindel (B)
	z_0	= Gangzahl des Wälzfräasers
	z_2	= Zähnezahzahl des Werkstücks
	v_z	= Vorschubgeschwindigkeit der Axialachse (Z)
	v_y	= Vorschubgeschwindigkeit der Tangentialachse (Y)
	u_{dz}	= Axial-Differential-Konstante
	u_{dy}	= Tangential-Differential-Konstante

Einflüsse auf den Sollwert der Werkstückachse C

Der erste Summand der obigen Gleichung bestimmt das Drehzahlverhältnis von Werkstücktisch und Fräser und damit die Zähnezahzahl des Werkstücks.

Der zweite Summand bewirkt bei Schrägverzahnungen die notwendige Zusatzdrehung der C-Achse in Abhängigkeit der axialen Vorschubbewegung des Fräasers zur Erzeugung der Zahnschräge.

Mit dem dritten Anteil wird ebenfalls eine Zusatzdrehung der C-Achse berücksichtigt, die die Tangentialbewegung des Fräasers zum Werkstück ausgleicht. Damit kann eine gleichmäßige Beanspruchung des Werkzeugs über seine gesamte Länge erreicht werden.

Werkstück-/Werkzeugparameter

Die Werte z_0 , z_2 , u_{dz} und u_{dy} sind Werkstück- bzw. Werkzeug-abhängig und werden vom NC-Bediener bzw. im Teileprogramm vorgegeben.

Differentialkonstanten

Mit den Differentialkonstanten u_{dz} und u_{dy} werden die Zahnschräge des Werkstücks sowie die Geometrie des Fräasers berücksichtigt. Die Ermittlung der Differentialkonstanten kann in anwenderspezifischen Zyklen erfolgen.

$$u_{dz} = (\sin \beta^\circ / (m_n * \pi)) * 360 \quad [\text{Grad} / \text{mm}]$$

$$u_{dy} = (\cos \gamma^\circ / (m_n * \pi)) * 360 \quad [\text{Grad} / \text{mm}]$$

mit:	m_n	= Normalmodul (in mm)
	β°	= Schrägungswinkel des Zahnrades
	γ°	= Steigungswinkel des Wälzfräasers

Teileprogrammausschnitt

Programmcode	Kommentar
EGDEF (C,B,1,Z,1,Y,1)	; Definition des EG-Achsverbundes mit Sollwertkopplung (1) von B, Z, Y auf C (Folgeachse).
EGON (C, "FINE", B, z0, z2, Z, udz, z2, Y, udy, z2)	; Kopplung einschalten.
...	

10.4.10.2 Erweitertes Beispiel mit nicht linearen Anteilen

Einführung

Das folgende Beispiel erweitert das Beispiel (siehe "Bild 10-10 Achsdefinition einer Wälzfräsmaschine (Beispiel) (Seite 474)") um Folgendes:

- Maschinenfehlerkompensationen, die von der Z-Achse nicht linear abhängig sind, und
- eine von der Z-Achse abhängige Komponente bei der Zahngeometrie.
Mit dieser kann z. B. die Zahnfläche in der Zahnradmitte leicht ballig gestaltet werden, so dass im Betrieb die Zahnmitte mehr belasten wird als die Zahnräder.

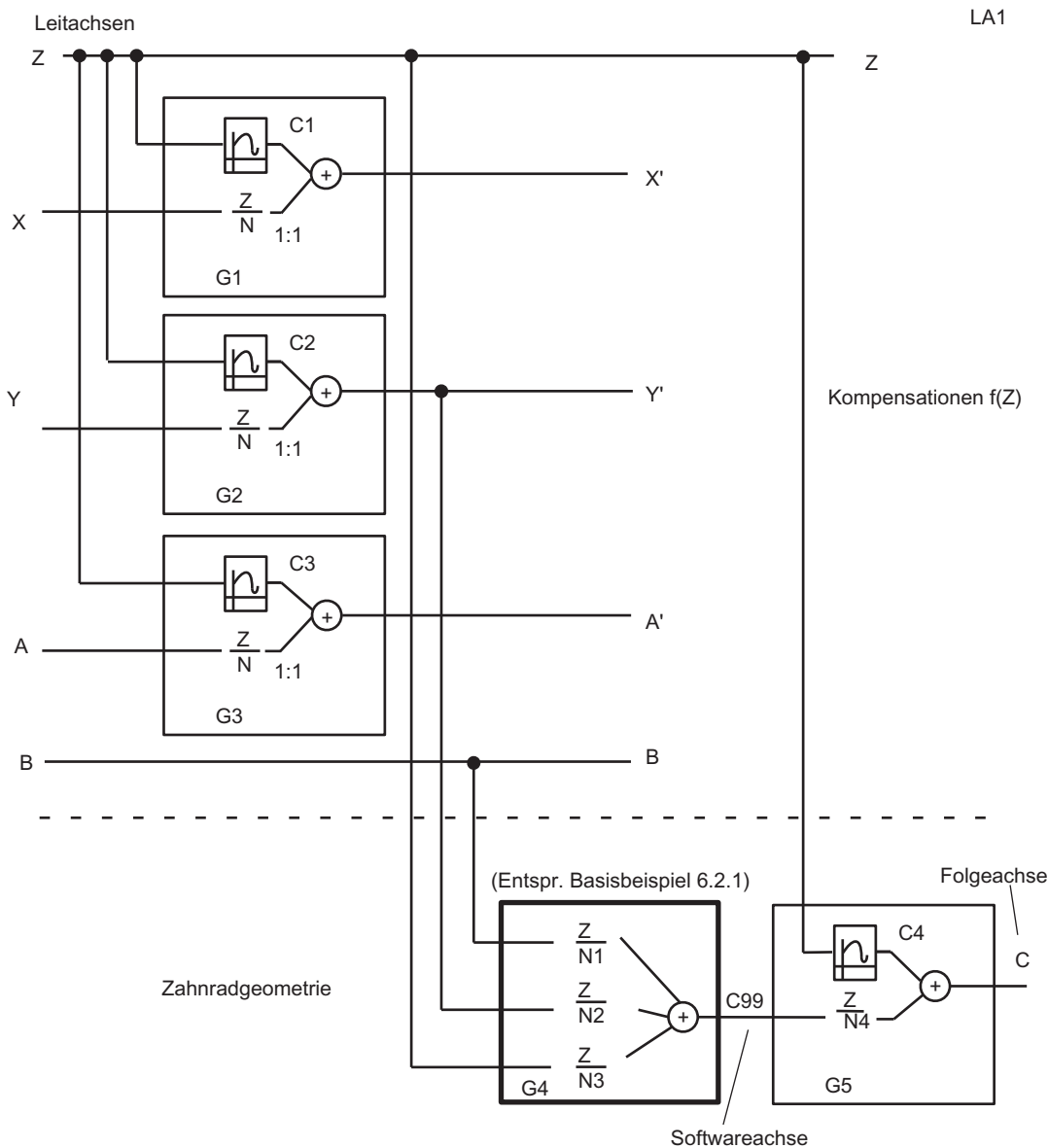


Bild 10-11 Erweitertes Beispiel mit nichtlinearen Maschinenfehlerkompensationen und nicht linearer Komponente an der Zahngeometrie

Der folgende Teileprogramm Ausschnitt ist als allgemeines Konzept zu verstehen, bei dem konkrete Ergänzungen von Kurventabellen und Zahnrad-/Maschinenparametern noch vorgenommen werden müssen. Zu ergänzende Anteile sind durch < ... > notiert. Auch konkret notierte Parameter müssen ggf. geändert werden z. B. Koppelfaktoren.

Programmcode		Kommentar
N100	CTABDEF(X, Z, 1, 0)	; Deklaration und Spezifikation der nicht periodischen Kurventabelle C1
N110	< ... >	; Vorgabe der Kurventabelle: Kurvenpunkte oder Polynomsätze
N190	CTABEND	
N200	CTABDEF(Y, Z, 2, 0)	; Deklaration und Spezifikation einer nicht periodischen Kurventabelle C2
N210	< ... >	; Vorgabe der Kurventabelle: Kurvenpunkte oder Polynomsätze
N290	CTABEND	
N300	CTABDEF(A, Z, 3, 0)	; Deklaration und Spezifikation einer nicht periodischen Kurventabelle C3
N310	< ... >	; Vorgabe der Kurventabelle: Kurvenpunkte oder Polynomsätze
N390	CTABEND	
N400	CTABDEF(C, Z, 4, 0)	; Deklaration und Spezifikation einer nicht periodischen Kurventabelle C4
N410	< ... >	; Vorgabe der Kurventabelle: Kurvenpunkte oder Polynomsätze
N490	CTABEND	
N500	EGDEF(X, Z, 1)	; Deklaration von Pfad über C1, Sollwertkopplung
N510	G1 F1000 X10	; Deklaration des Kommandoanteils von X
N520	EGONSYN(X, "NOC", <SynPosX>, Z, <SynPosX_Z>, 1, 0)	; Einschalten von Pfad über C1
N600	EGDEF(Y, Z, 1)	; Deklaration von Pfad über C2, Sollwertkopplung
N610	G1 F1000 Y10	; Deklaration des Kommandoanteils von Y
N620	EGONSYN(Y, "COARSE", <SynPosY>, Z, <SynPosY_Z>, 2, 0)	; Einschalten von Pfad über C2
N700	EGDEF(A, Z, 1)	; Deklaration von Pfad über C3, Sollwertkopplung
N710	G1 F1000 A10	; Deklaration des Kommandoanteils von A
N720	EGONSYN(A, "FINE", <SynPosA>, Z, <SynPosA_Z>, 3, 0)	; Einschalten von Pfad über C3
		; 1. Getriebestufe, C99 ist die Softwareachse zwischen den zwei Elektronischen Getrieben
N800	EGDEF(C99, Y, 1, Z, 1, B, 1)	

Programmcode	Kommentar
N810	EGONSYN(C99, "NOC", ; Einschalten von Leitachse B <SynPosC99>, B, <SynPosC99_B>, 18, 2, & Y, <SynPosC99_Y>, ; Einschalten von Leitachse Y R1 * π, 1, & Z, <SynPosC99_Z>, ; Einschalten von Leitachse Z 10, 1) ; Zeichen & bedeutet: Befehlsfortsetzung in Folgezeile kein LF und Kommentar im Programm erlaubt ; 2. Getriebestufe
N900	EGDEF(C, C99, 1, Z, ; Deklaration der Folgeachse C99 von Stufe 1 1) als Leitachse von Stufe 2, ; Sollwertkopplung
N910	; Deklaration von Pfad über C4, Sollwertkopp- lung
N920	EGONSYN(C, "NOC", ; Einschalten von Softwareachse C99 <SynPosC>, C99, <SynPosC_C99>, 1, 1, & Z, <SynPosC_Z>, 4, ; und von Leitachse Z über C4 0)
N999	M30

Maschinendaten

Es ist nur ein Ausschnitt angegeben, der über die notwendige Konfiguration von Geometrie-/Kanal- und Maschinenachsparemern hinausgeht.

\$MN_NUM_EG = 5	; Maximale Anzahl von Getrieben
\$MN_MM_NUM_CURVE_TABS = 5	; Maximale Anzahl von Kurventabellen
\$MN_MM_NUM_CURVE_SEGMENTS = 50	;Max. Anzahl von Kurvensegmenten
\$MN_MM_NUM_CURVE_POLYNOMS = 100	;Max. Anzahl von Kurvenpolynomen

Settingdaten

Falls von der beschriebenen Skalierung aus dem Kapitel "Elektronisches Getriebe (EG) (Seite 455)" Gebrauch gemacht wird, ändert sich der Funktionswert aus folgendem Maschinendatum entsprechend der Verschiebung:

MD43108 \$SD_LEAD_SCALE_OUT_POS[4] = 1.2 ; Skalierung für Tabelle C4

Systemvariablen

Aufgrund der oben angegebenen Definitionen werden in den zugeordneten Systemvariablen durch die Steuerung die folgenden Werte eingetragen.

Literatur:

Listenhandbuch Systemvariablen

Die nachfolgend aufgelisteten Systemvariablen sind insgesamt nur **als Kommentar** zu verstehen!

```

; ***** Getriebe X (G1)
$AA_EG_TYPE[X, Z] = 1           ; Sollwertkopplung
$AA_EG_NUMERA[X, Z] = 1        ; Kurventabellen Nr. = 1
$AA_EG_DENOM[X, Z] = 0        ; Nenner = 0 →es gilt Kurventabelle
$P_EG_BC[X] = "NOC"           ; Satzwechselkriterium
$AA_EG_NUM_LA[X] = 1           ; Anzahl der Leitachsen
$AA_EG_AX[0, X] = Z            ; Name der Leitachse
$AA_EG_SYN[X,Z] = <SynPosX_Z> ; Synchronposition der Leitachse Z
$AA_EG_SYNFA[X] = <SynPosX>   ; Synchronposition der Folgeachse
; ***** Getriebe Y (G2)
$AA_EG_TYPE[Y, Z] = 1           ; Sollwertkopplung
$AA_EG_NUMERA[Y, Z] = 2        ; Kurventabellen Nr. = 2
$AA_EG_DENOM[Y, Z] = 0        ; Nenner = 0 →es gilt Kurventabelle
$P_EG_BC[Y10] = "COARSE"      ; Satzwechselkriterium
$AA_EG_NUM_LA[Y] = 1           ; Anzahl der Leitachsen
$AA_EG_AX[0, Y] = Z            ; Name der Leitachse
$AA_EG_SYN[Y, Z] = <SynPosY_Z> ; Synchronposition der Leitachse Z
$AA_EG_SYNFA[Y] = <SynPosY>   ; Synchronposition der Folgeachse
; ***** Getriebe A (G3)
$AA_EG_TYPE[A, Z] = 1           ; Sollwertkopplung
$AA_EG_NUMERA[A, Z] = 3        ; Kurventabellen Nr. = 3
$AA_EG_DENOM[A, Z] = 0        ; Nenner = 0 →es gilt Kurventabelle
$P_EG_BC[A10] = "FINE"        ; Satzwechselkriterium
$AA_EG_NUM_LA[A] = 1           ; Anzahl der Leitachsen
$AA_EG_AX[0, A] = Z            ; Name der Leitachse
$AA_EG_SYN[A, Z] = <SynPosA_Z> ; Synchronposition der Leitachse Z
$AA_EG_SYNFA[A] = <SynPosA>   ; Synchronposition der Folgeachse
; ***** Getriebe C99 (G4)
$AA_EG_TYPE[C99, Y] = 1         ; Sollwertkopplung
$AA_EG_NUMERA[C99, Y] = 18      ; Zähler für Koppelfaktory
$AA_EG_DENOM[C99, Y] = 2        ; Nenner für Koppelfaktory
$AA_EG_TYPE[C99, Z] = 1         ; Sollwertkopplung
$AA_EG_NUMERA[C99, Z] = R1 * π  ; Zähler für Koppelfaktorz
$AA_EG_DENOM[C99, Z] = 1        ; Nenner für Koppelfaktorz
$AA_EG_TYPE[C99, B] = 1         ; Sollwertkopplung
$AA_EG_NUMERA[C99, B] = 10      ; Zähler für Koppelfaktorb
$AA_EG_DENOM[C99, B] = 1        ; Nenner für Koppelfaktorb
$P_EG_BC[C99] = "NOC"          ; Satzwechselkriterium

```

\$AA_EG_NUM_LA[C99] = 3	; Anzahl der Leitachsen
\$AA_EG_AX[0, C99] = Y	; Name der Leitachse Y
\$AA_EG_AX[1, C99] = Z	; Name der Leitachse Z
\$AA_EG_AX[2, C99] = B	; Name der Leitachse B
\$AA_EG_SYN[C99, Y] = <SynPosC99_Y>	; Synchronposition der Leitachse Y
\$AA_EG_SYN[C99, Z] = <SynPosC99_Z>	; Synchronposition der Leitachse Z
\$AA_EG_SYN[C99, B] = <SynPosC99_B>	; Synchronposition der Leitachse B
\$AA_EG_SYNFA[C99] = <SynPosC99>	; Synchronposition der Folgeachse
; ***** Getriebe C (G5)	
\$AA_EG_TYPE[C, Z] = 1	; Sollwertkopplung
\$AA_EG_NUMERA[C, Z] = 4	; Kurventabellen Nr. = 4
\$AA_EG_DENOM[C, Z] = 0	; Nenner = 0 → es gilt Kurventabelle
\$AA_EG_TYPE[C, C99] = 1	; Sollwertkopplung
\$AA_EG_NUMERA[C, C99] = 1	; Zähler für Koppelfaktor _{C99}
\$AA_EG_DENOM[C, C99] = 1	; Nenner für Koppelfaktor _{C99}
\$P_EG_BC[C] = "NOC"	; Satzwechselkriterium
\$AA_EG_NUM_LA[C] = 2	; Anzahl der Leitachsen
\$AA_EG_AX[0, C] = Z	; Name der Leitachse Z
\$AA_EG_AX[1, C] = C99	; Name der Leitachse C99
\$AA_EG_SYN[C, Z] = <SynPosC_Z>	; Synchronposition der Leitachse Z
\$AA_EG_SYN[C, C99] = <SynPosC_C99>	; Synchronposition der Leitachse C99
\$AA_EG_SYNFA[C] = <SynPosC>	; Synchronposition der Leitachse C

Maschinendaten

Auszug aus den MD:

```
; ***** Kanal 1
CHANDATA(1)
; ***** Achse 1, "X"
$MC_AXCONF_GEOAX_NAME_TAB[0] = "X"
$MC_AXCONF_CHANAX_NAME_TAB[0] = "X"
$MC_AXCONF_MACHAX_USED[0] = 1
$MN_AXCONF_MACHAX_NAME_TAB[0] = "X1"
$MA_SPIND_ASSIGN_TO_MACHAX[AX1] = 0
$MA_IS_ROT_AX[AX1] = FALSE
; ***** Achse 2, "Y"
$MC_AXCONF_GEOAX_NAME_TAB[1] = "Y"
$MC_AXCONF_CHANAX_NAME_TAB[1] = "Y"
$MC_AXCONF_MACHAX_USED[1] = 2
$MN_AXCONF_MACHAX_NAME_TAB[1] = "Y1"
```

```
$MA_SPIND_ASSIGN_TO_MACHAX[AX2] = 0
$MA_IS_ROT_AX[AX2] = FALSE
; ***** Achse 3, "Z"
$MC_AXCONF_GEOAX_NAME_TAB[2] = "Z"
$MC_AXCONF_CHANAX_NAME_TAB[2] = "Z"
$MC_AXCONF_MACHAX_USED[2] = 3
$MN_AXCONF_MACHAX_NAME_TAB[2] = "Z1"
$MA_SPIND_ASSIGN_TO_MACHAX[AX3] = 0
$MA_IS_ROT_AX[AX3] = FALSE
; ***** Achse 4, "A"
$MC_AXCONF_CHANAX_NAME_TAB[3] = "A"
$MC_AXCONF_MACHAX_USED[3] = 4
$MN_AXCONF_MACHAX_NAME_TAB[3] = "A1"
$MA_SPIND_ASSIGN_TO_MACHAX[AX4] = 0
$MA_IS_ROT_AX[AX4] = TRUE
$MA_ROT_IS_MODULO[AX4] = TRUE
; ***** Achse 5, "B"
$MC_AXCONF_CHANAX_NAME_TAB[4] = "B"
$MC_AXCONF_MACHAX_USED[4] = 5
$MC_SPIND_DEF_MASTER_SPIND = 1
$MN_AXCONF_MACHAX_NAME_TAB[4] = "B1"
$MA_SPIND_ASSIGN_TO_MACHAX[AX5] = 1
$MA_IS_ROT_AX[AX5] = TRUE
$MA_ROT_IS_MODULO[AX5] = TRUE
; ***** Achse 6, "C"
$MC_AXCONF_CHANAX_NAME_TAB[5] = "C"
$MC_AXCONF_MACHAX_USED[5] = 6
$MN_AXCONF_MACHAX_NAME_TAB[5] = "C1"
$MA_SPIND_ASSIGN_TO_MACHAX[AX6] = 0
$MA_IS_ROT_AX[AX6] = TRUE
$MA_ROT_IS_MODULO[AX6] = TRUE
; ***** Achse 10, "C99"
$MC_AXCONF_CHANAX_NAME_TAB[9] = "C99"
$MC_AXCONF_MACHAX_USED[9] = 10
$MA_SPIND_ASSIGN_TO_MACHAX[AX10] = 0
```

```
$MA_IS_ROT_AX[AX10] = TRUE  
$MA_ROT_IS_MODULO[AX10] = TRUE
```

10.5 Generische Kopplung

10.5.1 Kurzbeschreibung

10.5.1.1 Funktion

Funktion

Die "Generische Kopplung" ist eine allgemeine Kopplungsfunktion, in der alle Kopplungseigenschaften der bestehenden Kopplungsarten (Mitschleppen, Leitwertkopplung, Elektronisches Getriebe und Synchronspindel) zusammengefasst sind.

Die Funktion ermöglicht eine flexible Art der Programmierung:

- Der Anwender kann die für seine Applikation notwendigen Kopplungseigenschaften frei auswählen (Baukastenprinzip).
- Jede Kopplungseigenschaft ist einzeln programmierbar.
- Die Kopplungseigenschaften einer definierten Kopplung (z. B. Koppelfaktor) sind änderbar.
- Eine spätere Nutzung weiterer Kopplungseigenschaften ist möglich.
- Das Koordinatenbezugssystem der Folgeachse (Basiskoordinatensystem oder Maschinenkoordinatensystem) ist programmierbar.
- Bestimmte Kopplungseigenschaften können auch in Synchronaktionen programmiert werden.

Literatur:

Funktionshandbuch Synchronaktionen

Anpasszyklen

Die bisherigen Kopplungsaufrufe für das Mitschleppen (TRAIL*), Leitwertkopplung (LEAD*), Elektronisches Getriebe (EG*) und Synchronspindel (COUP*) werden über Anpasszyklen weiterhin unterstützt (siehe Kapitel "Anpasszyklen (Seite 534)").

10.5.1.2 Voraussetzungen

CP-Ausprägung

Die Generische Kopplung gibt es in einer Grundauführung und vier optionalen Ausprägungen:

- CP-STATIC
- CP-BASIC
- CP-COMFORT
- CP-EXPERT

Dieser Aufteilung liegen folgende Überlegungen zugrunde:

- Von der Grundauführung bis zur optionalen Ausführung CP-EXPERT nehmen der Funktionsumfang und das benötigte Applikationswissen zu.
- Entscheidend für die Wahl der Ausführung ist die gleichzeitig benötigte Anzahl von Kopplungen (Folgeachsen/-spindeln) und deren Eigenschaften.
 - Beispiel für Gleichzeitigkeit:
Wird hintereinander 1 x Synchronspindelpaar für die Teileübergabe von der Haupt- zur Gegenspindel und anschließend 1 x Mehrkantdrehen benötigt, ist die optionale Ausführung CP-BASIC geeignet und ausreichend. Wenn nicht ausgeschlossen werden soll, dass sich die beiden Abläufe auch zeitlich überlappen (Mehrkantdrehen läuft noch während Teileübergabe begonnen wird), ist dagegen die optionale Ausführung CP-COMFORT erforderlich.
 - Beispiel für Eigenschaft:
Wird ein Mitschleppverband mit einer Leitachse benötigt, ist die Grundauführung ausreichend. Für Mitschleppverbände mit zwei Leitachsen ist eine der optionalen Ausführungen erforderlich.
- Die einzelnen Ausführungen sind unabhängig voneinander. Sie können miteinander kombiniert werden und sind gleichzeitig aktivierbar.

Tabelle 10-2 Mengengerüst der abhängig von der Ausführung gleichzeitig aktivierbaren Koppelmodule

Typ	CP-Ausführungen erlauben ein oder mehrere unterschiedliche CPSETTYPE-Koppelobjekte gleichzeitig:	Grund-aus-führung	CP-STATIC	CP-BASIC	CP-COMFORT	CP-EXPERT
A	Mitschleppen	4	-	4	4	8
B	Synchronspindel mit 1 :1-Kopplung	-	1	-	-	-
C	o./u. Synchronspindel / Mehrkantdrehen o./u. Leitwertkopplung / Kurventabelleninterpolation o./u. MKS-Kopplung	-	-	1	4	8
D	o./u. Elektronisches Getriebe "einfach" o./u. Freie Generische Kopplung "einfach"	-	-	-	1	8
E	o./u. Elektronisches Getriebe o./u. Freie Generische Kopplung	-	-	-	-	5

o./u. steht für oder/und

Tabelle 10-3 Skalierung in der Verfügbarkeit von Kopplungseigenschaften

Typ A	Typ B	Typ C	Typ D	Typ E	
20	1	13	9	5	Maximale Anzahl an CPSETTYPE-bezogenen Funktionalitäten (pro Typ)
TRAIL - Mitschleppen					
20	-	13	9	5	Maximale Anzahl Mitschleppverbände mit folgenden Eigenschaften: → siehe CPSETTYPE="TRAIL" ¹⁾
1		2	2	2	Maximale Anzahl Leitwerte
-		-	+	+	Kopplung zwischen einer Spindel und einer Achse
+		+	+	+	Kopplung zwischen einer Rundachse und einer Linearachse
-		+	+	+	Aus Teileprogramm und Synchronaktionen
+		+	+	+	Überlagerung / Differenzdrehzahl erlaubt
-		-	-	+	Kaskadierung erlaubt
BCS		BCS / MCS	BCS / MCS	BCS / MCS	Koordinatenbezug (Standard: CPFRS="BCS")
Synchronspindel mit 1 :1-Kopplung					
-	1	-	-	-	Maximale Anzahl Synchronspindel / Mehrkantdrehen mit folgenden Eigenschaften: → siehe CPSETTYPE="COUP" ¹⁾
	1				Maximale Anzahl Leitwerte
	-				Aus Teileprogramm und Synchronaktionen
	-				Überlagerung / Differenzdrehzahl erlaubt
	-				Kaskadierung erlaubt
	MCS				Koordinatenbezug fix (CPFRS="MCS")
COUP - Synchronspindel / Mehrkantdrehen					
-	-	13	9	5	Maximale Anzahl Synchronspindel / Mehrkantdrehen mit folgenden Eigenschaften: → siehe CPSETTYPE="COUP" ¹⁾
		1	1	1	Maximale Anzahl Leitwerte
		-	-	-	Aus Teileprogramm und Synchronaktionen
		+	+	+	Überlagerung / Differenzdrehzahl erlaubt
		-	-	-	Kaskadierung erlaubt
		MCS	MCS	MCS	Koordinatenbezug fix (CPFRS="MCS")
LEAD - Leitwertkopplung / Kurventabelleninterpolation					
-	-	13	9	5	Maximale Anzahl Leitwertkopplung / Kurventabelleninterpolation mit folgenden Eigenschaften: → siehe CPSETTYPE="LEAD" ¹⁾
		1	1	1	Maximale Anzahl Leitwerte
		+	+	+	Aus Teileprogramm und Synchronaktionen
		+	+	+	Überlagerung / Differenzdrehzahl erlaubt
		-	-	+	Kaskadierung erlaubt

Typ A	Typ B	Typ C	Typ D	Typ E	
		BCS / MCS	BCS / MCS	BCS / MCS	Koordinatenbezug (Standard: CPFRS="BCS")
EG - Elektronisches Getriebe					
-	-	-	9	5	Maximale Anzahl Elektronisches Getriebe mit folgenden Eigenschaften: → siehe CPSETTYPE="EG" ¹⁾
			3	5	Maximale Anzahl Leitwerte
			-	-	Aus Teileprogramm und Synchronaktionen
			+	+	Überlagerung / Differenzdrehzahl erlaubt
			-	+	Kaskadierung erlaubt
			BCS / MCS	BCS / MCS	Koordinatenbezug (Standard: CPFRS="BCS")
			-	+	Nichtlineares Koppelgesetz (CPLCTID) erlaubt
CP - Freie Generische Kopplung					
-	-	-	9	5	Maximale Anzahl Freie Generische Kopplung mit folgenden Eigenschaften: Standard (entspricht CPSETTYPE="CP" ¹⁾)
			3	5	Maximale Anzahl Leitwerte
			+	+	Aus Teileprogramm und Synchronaktionen
			+	+	Überlagerung / Differenzdrehzahl erlaubt
			-	+	Kaskadierung erlaubt
			BCS / MCS	BCS / MCS	Koordinatenbezug (Standard: CPFRS="BCS")
			-	+	Nichtlineares Koppelgesetz (CPLCTID) erlaubt
¹⁾ Siehe " Kopplungstypen (CPSETTYPE) (Seite 536) "					

Hinweis

Bestehende Kopplungsoptionen (Leitwertkopplung, Elektronisches Getriebe und Synchronspindel) werden von der Generischen Kopplung nicht berücksichtigt. Der gleichzeitige Einsatz von bestehenden Kopplungsoptionen und der Generischen Kopplung ist nur möglich, wenn sich die Kopplungen auf verschiedene Achsen/Spindeln beziehen.

Speicherkonfiguration

Der Speicherplatz, der im dynamischen NC-Speicher für die Generische Kopplung reserviert werden soll, wird definiert mit den Maschinendaten:

MD18450 \$MN_MM_NUM_CP_MODULES (Maximal zulässige Anzahl der CP-Koppelmodule)

MD18452 \$MN_MM_NUM_CP_MODUL_LEAD (Maximal zulässige Anzahl der CP-Leitwerte)

Hinweis

Empfehlung: Bereits bei der Inbetriebnahme die zu erwartenden Maximalwerte einstellen, die an dieser Maschine im Maximalausbau jemals gleichzeitig benötigt werden.

Hardware-Voraussetzungen

Die Nutzung der Option "CP-EXPERT" bedingt den Einsatz von Systemen mit mehr als 6 Achsen.

10.5.2 Grundlagen

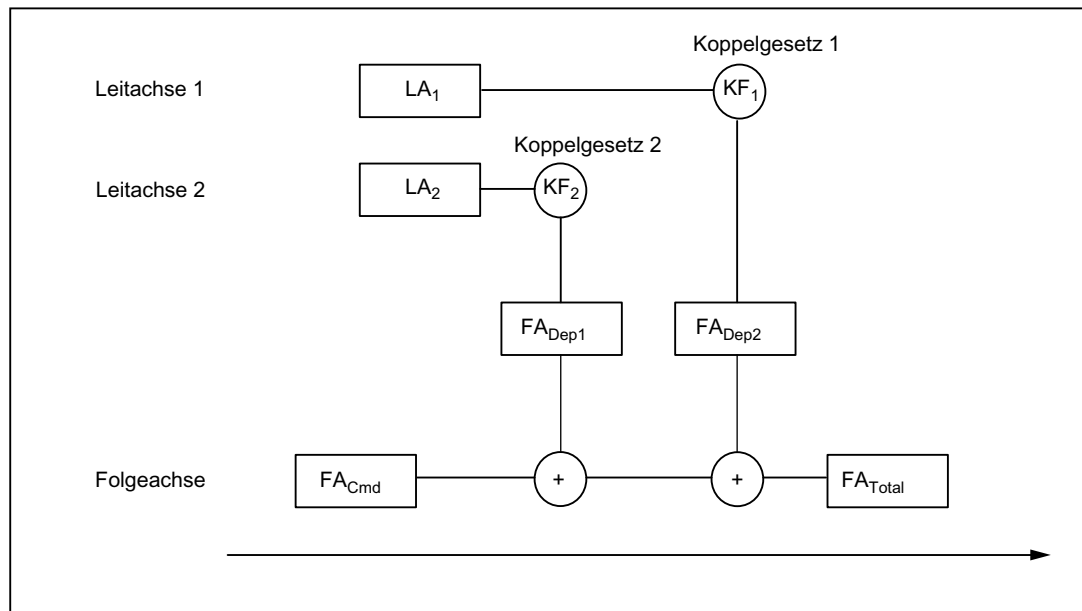
10.5.2.1 Koppelmodul

Mit Hilfe eines Koppelmoduls kann die Bewegung einer Achse (→ Folgeachse) abhängig von anderen Achsen (→ Leitachsen) interpoliert werden.

Koppelgesetz

Die Zusammenhänge zwischen den Leitachsen/-werten und der Folgeachse sind je Leitachse/-wert durch ein Koppelgesetz (Koppelfaktor oder Kurventabelle) definiert. Die einzelnen Bewegungsanteile aus den einzelnen Leitachsen/-werten wirken additiv.

Der Zusammenhang soll anhand des folgenden Beispiels (Folgeachse mit zwei Leitachsen) veranschaulicht werden:



- FA_{Total} Gesamtsollwert der Folgeachse
- FA_{Cmd} Im Teileprogramm gesetzter Sollwert
= Unabhängiger Bewegungsanteil der Folgeachse
- FA_{DEP1} Abhängiger Bewegungsanteil von der Leitachse 1
- FA_{DEP2} Abhängiger Bewegungsanteil von der Leitachse 2
- LA₁ Soll- oder Istwert der 1-ten Leitachse
- LA₂ Soll- oder Istwert der 2-ten Leitachse
- SynPosLA₁ Synchronposition der 1-ten Leitachse
- SynPosLA₂ Synchronposition der 2-ten Leitachse
- KF₁ Koppelfaktor der 1-ten Leitachse
- KF₂ Koppelfaktor der 2-ten Leitachse

Die Folgeachsposition ergibt sich aus Überlagerung (Summation) aus den abhängigen Bewegungsanteilen (FA_{DEP1} und FA_{DEP2}), die sich aus den einzelnen Koppelbeziehungen zu den Leitachsen ergeben, und aus dem unabhängigen Bewegungsanteil (FA_{Cmd}) der Folgeachse:

$$FA_{Total} = FA_{Cmd} + FA_{DEP1} + FA_{DEP2}$$

Die Bewegungsanteile der Folgeachse ergeben sich wie folgt:

- Abhängiger Bewegungsanteil von der Leitachse 1: $FA_{DEP1} = (LA_1 - SynPosLA_1) * KF_1$
- Abhängiger Bewegungsanteil von der Leitachse 2: $FA_{DEP2} = (LA_2 - SynPosLA_2) * KF_2$
- Unabhängiger Bewegungsanteil der Folgeachse: FA_{Cmd}

Folgeachsüberlagerung

Die Überlagerung der abhängigen und unabhängigen Bewegungsanteile der Folgeachse wird Folgeachsüberlagerung genannt.

Der unabhängige Bewegungsanteil der Folgeachse kann mit allen zur Verfügung stehenden Bewegungsbefehlen programmiert werden.

10.5.2.2 Schlüsselwörter und Kopplungseigenschaften

Schlüsselwörter

Die Programmierung bei den bestehenden Achskopplungen erfolgt über Sprachbefehle, z. B. Mitschleppen mit `TRAILON (X, Y, 2)`. Bei der Generischen Kopplung werden die Sprachbefehle durch Schlüsselwörter ersetzt.

Dies hat folgende Vorteile:

- Kopplungseigenschaften können einzeln programmiert werden (siehe nachfolgendes Beispiel).
- Programmierungen für mehrere Kopplungen können in einem Satz erfolgen (da Schlüsselwörter keinen eigenen Satz benötigen).
Vorteil: Reduktion der Abarbeitungszeit

Beispiel:

Die mit dem bestehenden Kopplungsaufwurf `TRAILON (X, Y, 2)` gesetzten Eigenschaften (Folgeachse, Leitachse und Koppelfaktor) werden in der Generischen Kopplung durch die folgenden Schlüsselwörter definiert:

`CPON= (X1) CPLA [X1] = (X2) CPLNUM [X1, X2] =2`

<code>CPON= (X1)</code>	Kopplung zur Folgeachse X1 einschalten.
<code>CPLA [X1] = (X2)</code>	Achse X2 als Leitachse definieren.
<code>CPLNUM [X1, X2] =2</code>	Zähler des Koppelfaktors auf 2 setzen.

Schreibweise

Zur eindeutigen Zuordnung sind die Schlüsselwörter mit dem Präfix "CP" gekennzeichnet (für Coupling). Je nach Bedeutung und Einsatzort wird ein dritter Buchstabe verwendet:

Schlüsselwort-Präfix	Bedeutung	Beispiel
CP*	Beschreibt eine Eigenschaft der gesamten Kopplung.	CPON ¹⁾
CPF*	Beschreibt eine Eigenschaft der Folgeachse (Following axis).	CPFPOS ¹⁾
CPL*	Beschreibt eine Eigenschaft bezogen auf die Leitachse (Leading axis) bzw. das Kopplungsgesetz.	CPLON ¹⁾ , CPLNUM ¹⁾
CPM*	Beschreibt eine Eigenschaft der gesamten Kopplung bei besonderen Zuständen.	CPMRESET ¹⁾

¹⁾ Schlüsselwort-Bedeutung siehe folgende Tabelle "Übersicht aller Schlüsselwörter und Kopplungseigenschaften".

Übersicht aller Schlüsselwörter und Kopplungseigenschaften

Die folgende Tabelle stellt eine Übersicht aller Schlüsselwörter der Generischen Kopplung und der damit programmierbaren Kopplungseigenschaften dar:

Schlüsselwort	Kopplungseigenschaft / Bedeutung	Standardeinstellung (CPSETTYPE="CP")
CPDEF	Anlegen eines Koppelmoduls	
CPDEL	Löschen eines Koppelmoduls	
CPLDEF	Definition einer Leitachse und Anlegen eines Koppelmoduls	
CPLDEL	Löschen einer Leitachse eines Koppelmoduls	
CPON	Einschalten eines Koppelmoduls	
CPOF	Ausschalten eines Koppelmoduls	
CPLON	Einschalten einer Leitachse eines Koppelmoduls	
CPLOF	Ausschalten einer Leitachse eines Koppelmoduls	
CPLNUM	Zähler des Koppelfaktors	1.0
CPLDEN	Nenner des Koppelfaktors	1.0
CPLCTID	Nummer der Kurventabelle	nicht gesetzt
CPLSETVAL	Kopplungsbezug	CMDPOS
CPFRS	Koordinatenbezugssystem	BCS
CPBC	Satzwechselkriterium	NOC
CPFPOS + CPON	Synchronposition der Folgeachse beim Einschalten	nicht gesetzt
CPLPOS + CPON	Synchronposition der Leitachse beim Einschalten	nicht gesetzt
CPFMSON	Synchronisationsmodus	CFAST
CPFMON	Verhalten der Folgeachse beim Einschalten	STOP
CPFMOF	Verhalten der Folgeachse beim vollständigen Ausschalten	STOP
CPFPOS + CPOF	Ausschaltposition der Folgeachse beim Ausschalten	nicht gesetzt
CPMRESET	Kopplungsverhalten bei RESET	NONE
CPMSTART	Kopplungsverhalten beim Teileprogrammstart	NONE
CPMPRT	Kopplungsverhalten beim Teileprogrammstart unter Satzsuchlauf via Programmtest	NONE
CPLINTR	Verschiebewert für den Eingangswert einer Leitachse	0.0
CPLINSC	Skalierfaktor für den Eingangswert einer Leitachse	1.0
CPLOUTTR	Verschiebewert für den Ausgangswert einer Kopplung	0.0
CPLOUTSC	Skalierfaktor für den Ausgangswert einer Kopplung	1.0

Schlüsselwort	Kopplungseigenschaft / Bedeutung	Standardeinstellung (CPSETTYPE="CP")
CPSYNCOV	Schwellwert für den Positionssynchronlauf "Grob"	MD37200
CPSYNFIP	Schwellwert für den Positionssynchronlauf "Fein"	MD37210
CPSYNCOV2	Zweiter Schwellwert für den Positionssynchronlauf "Grob"	MD37202
CPSYNFIP2	Zweiter Schwellwert für den Positionssynchronlauf "Fein"	MD37212
CPSYNCOV	Schwellwert für den Geschwindigkeitssynchronlauf "Grob"	MD37220
CPSYNFIV	Schwellwert für den Geschwindigkeitssynchronlauf "Fein"	MD37230
CPMBRAKE	Verhalten der Folgeachse bei bestimmten Stoppsignalen und -kommandos	1
CPMVDI	Verhalten der Folgeachse auf bestimmte NC/PLC-Nahtstellensignale	0 (für Bit 0 ... 3, 5) 1 (für Bit 4, 6)
CPMALARM	Unterdrückung spezieller kopplungsbezogener Alarmausgaben	MD11410 MD11415
CPSETTYPE	Kopplungstyp	CP

Hinweis

Kopplungseigenschaften, die nicht explizit programmiert werden (im Teileprogramm oder in Synchronaktionen), werden mit ihren Standardeinstellungen wirksam (siehe rechte Spalte in der Tabelle).

Abhängig von der Einstellung des Schlüsselworts `CPSETTYPE` können statt der Standardeinstellungen (`CPSETTYPE="CP"`) auch voreingestellte Kopplungseigenschaften wirksam werden (siehe Kapitel "Kopplungstypen (Seite 536)").

10.5.2.3 Systemvariablen

Der aktuelle Zustand einer mittels Schlüsselwort gesetzten Kopplungseigenschaft kann durch die entsprechende Systemvariable gelesen und beschrieben werden.

Hinweis

Beim Schreiben im Teileprogramm wird Vorlaufstopp erzeugt.

Schreibweise

Die Namen der Systemvariablen ergeben sich i. d. R. aus den Namen der entsprechenden Schlüsselwörter und dem Voranstellen eines entsprechenden Präfixes.

Der 1. Buchstabe des Präfixes beschreibt den Zugriffsort beim Lesen:

Systemvariablen-Präfix	Zugriffsort beim Lesen	Merkmale
\$PA_CP	Lesen von kanalbezogenen achs-spezifischen Kopplungseigenschaften in der Satzaufbereitung (Preparation)	Verwendung in Synchronaktionen ist nicht möglich. Erzeugt keinen impliziten Vorlaufstopp.
\$AA_CP	Lesen des aktuellen Zustandes des Koppelmoduls (kanalübergreifend).	Verwendung im Teileprogramm und in Synchronaktionen möglich. Erzeugt bei Verwendung im Teileprogramm einen impliziten Vorlaufstopp.

Hinweis

Der Vorlaufwert einer Systemvariablen \$PA_CP... unterscheidet sich nur bei aktiver Teileprogrammabarbeitung von den Werten der entsprechenden Systemvariablen \$AA_CP....

Bei Programmende bzw. Abbruch erfolgt eine entsprechende Synchronisation des Vorlaufs auf die Hauptlaufzustände.

Systemvariablen-Liste

Eine Liste aller in der Generischen Kopplung verwendbaren Systemvariablen befindet sich in den Datenlisten (siehe Kapitel "Systemvariablen (Seite 568)").

Eine ausführliche Beschreibung der Systemvariablen findet sich in:

Literatur:

Listenhandbuch Systemvariablen

10.5.3 Koppelmodule anlegen/löschen

10.5.3.1 Koppelmodul anlegen (CPDEF)

Ein axiales Koppelmodul wird durch die Definition der Folgeachse angelegt.

Programmierung

Syntax: CPDEF= (<Folgeachse/-spindel>)

Bezeichnung: **Coupling Definition**

Funktionalität: Definition eines Koppelmoduls. Die Kopplung wird nicht aktiviert.

Folgeachse/
-spindel: Typ: **AXIS**

Wertebereich: Alle im Kanal definierten Achs- und Spindelname

Beispiel:

Programmierung	Kommentar
CPDEF=(X2)	; Ein Koppelmodul wird angelegt mit der Achse X2 als Folgeachse.

Randbedingungen

- Die maximale Anzahl der Koppelmodule ist begrenzt (siehe Kapitel "Voraussetzungen (Seite 484)").
- Die Anwendung von CPDEF auf ein bereits angelegtes Koppelmodul ist möglich und wird nicht mit einem Alarm quittiert.

10.5.3.2 Koppelmodul löschen (CPDEL)

Ein mit CPDEF angelegtes Koppelmodul kann mit CPDEL wieder gelöscht werden.

Programmierung

Syntax: CPDEL= (<Folgeachse/-spindel>)

Bezeichnung: **Coupling Delete**

Funktionalität: Löschen eines Koppelmoduls. Mit dem Koppelmodul werden alle Leitachsmodule gelöscht und der reservierte Speicher wieder freigegeben.

Folgeachse/
-spindel: Typ: AXIS

Wertebereich: Alle im Kanal definierten Achs- und Spindelnamen

Beispiel:

Programmierung	Kommentar
CPDEL=(X2)	; Löschen des Koppelmoduls mit der Folgeachse X2.

Randbedingungen

- Der Schaltbefehl CPDEL löst bei aktiver Kopplung einen Vorlaufstopp aus. Ausnahme: bei CPSETTYPE="COUP" erfolgt kein Vorlaufstopp.
- Die Anwendung von CPDEL auf ein in der Satzaufbereitung aktives Koppelmodul bewirkt ein implizites Deaktivieren dieser Kopplung.
- Die Anwendung von CPDEL auf ein nicht definiertes Koppelmodul ist ohne Wirkung.

10.5.3.3 Leitachsen definieren (CPLDEF bzw. CPDEF+CPLA)

Die Leitachsen/-spindeln für ein definiertes Koppelmodul können mit dem Schlüsselwort CPLDEF oder mit dem Schlüsselwort CPLA in Verbindung mit CPDEF programmiert/angelegt werden.

Programmierung mit CPLDEF

Syntax: `CPLDEF [FAx] = (<Leitachse/-spindel>)`

Bezeichnung: **Coupling Lead Axis Definition**

Funktionalität: Definition einer Leitachse/-spindel zur Folgeachse/-spindel FAx. Im Koppelmodul wird ein Leitachs/-spindel-Modul angelegt. Wenn das Koppelmodul der Folgeachse/-spindel FAx noch nicht angelegt ist, wird das Koppelmodul implizit angelegt.

Leitachse/-spindel: Typ: **AXIS**

Wertebereich: **Alle im Kanal definierten Achs- und Spindelnamen**

Beispiel:

Programmierung	Kommentar
<code>CPLDEF[X2]=(X1)</code>	<code>; Definition der Leitachse X1 zur Folgeachse X2.</code>

Programmierung mit CPLA und CPDEF

Syntax: `CPLA [FAx] = (<Leitachse/-spindel>)`

Bezeichnung: **Coupling Lead Axis**

Funktionalität: Definition einer Leitachse/-spindel zur Folgeachse/-spindel FAx..

Leitachse/-spindel: Typ: **AXIS**

Wertebereich: Alle im Kanal definierten Achs- und Spindelnamen

Beispiel:

Programmierung	Kommentar
CPDEF=(X2) CPLA[X2]=(X1)	; Definition der Leitachse X1 zur Folgeachse X2.

Randbedingungen

- CPLDEF ist nur in Sätzen ohne CPDEF/CPON/CPOF/CPDEL erlaubt.
(Diese Einschränkung gilt nur für den Fall, dass sich die Schlüsselwörter auf das gleiche Koppelmodul beziehen.)
- Die maximale Anzahl der Leitachsmodul pro Koppelmodul ist begrenzt (siehe Kapitel "Voraussetzungen (Seite 484)").
- Definieren von Leitachsen auf ein bereits definiertes oder aktives Koppelmodul ist möglich. Die neu definierten Leitachsen und deren Eigenschaften (z. B. Koppelfaktor) werden nicht sofort aktiv. Zur Aktivierung ist ein entsprechender Einschaltbefehl (CPON oder CPLON) notwendig.

10.5.3.4 Leitachsen löschen (CPLDEL bzw. CPDEL+CPLA)

Definierte Leitachsen können mit CPLDEL oder mit CPLA in Verbindung mit CPDEL wieder gelöscht werden, d. h. aus dem Koppelmodul entfernt werden.

Programmierung mit CPLDEL

Syntax: CPLDEL [FAx] = (<Leitachse/-spindel>)

Bezeichnung: **Coupling Lead Axis Delete**

Funktionalität: Löschen einer Leitachse/-spindel der Kopplung zur Folgeachse/-spindel FAx. Das Leitachs/-spindel-Modul wird gelöscht und der entsprechende Speicher freigegeben. Besitzt das Koppelmodul keine Leitachse/-spindel mehr, so wird auch das Koppelmodul gelöscht und der Speicher freigegeben.

Leitachse/
-spindel: Typ: AXIS

Wertebereich: Alle im Kanal definierten Achs- und Spindelnamen

Beispiel:

Programmierung	Kommentar
CPLDEL[X2]=(X1)	; Löschen der Leitachse X1 der Kopplung zur Folgeachse X2.

Programmierung mit CPLA und CPDEL

Syntax: `CPLA [FAx] = (<Leitachse/-spindel>)`

Bezeichnung: **Coupling Lead Axis**

Funktionalität: Löschen einer Leitachse/-spindel. Das Leitachs/-spindel-Modul wird gelöscht und der entsprechende Speicher freigegeben. Besitzt das Koppelmodul keine Leitachse/-spindel mehr, so wird auch das Koppelmodul gelöscht und der Speicher freigegeben.

Leitachse/-spindel: Typ: **AXIS**

Wertebereich: **Alle im Kanal definierten Achs- und Spindelnamen**

Beispiel:

Programmierung	Kommentar
CPDEL=(X2) CPLA[X2]=(X1)	; Löschen der Leitachse X1 der Kopplung zur Folgeachse X2.

Randbedingungen

- CPLDEL ist nur in Sätzen ohne CPDEF/CPON/CPOF/CPDEL erlaubt. (Diese Einschränkung gilt nur für den Fall, dass sich die Schlüsselwörter auf das gleiche Koppelmodul beziehen.)
- Wird eine aktive Leitachse gelöscht, so wird die Kopplung zu dieser Leitachse implizit deaktiviert.
- Mit dem Löschen der letzten Leitachse wird das gesamte Koppelmodul gelöscht.

10.5.4 Kopplung ein-/ausschalten

10.5.4.1 Koppelmodul einschalten (CPON)

Mit dem Schaltbefehl CPON wird ein definiertes Koppelmodul aktiviert.

Zusammen mit dem Einschaltbefehl können Kopplungseigenschaften wie z. B. Koppelbezug und Koppelgesetz über eigene Schlüsselwörter programmiert werden (siehe Kapitel "Kopplungseigenschaften programmieren (Seite 500)").

Ohne Programmierung wird ein Mitschleppverband bzw. Synchronspindelpaar auf Basis einer Sollwertkopplung (Standardeinstellung für `CPLSETVAL`) mit dem Koppelgesetz 1:1 (Standardeinstellung für `CPLNUM/CPLDEN`) wirksam.

Programmierung

Syntax:	<code>CPON= (<Folgeachse/-spindel>)</code>
Bezeichnung:	Coupling On
Funktionalität:	Aktiviert die Kopplung der Folgeachse zu allen definierten Leitachsen.
Folgeachse/ -spindel:	Typ: <code>AXIS</code>
Wertebereich:	Alle im Kanal definierten Achs- und Spindelnamen

Beispiel:

Programmierung	Kommentar
<code>CPON= (X2)</code>	; Die Kopplung der Folgeachse X2 wird aktiviert.

Randbedingungen

Die Anwendung von `CPON` auf eine bereits aktivierte Kopplung bewirkt eine Neusynchronisation. Damit werden auch gegebenenfalls geänderte Kopplungseigenschaften wirksam. Eine eventuell verloren gegangene Synchronisation (Folgeachse war z. B. im Nachführen) wird wieder hergestellt.

10.5.4.2 Koppelmodul ausschalten (CPOF)

Eine aktivierte Kopplung kann mit dem Schaltbefehl `CPOF` deaktiviert werden. Das Deaktivieren, d. h. das Aufheben der Kopplung zu den Leitachsen, erfolgt entsprechend der gesetzten Ausschalteigenschaft (siehe `CPFMOF`).

Programmierung

Syntax:	<code>CPOF= (<Folgeachse/-spindel>)</code>
Bezeichnung:	Coupling Off
Funktionalität:	Deaktiviert die Kopplung der Folgeachse zu allen definierten Leitachsen.

Folgeachse/
-spindel: Typ: AXIS

Wertebereich: Alle im Kanal definierten Achs- und Spindelnamen

Beispiel:

Programmierung	Kommentar
CPOF=(X2)	; Die Kopplung der Folgeachse X2 wird deaktiviert.

Randbedingungen

- Der Schaltbefehl CPOF löst bei aktiver Kopplung einen Vorlaufstopp aus.
Ausnahme: bei CPSETTYPE="COUP" erfolgt kein Vorlaufstopp:
- Ein CPOF-Schaltbefehl auf ein bereits deaktiviertes bzw. gelöscht Koppelmodul ist ohne Wirkung.
- CPOF ist auch in Synchronaktionen programmierbar.

10.5.4.3 Leitachsen eines Koppelmoduls einschalten (CPLON)

CPLON aktiviert die Kopplung einer Leitachse zu einer Folgeachse. Sind mehrere Leitachsen für ein Koppelmodul definiert, werden diese separat mit CPLON aktiviert.

Programmierung

Syntax: CPLON [FAx] = <Leitachse/-spindel>

Bezeichnung: **Coupling Lead Axis On**

Funktionalität: Aktiviert die Kopplung einer Leitachse/-spindel zur Folgeachse/-spindel FAx.

Leitachse/
-spindel: Typ: AXIS

Wertebereich: Alle im Kanal definierten Achs- und Spindelnamen

Beispiel:

Programmierung	Kommentar
CPLON[X2]=(X1)	; Die Kopplung der Leitachse X1 zur Folgeachse X2 wird aktiviert.

Randbedingungen

CPLON ist auch in Synchronaktionen programmierbar.

10.5.4.4 Leitachsen eines Koppelmoduls ausschalten (CPLOF)

CPLOF deaktiviert die Kopplung einer Leitachse zu einer Folgeachse. Sind mehrere Leitachsen für ein Koppelmodul definiert, werden diese separat mit CPLOF deaktiviert.

Programmierung

Syntax: CPLOF [FAx] = <Leitachse/-spindel>

Bezeichnung: **Coupling Lead Axis Off**

Funktionalität: Deaktiviert die Kopplung einer Leitachse/-spindel zur Folgeachse/-spindel FAx.

Leitachse/
-spindel: Typ: AXIS

Wertebereich: Achsen des Kanals

Beispiel:

Programmierung	Kommentar
CPLOF[X2]=(X1)	; Die Kopplung der Leitachse X1 zur Folgeachse X2 wird deaktiviert.

Randbedingungen

CPLOF ist auch in Synchronaktionen programmierbar.

10.5.4.5 Implizites Anlegen und Löschen von Koppelmodulen

Einschaltkommandos können auch zum Anlegen von Koppelmodulen benutzt werden (ohne vorherige Definition mit CPDEF).

Beispiel

Programmierung	Kommentar
CPON=(X2) CPLA[X2]=(X1) oder CPLON[X2]=(X1) ...	; Legt ein Koppelmodul für die Folgeachse X2 mit der Leitachse X1 an und aktiviert das Koppelmodul.

Programmierung	Kommentar
CPOF=(X2)	; Das implizit angelegte Koppelmodul wird nach dessen Deaktivierung wieder gelöscht.

Randbedingungen

- Implizit (über Einschaltkommandos) angelegte Koppelmodule werden mit ihrer vollständigen Deaktivierung (CPOF) auch gleichzeitig gelöscht.
Vorteil: Das Löschen mit CPDEL/CPLDEL ist überflüssig.
Nachteil (evtl.): Alle gesetzten Kopplungseigenschaften gehen bei CPOF verloren.
- Implizit angelegte Koppelmodule können durch eine nachfolgende Anweisung CPDEF/CPLDEF in explizite Koppelmodule umgewandelt werden. Dadurch wird das Koppelmodul bei CPOF nicht gelöscht und die Daten bleiben erhalten.

10.5.5 Kopplungseigenschaften programmieren

10.5.5.1 Koppelgesetz (CPLNUM, CPLDEN, CPLCTID)

Der Funktionszusammenhang zwischen Leitwert und Folgewert wird je Leitachse/-spindel durch ein Koppelgesetz festgelegt. Dieser funktionale Zusammenhang kann linear durch einen Koppelfaktor oder nichtlinear durch eine Kurventabelle definiert werden. Die damit berechneten Folgewertanteile aus den einzelnen Leitwerten wirken additiv.

Programmierung: Koppelfaktor

Mit der Programmierung eines Koppelfaktors wird ein zuvor aktiviertes nichtlineares Koppelverhältnis (Kurventabelle) deaktiviert.

Der Koppelfaktor wird durch Zähler und Nenner programmiert.

Im Standardzustand, d. h. ohne explizite Programmierung nach Neuanlage eines Koppelmoduls, wird der Zähler und Nenner jeweils mit 1 vorbelegt.

Wird nur der Zähler programmiert, wirkt dieser als Faktor, da der Nenner 1 ist.

Genauere, lineare Verhältnisse können durch die Programmierung von Zähler und Nenner definiert werden.

Zähler des Koppelfaktors

Syntax: CPLNUM[FAx, LAx] = <Wert>

Bezeichnung: Coupling Lead Numerator

Funktionalität: Definiert den Zähler des Koppelfaktors für das Koppelgesetz der Folgeachse/-spindel FAx zur Leitachse/-spindel LAx.

Wert: Typ: REAL
Wertebereich: -2^{31} bis $+2^{31}$
Standardwert: +1.0

Beispiel:

Programmierung	Kommentar
CPLNUM[X2,X1]=1.3	; Der Zähler des Koppelfaktors der Kopplung der Folgeachse X2 zur Leitachse X1 soll 1.3 sein.

Nenner des Koppelfaktors

Syntax: CPLDEN [FAx, LAx] = <Wert>
Bezeichnung: **Coupling Lead Denominator**
Funktionalität: Definiert den Nenner des Koppelfaktors für das Koppelgesetz der Folgeachse/-spindel FAx zur Leitachse/-spindel LAx.
Wert: Typ: REAL
Wertebereich: -2^{31} bis $+2^{31}$
Standardwert: +1.0

Beispiel:

Programmierung	Kommentar
CPLDEN[X2,X1]=2	; Der Nenner des Koppelfaktors der Kopplung der Folgeachse X2 zur Leitachse X1 soll 2 sein.

Programmierung: Kurventabelle

Mit der Programmierung einer Tabellenummer wird ein zuvor aktiviertes lineares Koppelverhältnis (Koppelfaktor) deaktiviert.

Zum Leitwert der Leitachse/-spindel wird mittels der angegebenen Kurventabelle der leitachs/-spindel-spezifische Koppelanteil berechnet.

Syntax: CPLCTID [FAx, LAx] = <Wert>
Bezeichnung: **Coupling Lead Curve Table Id**

Funktionalität: Gibt die Nummer der Kurventabelle an, mit deren Hilfe berechnet wird, wie die Leitachse/-spindel L_{Ax} auf die Folgeachse/-spindel F_{Ax} einwirken soll.

Wert: Typ: INT

Wertebereich: -2¹⁵ bis +2¹⁵

Beispiel:

Programmierung	Kommentar
CPLCTID[X2,X1]=5	; Der leitachsspezifische Koppelanteil der Kopplung der Folgeachse X2 zur Leitachse X1 wird mit Kurventabelle Nr. 5 berechnet.

Randbedingungen

- Ein Koppelfaktor Null (CPLNUM=0) ist ein zulässiger Wert. In diesem Fall liefert die Leitachse/-spindel keine Weganteile für die Folgeachse/-spindel, ist aber noch Teil der Kopplung. Die Leitachse/-spindel hat also im Gegensatz zum ausgeschalteten Zustand noch Einfluss auf die Folgeachse/-spindel. Dies wirkt sich z. B. bei Reaktionen auf Fehlern, Endschaltern und NC/PLC-Nahtstellensignalen aus.
- CPLDEN=0 ist kein gültiger Wert und wird mit Alarm abgewiesen.
- CPLNUM, CPLDEN und CPLCTID sind auch in Synchronaktionen programmierbar.
- Die Verfügbarkeit nichtlinearer Koppelverhältnisse (CPLCTID) ist optionsabhängig (siehe Kapitel "Voraussetzungen (Seite 484)").

10.5.5.2 Kopplungsbezug (CPLSETVAL)

Der Folgewert kann wahlweise abgeleitet werden von:

- der Sollposition der Leitachse
- der Sollgeschwindigkeit / Drehzahl der Leitachse
- der Istposition der Leitachse

Dementsprechend können folgende Kopplungen programmiert werden:

- Sollwertkopplung
- Geschwindigkeitskopplung
- Istwertkopplung

Programmierung

Syntax:	CPLSETVAL [FA _x , LA _x] = "<Kopplungsbezug>"		
Bezeichnung:	Coupling Lead Set Value		
Funktionalität:	Legt den Abgriff der Leitachse/-spindel LA _x und den Einwirkungspunkt auf die Folgeachse/-spindel FA _x fest.		
Kopplungsbezug:	Typ:	STRING	
	Wertebereich:		
	"CMDPOS"	Commanded Position	Sollwertkopplung
	"CMDVEL"	Commanded Velocity	Geschwindigkeitskopplung
	"ACTPOS"	Actual Value	Istwertkopplung
	Standardwert:	"CMDPOS"	

Beispiel:

Programmierung	Kommentar
CPLSETVAL[X2,X1]="CMDPOS"	; Die Kopplung der Folgeachse X2 zur Leitachse X1 wird vom Sollwert abgeleitet.

Randbedingungen

Für ein Koppelmodul kann Geschwindigkeitskopplung nicht gleichzeitig mit Soll- oder Istwertkopplung für eine andere Leitachse aktiviert werden.

10.5.5.3 Koordinatenbezug (CPFRS)

Der Koordinatenbezug der Folgeachse/-spindel legt fest, in welchem Koordinatenbezugssystem der sich aus der Kopplung ergebende Kopplungsanteil wirksam wird: im Basiskoordinatensystem oder im Maschinenkoordinatensystem.

Außerdem wird festgelegt, welchen Koordinatenbezug die Leitwerte der Leitachse/-spindel haben müssen. Bei aktiver Transformation und wenn das Maschinenkoordinatensystem als Koordinatenbezug angegeben ist (CPFRS="MCS"), werden die Werte am Ausgang der Transformation als Leitwerte verwendet.

Programmierung

Syntax:	CPFRS [FA _x] = "<Koordinatenbezug>"
Bezeichnung:	Coupling Following Relation System

Funktionalität: Legt das Koordinatenbezugssystem für das Koppelmodul der Folgeachse/-spindel FAX fest.

Koordinaten-bezug: Typ: STRING

Wertebereich:

"BCS"	B asis C oordinate S ystem	Basiskoordinatensystem
"MCS"	M achine C oordinate S ystem	Maschinenkoordinatensystem

Standardwert: "BCS"

Beispiel:

Programmierung	Kommentar
CPFRS[X2]="BCS"	; Koordinatenbezug für das Koppelmodul mit der Folgeachse X2 ist das Basiskoordinatensystem.

Randbedingungen

- Der Koordinatenbezug muss beim Anlegen des Koppelmoduls angegeben werden, ansonsten wirkt der Standardwert. Ein nachträgliches Ändern ist nicht möglich.
- Eine gleichzeitig über RESET aktive Transformation und Kopplung wird nicht unterstützt. Lösung: Kopplung mit `CPMRESET="OF"` bei RESET ausschalten und mit `CPMSTART="ON"` beim Teileprogrammstart wieder einschalten.
- Der gleichzeitige Einsatz der bisherigen Funktion "Axiale Kopplung im Maschinenkoordinatensystem (MKS-Kopplung)" und der Generischen Kopplung wird nicht unterstützt.
- `CPFRS` ist nicht im Hauptlauf verfügbar.

10.5.5.4 Satzwechselverhalten (CPBC)

Mit dem Satzwechselkriterium kann angegeben werden, unter welchen Bedingungen der Satzwechsel bei aktivierter Kopplung in der Abarbeitung des Teileprogramms erlaubt werden soll. Der Zustand der Kopplung beeinflusst somit das Satzwechselverhalten. Ist die angegebene Bedingung nicht erfüllt, so wird der Satzwechsel gesperrt. Das Satzwechselkriterium wird nur bei aktiver Kopplung ausgewertet.

Das Satzwechselkriterium kann mit dem Schlüsselwort `CPBC` oder mit dem Programmierbefehl `WAITC` definiert werden. Die zuletzt programmierte Anweisung ist gültig.

Programmierung mit CPBC

Syntax:	CPBC [FA _x] = "<Satzwechselkriterium>"
Bezeichnung:	Coupling Block Change Criterium
Funktionalität:	Legt das Satzwechselkriterium bei aktivierter Kopplung fest.
Satzwechsel-kriteri- um:	Typ: STRING
	Wertebereich:
	"NOC" Satzwechsel erfolgt unabhängig vom Kopplungs- zustand.
	"IPOSTOP" Satzwechsel erfolgt beim sollwertseitigen Synchron- lauf.
	"COARSE" Satzwechsel erfolgt beim istwertseitigen Synchronlauf "Grob".
	"FINE" Satzwechsel erfolgt beim istwertseitigen Synchronlauf "Fein".
	Standardwert: "NOC"

Beispiel:

Programmierung	Kommentar
CPBC[X2]="IPOSTOP"	; Der Satzwechsel beim Abarbeiten des Teileprogramms er- folgt beim sollwertseitigen Synchronlauf (bei aktivier- ter Kopplung zur Folgeachse X2).

Programmierung mit WAITC

Syntax:	WAITC (FA _{x1} , BC)
Bezeichnung:	Wait for Coupling Condition
Funktionalität:	Legt das Satzwechselkriterium bei aktivierter Kopplung fest.
Parameter:	Fax: Bezeichnet die Folgeachse und damit das Koppelmodul. BC: Bestimmt das gewünschte Satzwechselkriterium.
FA _x :	Typ: STRING
	Wertebereich: Achsen des Kanals

BC:	Typ: STRING
	Wertebereich:
	"NOC" Satzwechsel erfolgt unabhängig vom Kopplungszustand.
	"IPOSTOP" Satzwechsel erfolgt beim sollwertseitigen Synchronlauf.
	"COARSE" Satzwechsel erfolgt beim istwertseitigen Synchronlauf "Grob".
	"FINE" Satzwechsel erfolgt beim istwertseitigen Synchronlauf "Fein".
	Standardwert: "NOC"

Beispiel:

Programmierung	Kommentar
WAITC (X2, "IPOSTOP")	; Der Satzwechsel beim Abarbeiten des Teileprogramms erfolgt beim sollwertseitigen Synchronlauf (bei aktivierter Kopplung zur Folgeachse X2).

Randbedingungen

WAITC kann im Unterschied zum Schlüsselwort CPBC nur alleine im Satz stehen.

10.5.5.5 Synchronposition der Folgeachse beim Einschalten (CPFPOS+CPON)

Beim Einschalten der Kopplung (CPON) kann für die Folgeachse das Anfahren einer definierten Synchronposition programmiert werden.

Die Synchronposition beim Einschalten wird sofort wirksam. Die Gesamtposition, die sich aus der Synchronposition und dem Koppelgesetz ergibt, wird entsprechend dem gesetzten Synchronisationsmodus (CPFMSON) unter Beachtung der Dynamikgrenzen angefahren.

Programmierung

Syntax:	CPON=FA _x CPFPOS [FA _x] = <Wert>
Bezeichnung:	Coupling Following Position
Funktionalität:	Legt die Synchronposition der Folgeachse beim Einschalten fest. Bei der Positionsangabe ist AC, IC und GP möglich.
Wert:	Typ: REAL

Wertebereich: Alle Positionen innerhalb der Verfahrbereichsgrenzen

Beispiel:

Programmierung	Kommentar
CPON=X2 CPFPOS[X2]=100	; Die Kopplung zur Folgeachse X2 wird aktiviert. Als Synchronposition der Folgeachse X2 wird 100 genommen.

Randbedingungen

- CPFPOS ist nur mit dem Einschaltkommando CPON/CPLON als Synchronposition wirksam. Mit dem Ausschaltkommando CPOF wird CPFPOS als Ausschaltposition gewertet (siehe Kapitel "Position der Folgeachse beim Ausschalten (Seite 512)").
- Bei CPFPOS ohne Einschaltkommando erfolgt ein Alarm.
- Wird die Synchronposition der Folgeachse beim Einschalten nicht gesetzt, wird die aktuelle Position der Folgeachse als Synchronposition wirksam. Mit Hilfe der Programmanweisung IC kann die aktuelle Position noch verschoben werden.
- Unabhängig von der programmierbaren Maßangabe (G70/G71) erfolgt die Positionsangabe im eingestellten Grundsystem.

Teileprogrammausschnitt (Beispiel)

Programmierung	Kommentar
CPON=(X2) CPFPOS[X2]=100	; Die Kopplung zur Folgeachse X2 wird aktiviert. Als Synchronposition der Folgeachse wird 100 genommen.
...	
G00 X2=123	; Folgeachse X2 wird auf Position 123 verfahren.
CPON=(X2)	; Als Synchronposition der Folgeachse wird die aktuelle Position (=123) genommen. Die vorab aktive Synchronposition 100 wird dadurch unwirksam.

10.5.5.6 Synchronposition der Leitachse beim Einschalten (CPLPOS)

Die als Leitwert eingehende aktuelle Leitachseposition kann verschoben werden. Die Synchronposition der Leitachse definiert somit den Nullpunkt der Eingangsgröße.

Programmierung

Syntax: CPLPOS [FAx, LAx] = <Wert>

Bezeichnung: Coupling Lead Position

Funktionalität: Legt die Synchronposition der Leitachse beim Einschalten fest. Bei der Positionsangabe ist nur AC möglich.

Wert: Typ: REAL

Wertebereich: Alle Positionen innerhalb der Verfahrbereichsgrenzen

Beispiel:

Programmierung	Kommentar
CPLPOS[X2,X1]=200	; Als Synchronposition der Leitachse X1 der Kopplung zur Folgeachse X2 wird 200 genommen.

Randbedingungen

- CPLPOS kann nur mit dem Einschaltkommando CPON / CPLON gesetzt werden. Bei CPLPOS ohne Einschaltkommando erfolgt ein Alarm.
- Wird die Synchronposition der Leitachse beim Einschalten (CPON) nicht gesetzt, wird die aktuelle Position der Leitachse als Synchronposition und damit als Nullpunkt der Eingangsgröße wirksam.
- Unabhängig von der programmierbaren Maßangabe (G70 / G71) erfolgt die Positionsangabe im eingestellten Grundsystem.

Teileprogrammausschnitt (Beispiel)

Programmierung	Kommentar
CPON=(X2) CPFPOS[X2]=100 CPLPOS[X2,X1]=200	; Die Kopplung zur Folgeachse X2 wird aktiviert. Als Synchronposition der Folgeachse wird 100, als Synchronposition der Leitachse X1 wird 200 genommen.
...	
N20 X1=280 F1000	; Leitachse X1 wird auf Position 280 verfahren.
CPON=(X2)	; Als Synchronposition der Leitachse wird die aktuelle Position X1=280 genommen. Die vorab aktive Synchronposition der Leitachse (200) wird dadurch unwirksam.

10.5.5.7 Synchronisationsmodus (CPFMSON)

Der Synchronisationsmodus bestimmt das Synchronisationsverhalten beim Einschalten der Kopplung.

Programmierung

Syntax: CPFMSON [FAx] = "<Synchronisationsmodus>"

Bezeichnung: **Coupling Following Mode Strategy On**

Funktionalität: Legt den Synchronisationsmodus beim Einkoppeln fest.

Synchronisationsmodus: Typ: STRING

Wertebereich:

"CFAST"	Closed Coupling Fast	Die Kopplung wird zeitoptimiert geschlossen.
"CCOARSE"	Closed If Gap Coarse	Die Kopplung wird erst eingeschaltet, wenn sich die gemäß Koppelgesetz geforderte Folgeachsposition im Bereich der aktuellen Folgeachsposition befindet.
"NTGT"	Next Tooth Gap Time Optimized	Die nächste Zahnücke wird zeitoptimiert angefahren.
"NTGP"	Next Tooth Gap Path Optimized	Die nächste Zahnücke wird wegoptimiert angefahren.
"NRGT"	Next Ratio Gap Time Optimized	Das nächste Segment gemäß Verhältnis aus Gangzahl zu Zähnezah wird zeitoptimiert angefahren.
"NRGP"	Next Ratio Gap Path Optimized	Das nächste Segment gemäß Verhältnis aus Gangzahl zu Zähnezah wird wegoptimiert angefahren.
"ACN"	Absolute Coordinate Negative	Nur bei Rundachsen! Die Rundachse verfährt in negativer Achsrichtung die Synchronposition an. Die Synchronisation erfolgt sofort.
"ACP"	Absolute Coordinate Positive	Nur bei Rundachsen! Die Rundachse verfährt in positiver Achsrichtung die Synchronposition an. Die Synchronisation erfolgt sofort.
"DCT"	Direct Coordinate Time Optimized	Nur bei Rundachsen! Die Rundachse verfährt zeitoptimiert zur programmierten Synchronposition an. Die Synchronisation erfolgt sofort.

"DCP" **Direct Coordinate Path** Nur bei Rundachsen!
 Optimized Die Rundachse verfährt **weg-optimiert** zur programmierten Synchronposition an. Die Synchronisation erfolgt sofort

Standardwert: "CFAST"

Beispiel:

Programmierung	Kommentar
CPFMSON[X2]="CFAST"	; Als Synchronisationsmodus der Kopplung zur Folgeachse X2 wird CFAST genommen.

10.5.5.8 Verhalten der Folgeachse beim Einschalten (CPFMON)

Das Verhalten der Folgeachse/-spindel beim Einschalten der Kopplung kann mit dem Schlüsselwort `CPFMON` programmiert werden.

Programmierung

Syntax: `CPFMON [FAx] = "<Einschaltverhalten>"`

Bezeichnung: **Coupling Following Mode On**

Funktionalität: Legt das Verhalten der Folgeachse/-spindel beim Einschalten der Kopplung fest.

Einschaltverhalten: Typ: STRING

Wertebereich:

"STOP" Stop	Nur bei Spindeln! Eine aktive Bewegung der Folgespindel wird vor dem Einschalten gestoppt.
"CONT" Continue	Nur bei Spindeln und Hauptlaufachsen! Die aktuelle Bewegung der Folgeachse/-spindel wird in die Kopplung als Startbewegung übernommen.

"ADD"	Additional	Nur bei Spindeln! Die Bewegungsanteile der Kopplung wirken zusätzlich zur aktuellen überlagerten Bewegung, d. h. die aktuelle Bewegung der Folgeachse/-spindel wird als überlagerte Bewegung beibehalten.
Standardwert:	"STOP"	

Beispiel:

Programmierung	Kommentar
CPFMON[X2]="CONT"	; Die aktuelle Bewegung der Folgeachse X2 wird als Startbewegung übernommen.

10.5.5.9 Verhalten der Folgeachse beim Ausschalten (CPFMOF)

Das weitere Verhalten der Folgeachse/-spindel beim vollständigen Ausschalten einer aktiven Kopplung kann mit dem Schlüsselwort `CPFMOF` programmiert werden.

Programmierung

Syntax:	<code>CPFMOF [FAx] = "<Ausschaltverhalten>"</code>	
Bezeichnung:	Coupling Following Mode Off	
Funktionalität:	Legt das Verhalten der Folgeachse/-spindel beim vollständigen Ausschalten der Kopplung fest.	
Ausschaltverhalten:	Typ: STRING	
	Wertebereich:	
	"STOP" Stop	Stopp der Folgeachse/-spindel. Eine aktive überlagerte Bewegung wird zum Stillstand abgebremst. Danach wird die Kopplung geöffnet (deaktiviert).
	"CONT" Continue	Nur bei Spindeln und Hauptlaufachsen! Die Folgespindel läuft mit der zum Ausschaltzeitpunkt aktuellen Drehzahl/Geschwindigkeit weiter.
	Standardwert:	"STOP"

Beispiel:

Programmierung	Kommentar
CPFMOF[S2]="CONT"	; Die Folgespindel S2 läuft mit der zum Ausschaltzeitpunkt aktuellen Drehzahl/Geschwindigkeit weiter.

10.5.5.10 Position der Folgeachse beim Ausschalten (CPFPOS+CPOF)

Beim Ausschalten einer Kopplung (CPOF) kann für die Folgeachse das Anfahren einer bestimmten Position gefordert werden.

Programmierung

Syntax: CPOF= (FAx) CPFPOS [FAx]= <Wert>

Funktionalität: Legt die Ausschaltposition der Folgeachse FAx fest.

Wert: Typ: REAL

Wertebereich: Alle Positionen innerhalb der Verfahrbereichsgrenzen

Beispiel:

Programmierung	Kommentar
CPOF=(X2) CPFPOS[X2]=100	; Die Kopplung zur Folgeachse X2 wird deaktiviert. Die gesetzte Ausschaltposition der Folgeachse (100) wird angefahren.

Randbedingungen

- CPFPOS ist nur mit dem Ausschaltkommando CPOF als Ausschaltposition wirksam. Mit dem Schaltkommando CPON wird CPFPOS als Einschaltposition gewertet (siehe Kapitel "Synchronposition der Folgeachse beim Einschalten (Seite 506)").
- Das Setzen einer Ausschaltposition ist nur zulässig mit dem Ausschaltmodus: CPMOF=STOP
- Die Ausschaltposition wird mit maximaler Dynamik angefahren.
- Das Satzwechselverhalten ist abhängig von der Parametrierung des Schlüsselwortes CPBC.

10.5.5.11 Zustand bei RESET (CPMRESET)

Bei RESET kann die Kopplung aktiviert, deaktiviert oder der aktuelle Zustand erhalten bleiben. Das Verhalten ist für jedes Koppelmodul getrennt einstellbar.

Programmierung

Syntax: CPMRESET [FAx] = "<Reset-Verhalten>"

Bezeichnung: **Coupling Mode RESET**

Funktionalität: Legt das Verhalten einer Kopplung bei RESET fest.

Reset-Verhalten: Typ: STRING

Wertebereich:

"NONE"	Der aktuelle Zustand der Kopplung bleibt erhalten.
"ON"	Ist das entsprechende Koppelmodul angelegt, so wird die Kopplung eingeschaltet. Es werden alle definierten Leitachsbeziehungen aktiviert. Dies erfolgt auch, wenn bereits alle oder Teile dieser Leitachsbeziehungen aktiv sind, d. h. auch bei einer vollständig aktivierten Kopplung erfolgt eine Neusynchronisation.
"OF"	Eine aktive überlagerte Bewegung wird zum Stillstand abgebremst. Danach wird die Kopplung ausgeschaltet. Wurde das entsprechende Koppelmodul ohne explizite Definition (CPDEF) angelegt, so wird das Koppelmodul gelöscht. Im anderen Fall bleibt es weiterhin angelegt, d. h. es kann weiterhin verwendet werden.
"OFC"	Nur bei Spindeln möglich! Die Folgespindel läuft mit der zum Ausschaltzeitpunkt aktuellen Drehzahl/Geschwindigkeit weiter. Die Kopplung wird ausgeschaltet. Wurde das entsprechende Koppelmodul ohne explizite Definition (CPDEF) angelegt, so wird das Koppelmodul gelöscht. Im anderen Fall bleibt es weiterhin angelegt, d. h. es kann weiterhin verwendet werden.
"DEL"	Eine aktive überlagerte Bewegung wird zum Stillstand abgebremst. Danach wird die Kopplung deaktiviert und anschließend gelöscht.
"DELC"	Nur bei Spindeln möglich! Die Folgespindel läuft mit der zum Ausschaltzeitpunkt aktuellen Drehzahl/Geschwindigkeit weiter. Die Kopplung wird deaktiviert und anschließend gelöscht.

Standardwert: "NONE"

Beispiel:

Programmierung	Kommentar
CPMRESET[X2]="DEL"	; Bei RESET wird die Kopplung zur Folgeachse X2 deaktiviert und anschließend gelöscht.

Randbedingungen

- Die mit `CPMRESET` gesetzte Kopplungseigenschaft bleibt solange erhalten, bis das Koppelmodul gelöscht wird (`CPDEL`).
- Bei gesetztem Kopplungstyp (`CPSETTYPE="TRAIL", "LEAD", "EG" oder "COUP"`) wird das Verhalten bei `RESET` durch das folgende Maschinendatum bestimmt:
MD20110 \$MN_RESET_MODE_MASK (Festlegung der Steuerungs-Grundstellung nach Reset / TP-Ende)
→ Siehe Abschnitt "Voreinstellungen" in "Kopplungstypen (CPSETTYPE) (Seite 536)".

10.5.5.12 Zustand bei Teileprogrammstart (CPMSTART)

Beim Teileprogrammstart kann eine Kopplung aktiviert, deaktiviert oder der aktuelle Zustand erhalten bleiben. Das Verhalten ist für jedes Koppelmodul getrennt einstellbar.

Programmierung

Syntax: `CPMSTART [FAx] = <Wert>`

Bezeichnung: **Coupling Mode Start**

Funktionalität: Legt das Verhalten einer Kopplung beim Teileprogrammstart fest.

Wert: Typ: `STRING`

Wertebereich:

"NONE" Der aktuelle Zustand der Kopplung bleibt erhalten.

"ON" Ist das entsprechende Koppelmodul angelegt, so wird die Kopplung eingeschaltet. Es werden alle definierten Leitachsbeziehungen aktiviert. Dies erfolgt auch, wenn bereits alle oder Teile dieser Leitachsbeziehungen aktiv sind, d. h. auch bei einer vollständig aktivierten Kopplung erfolgt eine Neusynchronisation.

"OF" Die Kopplung wird ausgeschaltet. Wurde das entsprechende Koppelmodul ohne explizite Definition (`CPDEF`) angelegt, so wird das Koppelmodul gelöscht. Im anderen Fall bleibt es weiterhin angelegt, d. h. es kann weiterhin verwendet werden.

"DEL" Die Kopplung wird deaktiviert und anschließend gelöscht.

Standardwert: `"NONE"`

Beispiel:

Programmierung	Kommentar
CPMSTART[X2]="ON"	; Beim Teileprogrammstart wird die Kopplung zur Folgeachse X2 eingeschaltet.

Randbedingungen

- Die mit CPMSTART gesetzte Kopplungseigenschaft bleibt solange erhalten, bis das Koppelmodul gelöscht wird (CPDEL).
- Bei gesetztem Kopplungstyp (CPSETTYPE="TRAIL", "LEAD", "EG" oder "COUP") wird das Verhalten bei Teileprogrammstart durch folgendes Maschinendatum bestimmt: MD20112 \$MC_START_MODE_MASK (Festlegung der Steuerungs-Grundstellung bei NC-START)
→ Siehe Abschnitt "Voreinstellungen" in "Kopplungstypen (CPSETTYPE)" (Seite 536).

10.5.5.13 Zustand bei Teileprogrammstart unter Suchlauf via Programmtest (CPMPRT)

Beim Teileprogrammstart unter Satzsuchlauf via Programmtest (SERUPRO) kann eine Kopplung aktiviert, deaktiviert oder der aktuelle Zustand erhalten bleiben. Das Verhalten ist für jedes Koppelmodul getrennt einstellbar.

Programmierung

Syntax: CPMPT [FAx] = <Wert>

Bezeichnung: **Coupling Mode Program Test**

Funktionalität: Legt das Verhalten einer Kopplung beim Teileprogrammstart unter Satzsuchlauf via Programmtest fest.

Wert: Typ: STRING

Wertebereich:

- "NONE" Der aktuelle Zustand der Kopplung bleibt erhalten.
- "ON" Ist das entsprechende Koppelmodul angelegt, so wird die Kopplung eingeschaltet. Es werden alle definierten Leitachsbeziehungen aktiviert. Dies erfolgt auch, wenn bereits alle oder Teile dieser Leitachsbeziehungen aktiv sind, d. h. auch bei einer vollständig aktivierten Kopplung erfolgt eine Neusynchronisation.
- "OF" Die Kopplung wird ausgeschaltet. Wurde das entsprechende Koppelmodul ohne explizite Definition (CPDEF) angelegt, so wird das Koppelmodul gelöscht. Im anderen Fall bleibt es weiterhin angelegt, d. h. es kann weiterhin verwendet werden.
- "DEL" Die Kopplung wird deaktiviert und anschließend gelöscht.

Standardwert: "NONE"

Beispiel:

Programmierung	Kommentar
CPMPRT[X2]="ON"	; Beim Teileprogrammstart unter Satzsuchlauf via Programmtest wird die Kopplung zur Folgeachse X2 eingeschaltet.

Randbedingungen

- Die mit CPMPRT gesetzte Kopplungseigenschaft bleibt solange erhalten, bis das Koppelmodul gelöscht wird (CPDEL).
- Wenn CPMPRT="NONE" gesetzt ist, dann wird das Verhalten bei Teileprogrammstart unter Satzsuchlauf via Programmtest (SERUPRO) durch CPMSTART bestimmt.
- Bei gesetztem Kopplungstyp (CPSETTYPE="TRAIL", "LEAD", "EG" oder "COUP") wird das Verhalten bei Teileprogrammstart unter Satzsuchlauf via Programmtest durch folgende Maschinendaten bestimmt:
 MD22620 \$MC_START_MODE_MASK_PRT (Festlegung der Steuerungs-Grundstellung bei speziellen Starts)
 MD22621 \$MC_ENABLE_START_MODE_MASK_PRT (Freischaltung von MD22620)
 MD20112 \$MC_START_MODE_MASK (Festlegung der Steuerungs-Grundstellung bei NC-START)
 → Siehe Abschnitt "Voreinstellungen" in "Kopplungstypen (CPSETTYPE) (Seite 536)".

10.5.5.14 Verschiebung / Skalierung (CPLINTR, CPLINSC, CPLOUTTR, CPLOUTSC)

Eine bestehende Koppelbeziehung zwischen einer Folgeachse und einer Leitachse kann skaliert und verschoben werden.

Die Wirkung dieser Funktionen auf den Gesamtsollwert der Folgeachse wird aus folgender Formel ersichtlich:

$$FA_{Total} = FA_{cmd} + \{ transOut_1 + scaleOut_1 * [(LA_1 - SynPosLA_1) * scaleIn_1 + transIn_1] * KF_1 \} + \{ transOut_2 + scaleOut_2 * [(LA_2 - SynPosLA_2) * scaleIn_2 + transIn_2] * KF_2 \}$$

FA_{Total}	Gesamtsollwert der Folgeachse
FA_{Cmd}	Im Teileprogramm gesetzter Sollwert
$LA_{1/2}$	Soll- oder Istwert der 1-ten bzw. 2-ten Leitachse/-wert
$SynPosLA_{1/2}$	Synchronposition der 1-ten bzw. 2-ten Leitachse/-wert
$scaleIn_{1/2}$	Skalierfaktor des 1-ten bzw. 2-ten Leitwerts
$transIn_{1/2}$	Verschiebung des 1-ten bzw. 2-ten Leitwerts
$KF_{1/2}$	Koppelfaktor der 1-ten bzw. 2-ten Leitachse/-wert
$scaleOut_{1/2}$	Skalierfaktor des 1-ten bzw. 2-ten Ausgangswerts
$transOut_{1/2}$	Verschiebung des 1-ten bzw. 2-ten Ausgangswerts

Hinweis

Skalier- und Verschiebungswerte können für jede Leitachse definiert werden.

Programmierung

Verschiebung des Eingangswerts

Syntax: `CPLINTR [FAx, LAx] = <Wert>`

Bezeichnung: **Coupling Lead In Translation Displacement**

Funktionalität: Definiert den Verschiebewert für den Eingangswert der Leitachse LAx.

Wert: Typ: REAL

Standardwert: 0

Beispiel:

Programmierung	Kommentar
<code>CPLINTR[X2,X1]==-50</code>	; Der Eingangswert der Leitachse X1 wird um den Wert 50 in negativer Achsrichtung verschoben.

Skalierung des Eingangswerts

Syntax: `CPLINSC [FAx, LAx] = <Wert>`

Bezeichnung: **Coupling Lead In Scale Factor**

Funktionalität: Definiert den Skalierfaktor für den Eingangswert der Leitachse LAx.

Wert: Typ: REAL

Standardwert: 1

Beispiel:

Programmierung	Kommentar
CPLINSC[X2,X1]=0.5	; Der Eingangswert der Leitachse X1 wird mit dem Faktor 0.5 multipliziert.

Verschiebung des Ausgangswerts

Syntax: CPLOUTTR[FA_x, LA_x] = <Wert>

Bezeichnung: **Coupling Lead Out Translation Displacement**

Funktionalität: Definiert den Verschiebewert für den Ausgangswert der Kopplung der Folgeachse FA_x zur Leitachse LA_x.

Wert: Typ: REAL

Standardwert: 0

Beispiel:

Programmierung	Kommentar
CPLOUTTR[X2,X1]=100	; Der Ausgangswert der Kopplung der Folgeachse X2 zur Leitachse X1 wird um den Wert 100 in positiver Achsrichtung verschoben.

Skalierung des Ausgangswerts

Syntax: CPLOUTSC[FA_x, LA_x] = <Wert>

Bezeichnung: **Coupling Lead Out Scale Factor**

Funktionalität: Definiert den Skalierfaktor für den Ausgangswert der Kopplung der Folgeachse FA_x zur Leitachse LA_x.

Wert: Typ: REAL

Standardwert: 1

Beispiel:

Programmierung	Kommentar
CPLOUTSC[X2,X1]=3	; Der Ausgangswert der Kopplung der Folgeachse X2 zur Leitachse X1 wird mit dem Faktor 3 multipliziert.

Hinweis

Die folgenden, in der bestehenden Kopplungsart "Leitwertkopplung" verwendeten Settingdaten werden bei der Generischen Kopplung unabhängig vom eingestellten Kopplungstyp (CPSETTYPE) berücksichtigt:

SD43102 \$SA_LEAD_OFFSET_IN_POS[Fax] (Verschiebung des Leitwerts)

SD43104 \$SA_LEAD_SCALE_IN_POS[Fax] (Skalierung des Leitwerts)

SD43106 \$SA_LEAD_OFFSET_OUT_POS[Fax] (Verschiebung des Funktionswerts der Kurventabelle)

SD43108 \$SA_LEAD_SCALE_OUT_POS[Fax] (Skalierung des Funktionswerts der Kurventabelle)

Diese Settingdaten wirken:

- auf alle Leitachsen, die mittels einer Kurventabelle an die Folgeachse gekoppelt sind. Dies ist bei Kopplungen mit mehr als einer Leitachse zu berücksichtigen!
- additiv zu den CP-Schlüsselwörtern CPLINTR, CPLINSC, CPLOUTTR und CPLOUTSC.

10.5.5.15 Synchronlaufüberwachung Stufe 1 (CPSYNCOV, CPSYNFIP, CPSYNCOV, CPSYNFIV)

Synchronlaufüberwachung Stufe 1

In jedem Interpolatortakt wird der Synchronlauf des Kopplungsverbandes soll- und istwertseitig überwacht. Die Synchronlaufüberwachung spricht an, sobald die **Synchronlaufdifferenz** (Differenz des Soll- oder Istwerts der Folgeachse zu dem Wert, der sich nach dem Kopplungsgesetz aus den Soll- oder Istwerten der Leitachsen ergäbe) einen der folgenden projektier- und programmierbaren Schwellenwerte erreicht:

- bei Soll-/Istwertkopplung (siehe "Kopplungsbezug (CPLSETVAL) (Seite 502)":
 - Schwellenwert Positionssynchronlauf "Grob"
 - Schwellenwert Positionssynchronlauf "Fein"
- bei Geschwindigkeitskopplung (siehe "Kopplungsbezug (CPLSETVAL) (Seite 502)":
 - Schwellenwert Geschwindigkeitssynchronlauf "Grob"
 - Schwellenwert Geschwindigkeitssynchronlauf "Fein"

Die aktuelle Synchronlaufdifferenz kann mit den folgenden CP-Systemvariablen ausgelesen werden:

Systemvariable	Bedeutung
\$AA_SYNCDIFF [Fax]	Synchronlaufdifferenz sollwertseitig
\$VA_SYNCDIFF [Fax]	Synchronlaufdifferenz istwertseitig

Hinweis

Die Synchronlaufdifferenzen sind vorzeichenbehaftet und ermöglichen es somit, den Vor- oder Nachlauf der Folgeachse zu ermitteln.

Zustand der Kopplung beim Synchronlauf

Zustand	Beschreibung
Nicht synchronisiert	Solange die Synchronlaufdifferenz größer ist als der Schwellenwert für Positionssynchronlauf "Grob" bzw. Geschwindigkeitssynchronlauf "Grob", wird der Koppelverband als nicht synchron bezeichnet.
Synchronlauf "Grob" erreicht	Die Synchronlaufdifferenz hat den Schwellenwert für Positionssynchronlauf "Grob" bzw. Geschwindigkeitssynchronlauf "Grob" erreicht. Im Falle der istwertseitigen Synchronlaufdifferenz wird das folgende NC/PLC-Nahtstellensignal gesetzt: DB31, ... DBX98.1 (Synchronlauf grob)
Synchronlauf "Fein" erreicht	Die Synchronlaufdifferenz hat den Schwellenwert für Positionssynchronlauf "Fein" bzw. Geschwindigkeitssynchronlauf "Fein" erreicht. Im Falle der istwertseitigen Synchronlaufdifferenz wird das folgende NC/PLC-Nahtstellensignal gesetzt: DB31, ... DBX98.0 (Synchronlauf fein)

Der Zustand der Kopplung beim Synchronlauf kann mit der folgenden Systemvariablen gelesen werden:

Systemvariable	Bedeutung	
\$AA_SYNC [FAX]	Zustand der Kopplung	
	Wert	Zustand
	0	Nicht synchronisiert
	1	Synchronlauf "Grob" erreicht
	2	Synchronlauf "Fein" erreicht

Signalreaktion

Synchronlaufsignale selbst bewirken keinen Stopp der beteiligten Achsen, können diesen aber mittels Synchronaktion oder NC/PLC-Nahtstellensignalen auslösen (siehe Kapitel "R3: Erweitertes Stillsetzen und Rückziehen (Seite 573)").

Projektierung

Die Schwellenwerte für die erste Stufe der Synchronlaufüberwachung werden eingestellt:

- für Soll-/Istwertkopplung in den Maschinendaten:
 - MD37200 \$MA_COUPLE_POS_TOL_COARSE (Schwellwert für "Synchronlauf grob")
 - MD37210 \$MA_COUPLE_POS_TOL_FINE (Schwellwert für "Synchronlauf fein")
- für Geschwindigkeitskopplung in den Maschinendaten:
 - MD37220 \$MA_COUPLE_VELO_TOL_COARSE (Geschwindigkeitstoleranz "grob")
 - MD37230 \$MA_COUPLE_VELO_TOL_FINE (Geschwindigkeitstoleranz "fein")

Programmierung

Über CP-Schlüsselwörter können die Schwellenwerte für die erste Stufe der Synchronlaufüberwachung auch programmiert werden:

Schwellenwert Positionssynchronlauf "Grob"

Syntax: CPSYNCOF [FAx] = <Wert>

Bezeichnung: **Coupling Synchronous Difference Coarse Position**

Funktionalität: Definiert den Schwellwert für den Positionssynchronlauf "Grob".

Wert: Typ: REAL

Der Standardwert entspricht der Einstellung im Maschinendatum:
MD37200 \$MA_COUPLE_POS_TOL_COARSE [FAx]

Schwellenwert Positionssynchronlauf "Fein"

Syntax: CPSYNFIP [FAx] = <Wert>

Bezeichnung: **Coupling Synchronous Difference Fine Position**

Funktionalität: Definiert den Schwellwert für den Positionssynchronlauf "Fein".

Wert: Typ: REAL

Der Standardwert entspricht der Einstellung im Maschinendatum:
MD37210 \$MA_COUPLE_POS_TOL_FINE [FAx]

Schwellenwert Geschwindigkeitssynchronlauf "Grob"

Syntax: CPSYNCOV [FAx] = <Wert>

Bezeichnung: **Coupling Synchronous Difference Coarse Velocity**

Funktionalität: Definiert den Schwellwert für den Geschwindigkeitssynchronlauf "Grob".

Wert: Typ: REAL

Der Standardwert entspricht der Einstellung im Maschinendatum:
MD37220 \$MA_COUPLE_VELO_TOL_COARSE [FAx]

Schwellenwert Geschwindigkeitssynchronlauf "Fein"

Syntax: CPSYNFIV [FAx] = <Wert>

Bezeichnung: **Coupling Synchronous Difference Fine Velocity**

Funktionalität: Definiert den Schwellwert für den Geschwindigkeitssynchronlauf "Fein".

Wert: Typ: REAL

Der Standardwert entspricht der Einstellung im Maschinendatum:
MD37230 \$MA_COUPLE_VELO_TOL_FINE [FAx]

Beispiel

Programmcode	Kommentar
CPDEF=(S2) CPLA[S2]=(S1)	; Definition einer Spindelkopplung: Leitspindel S1 zur Folgespindel S2
CPON=(S2) CPSYNCOPI[S2]=0.5	; Einschalten der Kopplung zur Folgespindel S2.
CPSYNFIP[S2]=0.25	Die Schwellwerte für den Positionssynchronlauf wer- den auf 0.5 ("grob") und 0.25 ("fein") gesetzt.
...	

Randbedingungen

- Bei der Betrachtung der Synchronlaufdifferenz wird eine evtl. aktive Kopplungskaskade nicht berücksichtigt. Das bedeutet: wenn im betrachteten Koppelmodul die Leitachse selbst eine Folgeachse in einem anderen Koppelmodul ist, dann wird die aktuelle Ist- oder Sollposition als Eingangsgröße zur Berechnung der Synchronlaufdifferenz verwendet. Diese Synchronlaufdifferenz gibt somit nicht den gesamten Synchronlauffehler der Kaskade an.
- Wenn die Leitachse keine reale Achse sondern eine simulierte Achse ist, dann steht der Istwert zur istwertseitigen Synchronlaufüberwachung nicht zur Verfügung. In diesem Fall werden die modellierten Istwerte (gemäß Maschinendaten-Einstellung) herangezogen.

10.5.5.16 Synchronlaufüberwachung Stufe 2 (CPSYN COP2, CPSYN FIP2)

Synchronlaufüberwachung Stufe 2

Bei aktiver CP-Positionskopplung (Soll- oder Istwertkopplung; siehe "Kopplungsbezug (CPLSETVAL) (Seite 502)") kann mit der zweiten Stufe der Synchronlaufüberwachung nach Erreichen des Satzwechselkriteriums "COARSE"/"FINE" (siehe "Satzwechselverhalten (CPBC) (Seite 504)") die Einhaltung einer von den Schwellenwerten der ersten Stufe unabhängigen Synchronlauf toleranz istwertseitig überwacht werden.

Für die zweite Stufe der Synchronlaufüberwachung sind die beiden folgenden Schwellenwerte zu projektieren bzw. programmieren:

- Schwellenwert Positionssynchronlauf "Grob" 2
- Schwellenwert Positionssynchronlauf "Fein" 2

Technischer Hintergrund

Im Rahmen der Überwachungen "Synchronlauf erreicht" und "Synchronfenster verlassen" ist es für eine richtige Beurteilung von eventuell auftretenden Problemen beim Mehrkantdrehen (Oberfläche, Verschnitt usw.) – und teilweise auch bei schwingender Synchronspindel – sehr nützlich, einen Toleranzbereich unabhängig von den Synchronlauf toleranzen grob/fein der ersten Stufe der Synchronlaufüberwachung definieren zu können (analog zu Genauhalt und Stillstandsüberwachung bei Achsen), um eine projektierbare Fehlermeldung bzw. Warnung erhalten zu können.

Projektierung

Die Schwellenwerte für die zweite Stufe der Synchronlaufüberwachung werden eingestellt in den Maschinendaten:

MD37202 \$MA_COUPLE_POS_TOL_COARSE_2 (zweiter Schwellwert für "Synchronlauf grob")

MD37212 \$MA_COUPLE_POS_TOL_FINE_2 (zweiter Schwellwert für "Synchronlauf fein")

Hinweis

Ist der jeweilige Schwellenwert = 0, dann ist die zugehörige Überwachung inaktiv. Dieses ist auch der Standardwert, damit die Kompatibilität zu älteren Softwareständen erhalten bleibt.

Programmierung

Über CP-Schlüsselwörter können die Schwellenwerte für die zweite Stufe der Synchronlaufüberwachung auch programmiert werden:

Schwellenwert Positionssynchronlauf "Grob" 2

Syntax: CPSYN COP2 [FAX] = <Wert>

Bezeichnung: **Coupling Synchronous Difference Coarse Position 2**

Funktionalität: Definiert den zweiten Schwellwert für den Positionssynchronlauf "Grob".

Wert: Typ: REAL

Der Standardwert entspricht der Einstellung im Maschinendatum:
MD37202 \$MA_COUPLE_POS_TOL_COARSE_2 [FAx]

Schwellenwert Positionssynchronlauf "Fein" 2

Syntax: CPSYNFIP2 [FAx] = <Wert>

Bezeichnung: **Coupling Synchronous Difference Fine Position 2**

Funktionalität: Definiert den zweiten Schwellwert für den Positionssynchronlauf "Fein".

Wert: Typ: REAL

Der Standardwert entspricht der Einstellung im Maschinendatum:
MD37212 \$MA_COUPLE_POS_TOL_FINE_2 [FAx]

Die Programmierung wirkt analog zum allgemeinen CP-Verhalten im Teileprogramm erst mit dem nächsten Schaltbefehl, bei Synchronaktionen sofort.

Ablauf

Starten

Die zweite Stufe der Synchronlaufüberwachung startet bei aktiver Kopplung, sobald folgende Bedingungen erfüllt sind:

- Der sollwertseitige Synchronlauf ist erreicht:
DB31, ... DBX99.4 (Synchronisation läuft) = 0
- Der istwertseitige Synchronlauf "grob"/"fein" mit der Toleranz der ersten Stufe der Synchronlaufüberwachung (siehe "Synchronlaufüberwachung Stufe 1 (CPSYNCOV, CPSYNFIP, CPSYNCOV, CPSYNFIV) (Seite 519)") ist erreicht:
DB31, ... DBX98.1 (Synchronlauf grob) = 1 / DB31, ... DBX98.0 (Synchronlauf fein) = 1

Überwachen

Solange die istwertseitige Synchronlaufdifferenz die Schwellenwerte für Positionssynchronlauf "Fein" 2 und Positionssynchronlauf "Grob" 2 nicht überschreitet, sind folgende NC/PLC-Nahtstellensignale gesetzt:

DB31, ... DBX103.4 (Synchronlauf 2 fein)

DB31, ... DBX103.5 (Synchronlauf 2 grob)

Kann die Folgeachse/-spindel aufgrund temporärer Überlastung im Bearbeitungsvorgang (z. B. Zustellvorschub beim Mehrkantschlagen zu hoch) den Vorgaben der Leitachse(n)/spindel(n) nicht mehr folgen und ist die Abweichung größer als die eingestellte Toleranz, dann wird die Überschreitung der Toleranz "fein" / "grob" jeweils durch einen löschbaren Anzeigalarm angezeigt:

Alarm 22026 "Kanal %1 Satz %2 Folgeachse/spindel %3 Synchronlauf(2): Toleranz grob überschritten"

Alarm 22025 "Kanal %1 Satz %2 Folgeachse/spindel %3 Synchronlauf(2): Toleranz fein überschritten"

Die Bearbeitung wird dadurch nicht unterbrochen.

Hinweis

Es können auch beide Alarmer gleichzeitig auftreten.

Beenden

Die zweite Stufe der Synchronlaufüberwachung wird in folgenden Fällen beendet:

- Ausschalten der Kopplung (beim Ausschaltbefehl: CPOF, CPDEL, CPLOF, CPLDEL)
- neue Synchronisationsanforderung durch:
 - CPON / CPLON / CPDO / CPLOF
 - DB31, ... DBX31.4 (Folgespindel synchronisieren) = 1
 - interne Synchronisationsanforderungen

Falls die Startbedingungen in diesen Fällen wieder erfüllt sind, wird die Überwachung neu gestartet.

- bei Koppelgesetzänderungen (CPLNUM, CPLDEN, CPLCTID) in Synchronaktionen
- Rücksetzen des sollwertseitigen Synchronlaufs aufgrund von fehlenden Freigaben für die Folgespindel (Not-Halt, Alarmreaktionen)

Beim Beenden der Überwachung werden die Signale DB31, ... DBX103.4/5 zurückgesetzt.

Randbedingungen

Ausschlussbedingungen

In folgenden Fällen erfolgt keine Überwachung:

- MD37202 bzw. MD37212 = 0.
- Geschwindigkeitskopplung ist aktiv (CPLSETVAL="CMDVEL").
- DB31, ... DBX31.5 (Synchronisation der FS sperren) = 1
- Schnellstopp der Folgeachse/-spindel oder einer der aktiven Leitachsen/-spindeln.
- SERUPRO oder Satzsuchlauf ist aktiv.

- Beim kanalweisen Einfahren, wenn die Folgeachse/-spindel oder eine aktive Leitachse/-spindel nicht real fährt.
- DB31, ... DBX63.3 (Achse-/Spindelsperre aktiv) = 1 für die Folgeachse/-spindel oder für eine der aktiven Leitachsen/-spindeln.

"Nachführen der Synchronlaufabweichung " ist aktiv

Solange die Funktion "Nachführen der Synchronlaufabweichung" aktiv ist (DB31, ... DBX31.6 = 1; siehe "Nachführen der Synchronlaufabweichung (Seite 551)"), wird die Überwachung ausgesetzt.

Synchronlaufüberwachung Stufe 2 bei den klassischen Kopplungsarten

Bei den klassischen Kopplungsarten Mitschleppen, Leitwertkopplung, Elektronisches Getriebe und Synchronspindel ist die "Synchronlaufüberwachung Stufe 2" nur über die CP-Anpasszyklen verfügbar (siehe "Anpasszyklen (Seite 534)").

Beispiel

Programmcode	Kommentar
G0 Z-300 X50	
CPDEF=(S2) CPLA=(S1) CPLNUM=2 CPLDEN=1 CPBC="FINE"	; Übersetzungsverhältnis entsprechend WZG.
M3 S2000	
CPON=(S2) CPLA=(S1) CPSYNCO2[X]=1.6 CPSYNFIP2[X]=0.8	; Satzwechsel erfolgt mit Erreichen von "Synchronlauf fein". ; Gleichzeitig wird die Synchronlaufüberwachung Stufe 2 aktiviert ; (Toleranzen: grob 1.6, fein 0.8).
X20	
G1 Z-200	; Bearbeitungssatz, Materialabtrag. ; Synchronlauf ist qualitätsbestimmend.
CPOF=(S2)	; Abwahl Kopplung, Ausschalten der Überwachung.

10.5.5.17 Reaktion auf Stopp-Signale und -kommandos (CPMBRAKE)

Das Verhalten der Folgeachse bei bestimmten Stopp-Signalen und -kommandos kann mit dem CP-Schlüsselwort CPMBRAKE festgelegt werden.

Programmierung

Syntax: CPMBRAKE [FAx] = <Wert>

Bezeichnung: **Coupling Mode Brake**

Funktionalität: CPMBRAKE ist ein bitcodiertes CP-Schlüsselwort, welches das Bremsverhalten der Folgeachse FAX bei folgenden Ereignissen festlegt:

Bit	Ereignis
0	<ul style="list-style-type: none"> • NST DB31, ... DBX4.3 (Vorschub-Halt / Spindel-Halt) steht an oder • CP-SW-Limit-Stop ist gesetzt (siehe "CP-SW-Limit-Überwachung (Seite 544)")

Hinweis:

Bit 0 ist nur beim Kopplungstyp "Freie Programmierbarkeit" (CPSETTYPE=CP) von Bedeutung.

Wert	Bedeutung
0	Die Bremse wird nicht auf die Leitachsen übertragen.
1	Die Bremse wird kontextsensitiv auf die Leitachsen übertragen.
1 - 31	reserviert

Für die Programmierung gelten folgende Regeln:

- CPMBRAKE muss in einem Satz mit CPDEF oder CPON programmiert werden (⇒ ist nur für eine nicht-aktive Kopplung programmierbar).
- Ohne explizite Programmierung von CPMBRAKE gilt bei der Definition einer Kopplung folgende Vorbelegung:
Bit 0 = 1 bei Vorlaufkopplungen mit CPSETTYPE=CP (sonst Bit 0 = 0)

Beispiele

Beispiel 1:

Programmierung	Kommentar
CPDEF=(AX5) CPLA[AX5]=(AX4) CPMBRAKE[AX5]=0 ;	Definition einer Kopplung (Leitachse AX4 zur Folgeachse AX5). NST "Vorschub-Halt/Spindel-Halt" oder "CP-SW-Limit-Stop" soll den Kopplungsverband nicht abbrem- sen.
...	

Beispiel 2:

Programmierung	Kommentar
CPDEF=(S2) CPLA[S2]=(S1) ;	Definition einer Spindelkopplung: Leitspindel S1 zur Folgespindel S2
CPON=(S2) CPMBRAKE[S2]=1 ;	Einschalten der Kopplung zur Folgespindel S2. NST "Vorschub-Halt/Spindel-Halt" oder "CP-SW-Limit-Stop" soll den Kopplungsverband abbrem- sen.
...	

10.5.5.18 Reaktion auf bestimmte NC/PLC-Nahtstellensignale (CPMVDI)

Mit dem CP-Schlüsselwort `CPMVDI` kann das Verhalten des Koppelmoduls auf bestimmte NC/PLC-Nahtstellensignale festgelegt werden.

Programmierung

Syntax: `CPMVDI [FAx] = <Wert>`

Bezeichnung: **Coupling Mode VDI Signal**

Funktionalität: `CPMVDI` ist ein bitcodiertes CP-Schlüsselwort, welches das Verhalten des Koppelmoduls der Folgeachse FAx auf bestimmte NC/PLC-Nahtstellensignale festlegt.

Zum Setzen einzelner Bits können die Bitverknüpfungsoperatoren `B_OR`, `B_AND`, `B_NOT` und `B_XOR` benutzt werden.

Bit	Bedeutung
0	Reserviert.
1	Reserviert.
2	Reserviert.
3	Die Wirksamkeit des NC/PLC-Nahtstellensignals: DB31, ... DBX1.3 (Achs-/Spindelsperre) auf die Folgeachse/-spindel ist über Bit 3 einstellbar: Bit 3 = 0 DB31, ... DBX1.3 wird für die Folgeachse/-spindel nicht wirksam. Der Zustand der Folgeachse/-spindel bezüglich der Achs-/Spindelsperre wird allein aus dem Zustand der Leitachsen/-spindeln abgeleitet. Bit 3 = 1 DB31, ... DBX1.3 wird für die Folgeachse/-spindel wirksam. Der Zustand der Leitachsen/-spindeln bezüglich der Achs-/Spindelsperre wird der Folgeachse/-spindel nicht aufgeprägt.

Hinweis:

Der Zustand Achs-/Spindelsperre der Folgeachse/-spindel wird bei gesetztem Bit 3 = 1 nur dann wirksam, wenn die Zustände Programmtest oder SERUPRO nicht aktiv sind (siehe auch Bit 5).

4	Mit Bit 4 wird die Freigabe der abhängigen Bewegungsanteile bei Wirksamkeit des NC/PLC-Nahtstellensignals DB31, ... DBX1.3 (Achs-/Spindelsperre) für die Folgeachse/-spindel festgelegt: Bit 4 = 0 Abhängige Bewegungsanteile der Leitachsen/-spindeln werden unabhängig vom Zustand der Achs-/ Spindelsperre der jeweiligen Leitachse/-spindel wirksam. Bit 4 = 1 Abhängige Bewegungsanteile der Leitachsen/-spindeln werden nur dann wirksam, wenn der Zustand der Achs-/ Spindelsperre der Leitachse/-spindel mit dem Zustand der Achs-/Spindelsperre der Folgeachse/-spindel übereinstimmt. Ist dies nicht der Fall, werden die Anteile unterdrückt.
---	--

Hinweis:

Bit 4 ist nur bei gesetztem Bit 3 von Bedeutung, also wenn für die Folgeachse/-spindel das NC/PLC-Nahtstellensignal DB31, ... DBX1.3 (Achs-/Spindelsperre) wirksam ist (siehe Bit 3 und 5).

5	Die Wirksamkeit des NC/PLC-Nahtstellensignals: DB21, ... DBX25.7 (Programmtest angewählt) bzw. DB21, ... DBX1.7 (Programmtest aktivieren) auf die Folgeachse/-spindel ist über Bit 5 einstellbar: Bit 5 = 0 DB21, ... DBX25.7 bzw. DB21, ... BX1.7 wird für die Folgeachse/-spindel nicht wirksam. Der Zustand der Folgeachse/-spindel bezüglich der Achs-/Spindelsperre wird allein aus dem Zustand der Leitachsen/-spindeln abgeleitet. Bit 5 = 1 Ist für eine Achse des Koppelmoduls der Zustand "Programmtest" wirksam, so wird für die Folgeachse/Spindel der eigene Zustand bezüglich der Achs-/Spindelsperre aktiv. Der Zustand der Leitachsen/-spindeln bezüglich
---	--

der Achs-/Spindelsperre wird der Folgeachse/-spindel nicht aufgeprägt.

Hinweis:

Für die Folgeachse/-spindel wird der Zustand Programmtest bei gesetztem Bit 5 auch dann wirksam, wenn sich die Leitachsen/-spindeln in einem anderen Zustand befinden.

- 6 Mit Bit 6 wird die Freigabe der abhängigen Bewegungsanteile bei Wirksamkeit des NC/PLC-Nahtstellensignals DB21, ... DBX25.7 (Programmtest angewählt) bzw. DB21, ... DBX1.7 (Programmtest aktivieren) für die Folgeachse/-spindel festgelegt:

Bit 6 = 0 Abhängige Bewegungsanteile der Leitachsen/-spindeln werden unabhängig vom Zustand der Achs-/Spindelsperre der jeweiligen Leitachse/-spindel wirksam.

Bit 6 = 1 Abhängige Bewegungsanteile der Leitachsen/-spindeln werden nur dann wirksam, wenn der Zustand der Achs-/Spindelsperre der Leitachse/-spindel mit dem Zustand der Achs-/Spindelsperre der Folgeachse/-spindel übereinstimmt. Ist dies nicht der Fall, werden die Anteile unterdrückt.

Hinweis:

Bit 6 ist nur bei gesetztem Bit 5 von Bedeutung, also wenn für die Folgeachse/-spindel der Zustand Programmtest wirksam ist (siehe Bit 3 und 5).

- 7 Reserviert.

- 8 Reserviert.

Hinweis

Die Achs-/Spindelsperre, die über das NC/PLC-Nahtstellensignal DB31, ... DBX1.3 für die Folgeachse/-spindel gesetzt wird, kann, wie für jede andere Achse/Spindel, durch die Zustände Programmtest (DB21, ... DBX25.7 bzw. DB21, ... DBX1.7) und SERUPRO überschrieben werden.

Wirkung der Bits 3/5 und 4/6

Die Wirksamkeit der verschiedenen Bewegungsanteile für die Folgeachse/-spindel in Abhängigkeit der zugehörigen Achs-/Spindelsperre ist in der folgenden Tabelle dargestellt:

A/S-Sperre FA	A/S-Sperre LA ₁	A/S-Sperre LA ₂	CPMVDI Bit 3/5	CPMVDI Bit 4/6	FA _{Total}	Bedeutung für FA
0	0	0	0	0	FA _{Cmd} + FA _{DEP1} + FA _{DEP2}	Reales Verfahren.
0	0	0	1	0	FA _{Cmd} + FA _{DEP1} + FA _{DEP2}	Reales Verfahren.
0	1	1	0	0	FA _{Cmd} + FA _{DEP1} + FA _{DEP2}	Simuliertes Verfahren.

A/S-Sperre FA	A/S-Sperre LA ₁	A/S-Sperre LA ₂	CPMVDI Bit 3/5	CPMVDI Bit 4/6	FA _{Total}	Bedeutung für FA
0	0	1	0	0	-	Alarm 16773 Unterschiedliche Leitachszustände bzgl. A/S-Sperre.
1	0	0	0	0	FA _{Cmd} + FA _{DEP1} + FA _{DEP2}	Reales Verfahren, Spindel-sperre der FA wirkt nicht.
1	0	1	1	0	FA _{Cmd} + FA _{DEP1} + FA _{DEP2}	Simuliertes Verfahren, Spindel-sperre der FA wirkt.
1	0	1	1	1	FA _{Cmd} + FA _{DEP2}	Simuliertes Verfahren, da Bit 4 gesetzt ist wird FA _{DEP1} unterdrückt.
0	1	1	0	1		Alarm 16694 Bit 4/6 = 1 wird nur bei Bit 3/5 = 1 unterstützt.
1	0	0	0	1		Alarm 16694 Bit 4/6 = 1 wird nur bei Bit 3/5 = 1 unterstützt.

A/S-Sperre:

Achs-/Spindelsperre

Es ist der sich ergebende, interne Zustand der Achs-/Spindelsperre gemeint. Die Spindelsperre, die über das NC/PLC-Nahtstellensignal DB31, ... DBX1.3 (Achs-/Spindelsperre) gesetzt wird, kann durch Zustände wie Programmtest (DB21, ... DBX25.7 bzw. DB21, ... DBX1.7) und SERUPRO überschrieben werden und erzeugt damit einen vom NC/PLC-Nahtstellensignal abweichenden Zustand der Achse/Spindel.

FA:

Folgeachse

LA₁:

Leitachse 1

LA₂:

Leitachse 2

FA_{Total}:

Gesamtsollwert der Folgeachse

FA_{Cmd}:

Unabhängiger Bewegungsanteil der Folgeachse

FA_{DEP1}:

Abhängiger Bewegungsanteil von der Leitachse 1

FA_{DEP2}:

Abhängiger Bewegungsanteil von der Leitachse 2

Reales Verfahren:	Reales Verfahren bedeutet, dass die Positionsbewegungen an die Lageregelung gehen.
Simuliertes Verfahren:	Simuliertes Verfahren bedeutet, dass keine Positionsbewegungen an die Lageregelung gehen. Die reale Maschinenachse bleibt stehen. Dies entspricht dem Zustand einer gesetzten Achs-/Spindelsperre bzw. Programmtest.

Hinweis

Der Zustand in den Spalten für die Leitachsen 1 und 2 gelten auch für mehrere Leitachsen/spindeln, die den gleichen Zustand bezüglich Achs-/Spindelsperre haben.

10.5.5.19 Alarmunterdrückung (CPMALARM)

Mit dem CP-Schlüsselwort `CPMALARM` können kopplungsbezogene Alarmer unterdrückt werden.

Programmierung

Syntax: `CPMALARM [FAx] = <Wert>`

Bezeichnung: **Coupling Mode Alarm**

Funktionalität: CPMALARM ist ein bitcodiertes CP-Schlüsselwort zur Unterdrückung spezieller kopplungsbezogener Alarmausgaben.
Zum Setzen einzelner Bits können die Bitverknüpfungsoperatoren B_OR, B_AND, B_NOT und B_XOR benutzt werden.

Bit	Wert	Bedeutung
0	= 1	Alarm 16772 wird unterdrückt.
1	= 1	Alarm 16773 wird unterdrückt.
2	= 1	Alarm 16774 wird unterdrückt.
3	= 1	Alarm 22012 wird unterdrückt.
4	= 1	Alarm 22013 wird unterdrückt.
5	= 1	Alarm 22014 wird unterdrückt.
6	= 1	Alarm 22015 wird unterdrückt.
7	= 1	Alarm 22016 wird unterdrückt.
8	= 1	Alarm 22025 wird unterdrückt.
9	= 1	Alarm 22026 wird unterdrückt.
10	= 1	Alarm 22040 wird bei der zyklischen Kontrolle bei bereits eingeschalteter Lageregelung bei Folge- und Leitspindeln unterdrückt.
11	= 1	Alarm 16771 wird unterdrückt.
12 - 31		Reserviert.

Die Standardwerte entsprechen den Einstellungen in den Maschinendaten:

- MD11410 \$MN_SUPPRESS_ALARM_MASK (Maske zur Unterdrückung spezieller Alarmausgaben)
- MD11415 \$MN_SUPPRESS_ALARM_MASK_2 (Maskierung von Alarmausgaben)

Beispiel

Programmcode	Kommentar
CPMALARM[X2]='H300'	; Für die Kopplung der Folgeachse X2 werden die Alarme 22025 und 22026 unterdrückt.

10.5.6 Kopplungskaskadierung

Kopplungskaskade

Koppelmodule können hintereinander geschaltet werden. Die Folgeachse/-spindel eines Koppelmoduls ist dann die Leitachse/-spindel eines anderen Koppelmoduls. Damit entsteht eine Kopplungskaskade.

Es sind auch mehrere Kopplungskaskaden hintereinander möglich. Die interne Reihenfolge der einzelnen Koppelmodule erfolgt so, dass es zu keinem Positionsversatz im Kopplungszusammenhang kommt. Dies gilt auch für eine kanalübergreifende Kaskadierung.

Beispiel:

Es werden zwei neue Koppelmodule angelegt. Für das Koppelmodul mit der Folgeachse X2 wird die Leitachse X1 definiert. Für das Koppelmodul mit der Folgeachse A2 werden die Leitachsen X2 und A1 definiert.

Programmierung

```
CPDEF=(X2) CPLA[X2]=(X1) CPDEF=(A2) CPLA[A2]=(X2) CPLA[A2]=(A1)
```

Randbedingungen

- Die Verfügbarkeit der Kaskadierung ist optionsabhängig (siehe Kapitel "Voraussetzungen (Seite 484)").
- Kaskaden zwischen Kopplungen bestehender Kopplungsfunktionen und Kopplungen der Generischen Kopplung sind nicht möglich.
- Eine Ringkopplung ist nicht erlaubt. Sie wird mit Alarm 16778 abgewiesen: "Ringkopplung bei Folgeachse FAX und Leitachse LAX nicht erlaubt" (Eine Ringkopplung entsteht, wenn eine Folgeachse wiederum Leitachse ihres eigenen Koppelmoduls oder Leitachse in einem vorangeschalteten Koppelmodul ist.)

10.5.7 Kompatibilität

10.5.7.1 Anpasszyklen

Anpasszyklen

Durch die Bereitstellung der Anpasszyklen als festem Bestandteil der NC-Software wird eine syntaktische und funktionale Kompatibilität zu den Kopplungsaufrufen der bestehenden Kopplungsarten (Mitschleppen, Leitwertkopplung, Elektronisches Getriebe und Synchronspindel) sichergestellt. D. h. solange der Maschinenhersteller / Anwender keine neuen Kopplungseigenschaften benötigt, muss er seine bisherigen Kopplungsaufrufe und davon abhängigen Applikationsanteile (z. B. in PLC-Auswertung von Kopplungssignalen) nicht modifizieren.

Zuordnung zu bestehenden Kopplungskommandos

Die Anzahl der Anpasszyklen entspricht der Anzahl der bestehenden Kopplungskommandos.
Die Zuordnung ist wie folgt:

Kopplungskommando	Anpasszyklus
TRAILON	cycle700
TRAILOF	cycle701
LEADON	cycle702
LEADOF	cycle703
COUPDEF	cycle704
COUPON	cycle705
COUPONC	cycle706
COUPOF	cycle707
COUPOFS	cycle708
COUPDEL	cycle709
COUPRES	cycle710
EGDEF	cycle711
EGON	cycle712
EGONSYN	cycle713
EGONSYNE	cycle714
EGOFC	cycle715
EGOFS	cycle716
EGDEL	cycle717

Speicherort

Die Anpasszyklen liegen im Verzeichnis "CST".

Anwenderspezifische Anpasszyklen

Bei Bedarf (funktionelle Ergänzung) kann der Anwender einen Anpasszyklus in das Verzeichnis "CMA" oder "CUS" kopieren und dort seine Änderungen/Erweiterungen vornehmen. Beim Einlesen der Anpasszyklen wird die Reihenfolge CUS → CMA → CST beachtet und die zuerst gefundene Zyklusvariante übernommen, d. h. die vom Anwender in das Verzeichnis CMA / CUS kopierten Anpasszyklen werden vorrangig ausgewählt.

Hinweis

Bei einer Hochrüstung der NC-Software wird im Verzeichnis "CST" eine Protokolldatei (Changelog-Datei) abgelegt, in der auf notwendige Änderungen der Anpasszyklen hingewiesen wird.

10.5.7.2 Kopplungstypen (CPSETTYPE)

Kopplungstypen

Wird eine Voreinstellung der Kopplungseigenschaften entsprechend den bestehenden Kopplungsarten (Mitschleppen, Leitwertkopplung, Elektronisches Getriebe und Synchronspindel) gewünscht, ist beim Anlegen des Koppelmoduls (CPON/CPLON oder CPDEF/CPLDEF) zusätzlich das Schlüsselwort CPSETTYPE zu verwenden.

Programmierung

Syntax: CPSETTYPE [FAx] = <Wert>

Bezeichnung: **Coupling Set Type**

Funktionalität: Legt die Voreinstellungen der Kopplungseigenschaften fest (Kopplungstyp).

Wert: Typ: STRING

Wertebereich:

- "CP" Freie Programmierbarkeit
- "TRAIL" Kopplungstyp "Mitschleppen"
- "LEAD" Kopplungstyp "Leitwertkopplung"
- "EG" Kopplungstyp "Elektronisches Getriebe"
- "COUP" Kopplungstyp "Synchronspindel"

Standardwert: "CP"

Beispiel:

Programmierung	Kommentar
CPLON[X2]=(X1) CPSETTYPE[X2]="LEAD"	; Legt ein Koppelmodul für die Folgeachse X2 mit der Leitachse X1 an und aktiviert das Koppelmodul. Die Kopplungseigenschaften sind so gesetzt, dass sie der bestehenden Kopplungsart Leitwertkopplung entsprechen.

Voreinstellungen

Die Voreinstellungen der programmierbaren Kopplungseigenschaften für die verschiedenen Kopplungstypen sind der folgenden Tabelle zu entnehmen:

Schlüsselwort	Kopplungstyp				
	Standard (CP)	Mitschleppen (TRAIL)	Leitwertkopplung (LEAD)	Elektronisches Getriebe (EG)	Synchronspindel (COUP)
CPDEF		-	-		
CPDEL		-	-		
CPLDEF		-	-		
CPLDEL		-	-		
CPON					
CPOF					
CPLON					
CPLOF					
CPLNUM	1.0	1.0	-	1.0	1.0
CPLDEN	1.0	1.0	-	1.0	1.0
CPLCTID	nicht gesetzt	-	0	nicht gesetzt	-
CPLSETVAL	CMDPOS	CMDPOS	CMDPOS	CMDPOS	CMDPOS
CPFRS	BCS	BCS	BCS	BCS	MCS
CPBC	NOC	NOC	NOC	FINE	IPOSTOP
CPFPOS + CPON	nicht gesetzt	-	-	nicht gesetzt	nicht gesetzt
CPFPOS + CPOF	nicht gesetzt	-	-	-	nicht gesetzt
CPFMSON	CFAST	CFAST	CCOARSE	NRGT	CFAST
CPFMON	STOP	VL ¹⁾ : STOP HL ²⁾ : CONT	VL ¹⁾ : CONT HL ²⁾ : CONT	STOP	CONT
CPFMOF	STOP	VL ¹⁾ : STOP HL ²⁾ : STOP	VL ¹⁾ : STOP HL ²⁾ : STOP	STOP	CONT
CPLPOS + CPON	nicht gesetzt	-	-	nicht gesetzt	-
CPMRESET	NONE	MD20110	MD20110	MD20110	MD20110
CPMSTART	NONE	MD20112	MD20112	MD20112	MD20112
CPMPRT	NONE	MD20112 / MD22620 ³⁾	MD20112 / MD22620 ³⁾	MD20112 / MD22620 ³⁾	MD20112 / MD22620 ³⁾
CPLINTR	0.0	0.0	0.0	0.0	0.0
CPLINSC	1.0	1.0	1.0	1.0	1.0
CPLOUTTR	0.0	0.0	0.0	0.0	0.0
CPLOUTSC	1.0	1.0	1.0	1.0	1.0

Schlüsselwort	Kopplungstyp				
	Standard (CP)	Mitschleppen (TRAIL)	Leitwertkopplung (LEAD)	Elektronisches Getriebe (EG)	Synchronspindel (COUP)
CPSYNCOV	MD37200	MD37200	MD37200	MD37200	MD37200
CPSYNFIP	MD37210	MD37210	MD37210	MD37210	MD37210
CPSYNCOV2	MD37202	MD37202	MD37202	MD37202	MD37202
CPSYNFIP2	MD37212	MD37212	MD37212	MD37212	MD37212
CPSYNCOV	MD37220	MD37220	MD37220	MD37220	MD37220
CPSYNFIV	MD37230	MD37230	MD37230	MD37230	MD37230
CPMBRAKE					
	Bit 0	1	-	-	-
CPMVDI					
	Bit 3	0	0	0	0
	Bit 4	1	1	1	1
	Bit 5	0	0	0	0
	Bit 6	1	1	1	1
CPMALARM	MD11410 MD11415	MD11410 MD11415	MD11410 MD11415	MD11410 MD11415	MD11410 MD11415
Legende: 1) Vorlauf 2) Hauptlauf 3) zusätzlich abhängig von MD22621 - nicht relevant bzw. nicht zugelassen					

Weitere Eigenschaften

Die Wertebereiche bzw. Verfügbarkeiten weiterer Eigenschaften bei gesetztem Kopplungstyp (CPSETTYPE) sind in der folgenden Tabelle zusammengefasst:

	Standard (CP)	Mitschleppen (TRAIL)	Leitwertkopplung LEAD)	Elektronisches Getriebe (EG)	Synchronspindel (COUP)
Anzahl Leitachsen	≤ 5	≤ 2	1	≤ 5	1
Achstyp Folgeachse	Achse/Spindel	Achse/Spindel	Achse/Spindel	Achse/Spindel	Spindel
Def. / Löschen Koppelmodul	CPDEF/CPDEL oder CPON/CPOF	CPON/CPOF	CPON/CPOF	CPDEF/CPDEL	CPDEF/CPDEL
Def. / Löschen Leitachse	CPLDEF/CPLDEL oder CPLON/CPLOF	CPLON/CPLOF	CPLON/CPLOF	CPLDEF/CPLDEL	CPLDEF/CPLDEL
Kaskadierung	+	+	+	+	-

	Standard (CP)	Mitschleppen (TRAIL)	Leitwertkopplung LEAD)	Elektronisches Getriebe (EG)	Synchronspindel (COUP)
Dynamik-betrachtung der Leitspindel	-	-	-	-	+
Implizite An- / Abwahl der Lageregelung ¹⁾	-	-	-	-	+

Legende:
¹⁾ siehe auch: Funktionshandbuch Erweiterungsfunktionen; Synchronspindel (S3)
 - nicht relevant bzw. nicht zugelassen

Die Verfügbarkeiten der angegebenen Eigenschaften sind abhängig von der zur Verfügung stehenden Ausführung (siehe Kapitel "Voraussetzungen (Seite 484)").

Beispiel:

Der Kopplungstyp Mitschleppen (CPSETTYPE="TRAIL") erlaubt maximal zwei Leitachsen und Kaskadierung. Dies steht aber nicht in der Grundausführung zur Verfügung, sondern setzt die Option CP-EXPERT voraus.

Randbedingungen

- CPSETTYPE ist auch in Synchronaktionen programmierbar.
- Bei gesetztem Kopplungstyp (CPSETTYPE) sind bestimmte Kopplungseigenschaften voreingestellt und nicht mehr änderbar. Das nachträgliche Ändern mittels Schlüsselwort führt zu einem Fehler und wird mit einem Alarm abgewiesen:

CPSETTYPE=	TRAIL	LEAD	EG	COUP
CPDEF	Alarm 16686	Alarm 16686		
CPDEL	Alarm 16686	Alarm 16686		
CPLDEF				
CPLDEL				
CPON			Alarm 16686	Alarm 16686
CPLON				
CPOF			Alarm 16686	Alarm 16686
CPLOF				
CPRES	Alarm 16686	Alarm 16686	Alarm 16686	
CPLNUM	Alarm 16686	Alarm 16686		
CPLDEN	Alarm 16686	Alarm 16686		
CPLCTID	Alarm 16686			Alarm 16686

CPSETTYPE=	TRAIL	LEAD	EG	COUP
CPLSETVAL	Alarm 16686 bei CMDVEL	Alarm 16686 bei CMDVEL	Alarm 16686 bei CMDVEL	
CPFRS				Alarm 16686 bei BCS
CPBC	Alarm 16686	Alarm 16686		
CPFPOS + CPON	Alarm 16686	Alarm 16686		
CPFPOS + CPOF	Alarm 16686	Alarm 16686	Alarm 16686	
CPFMSON	Alarm 16686	Alarm 16686		Alarm 16686
CPFMON	Alarm 16686	Alarm 16686 bei ADD/STOP	Alarm 16686	Alarm 16686 bei STOP
CPFMOF	Alarm 16686	Alarm 16686 bei ADD/STOP	Alarm 16686 bei ADD	Alarm 16686 bei ADD
CPLPOS + CPON	Alarm 16686	Alarm 16686		Alarm 16686
CPMRESET	Alarm 16686	Alarm 16686	Alarm 16686	Alarm 16686
CPMSTART	Alarm 16686	Alarm 16686	Alarm 16686	Alarm 16686
CPMPRT	Alarm 16686	Alarm 16686	Alarm 16686	Alarm 16686
CPMBRAKE	Alarm 16686	Alarm 16686	Alarm 16686	Alarm 16686
Anzahl Leitachsen (\sum LA)	Alarm 16672 bei \sum LA > 2	Alarm 16672 bei \sum LA > 1	Alarm 16672 bei \sum LA > 5	Alarm 16672 bei \sum LA > 1
Achstyp Folgeachse				Alarm 14092 bei Achse

10.5.7.3 Projektierte Kopplung (CPRES)

Bei gesetztem Kopplungstyp "Synchronspindel" (siehe CPSETTYPE) können anstatt der programmierten Kopplungseigenschaften die in Maschinen- und Settingdaten hinterlegten Kopplungsparameter aktiviert werden.

Literatur:

Funktionshandbuch Erweiterungsfunktionen; Synchronspindel (S3);
Kapitel "Programmierung von Synchronspindelkopplungen"

Programmierung

Syntax: CPRES= (<Folgespindel>)

Bezeichnung: **Coupling Restore**

Funktionalität: Aktiviert die projektierten Daten der Synchronspindelkopplung zur Folge-
spindel FAX.

Folgespindel: Typ: AXIS

Wertebereich: Alle im Kanal definierten Spindelnamen

Beispiel:

Programmierung	Kommentar
CPLON[S2]=(S1) CPSETTYPE[S2]="COUP"	; Legt ein Koppelmodul für die Folgespindel S2 mit der Leitspindel S1 an und aktiviert das Koppelmodul. Die Kopplungseigenschaften sind so gesetzt, dass sie der bestehenden Kopplungsart Synchronspindel entsprechen.
...	
CPRES=(S2)	; Aktiviert die projizierten Daten der Synchronspindelkopplung zur Folgespindel S2.

Randbedingungen

- CPRES ist nur zugelassen bei gesetztem Kopplungstyp "Synchronspindel" (CPSETTYPE="COUP").
- Die Anwendung von CPRES auf eine bereits aktivierte Kopplung bewirkt eine Neusynchronisation.
- Die Anwendung von CPRES auf ein nicht definiertes Koppelmodul ist ohne Wirkung.

10.5.8 Kanalübergreifende Kopplung, Achstausch

Folge- und Leitachsen müssen dem aufrufenden Kanal bekannt sein.

Folgeachse

Die Folgeachse wird beim Programmieren eines CP-Schlüsselworts im Teileprogramm je nach Achstauschprojektierung (MD30552) zum Tausch in den Kanal angefordert oder mit dem Sprachbefehl GETD in den aufrufenden Kanal geholt.

Der Achstausch der Folgeachse ist nach Aktivierung des Koppelmoduls nur noch im Kanal erlaubt. Ein Tausch vom Kanal in den Hauptlauf und umgekehrt ist also weiterhin möglich, ein Achstausch über Kanalgrenzen nicht. Es gelten die Randbedingungen und Eigenschaften für den Achstausch. Der Achstausch über Kanalachsen wird nach der Deaktivierung des Koppelmoduls wieder freigegeben.

Literatur:

Funktionshandbuch Erweiterungsfunktionen; BAGs, Kanäle, Achstausch (K5)

Leitachsen

Der Achstausch der Leitachsen ist unabhängig vom Zustand der Kopplung möglich.

10.5.9 Verhalten bei Rundachsen

Rundachsen als Folge- oder Leitachsen

Es ist möglich, Rundachsen zu einer Linearachse zu koppeln und umgekehrt. Dabei ist zu beachten, dass über das Koppelgesetz eine direkte Zuordnung von Grad zu mm erfolgt.

Beispiel:

A = Rundachse, X = Linearachse

Programmierung	Kommentar
N10 G0 A0 X0	; Verfahrbewegung: X = 0 mm, A = 0 Grad
N20 CPON=(A) CPLA[A]=(X) CPLNUM[A,X]=2	; Ein Koppelmodul für die Rundachse A mit der Linearachse X als Leitachse wird angelegt und aktiviert. Der Koppelfaktor hat den Wert 2.
N30 X100	; Verfahrbewegung: X = 100 mm, A = 200 Grad (= 100*2)

Moduloreduzierte Rundachsen als Leitachsen

Bei moduloreduzierten Rundachsen als Leitachsen wird die Eingangsgröße bei der Reduktion der Leitachse nicht mit reduziert. Als Eingangsgröße wird weiterhin die nicht reduzierte Position genommen, d. h. es wird der zurückgelegte Weg betrachtet.

Beispiel:

A = Moduloreduzierte Rundachse, X = Linearachse

Programmierung	Kommentar
N10 G0 A0 X0	; Verfahrbewegung: A = 0 Grad, X = 0 mm
N20 CPON=(X) CPLA[X]=(A) CPLNUM[X,A]=0.5	; Ein Koppelmodul für die Linearachse X mit der Rundachse A als Leitachse wird angelegt und aktiviert. Der Koppelfaktor hat den Wert 0.5.
N30 A200	; Verfahrbewegung: A = 200 Grad, X = 100 mm (= 200*0.5)

Programmierung	Kommentar
N40 A=IC(200)	; A verfährt um 200 Grad in positiver Richtung auf 400 Grad, Anzeige A = 40. X verfährt um 100 mm auf 200.
N50 A=IC(100)	; A verfährt von 40 Grad auf 140 Grad, X verfährt um weitere 50 mm auf 250.
N60 A=ACP(80)	; A verfährt in positiver Richtung auf 50 Grad, der Verfahrweg beträgt 300 Grad in positiver Richtung. X verfährt entsprechend um 150 mm in positiver Richtung. Endposition ist somit X = 400.

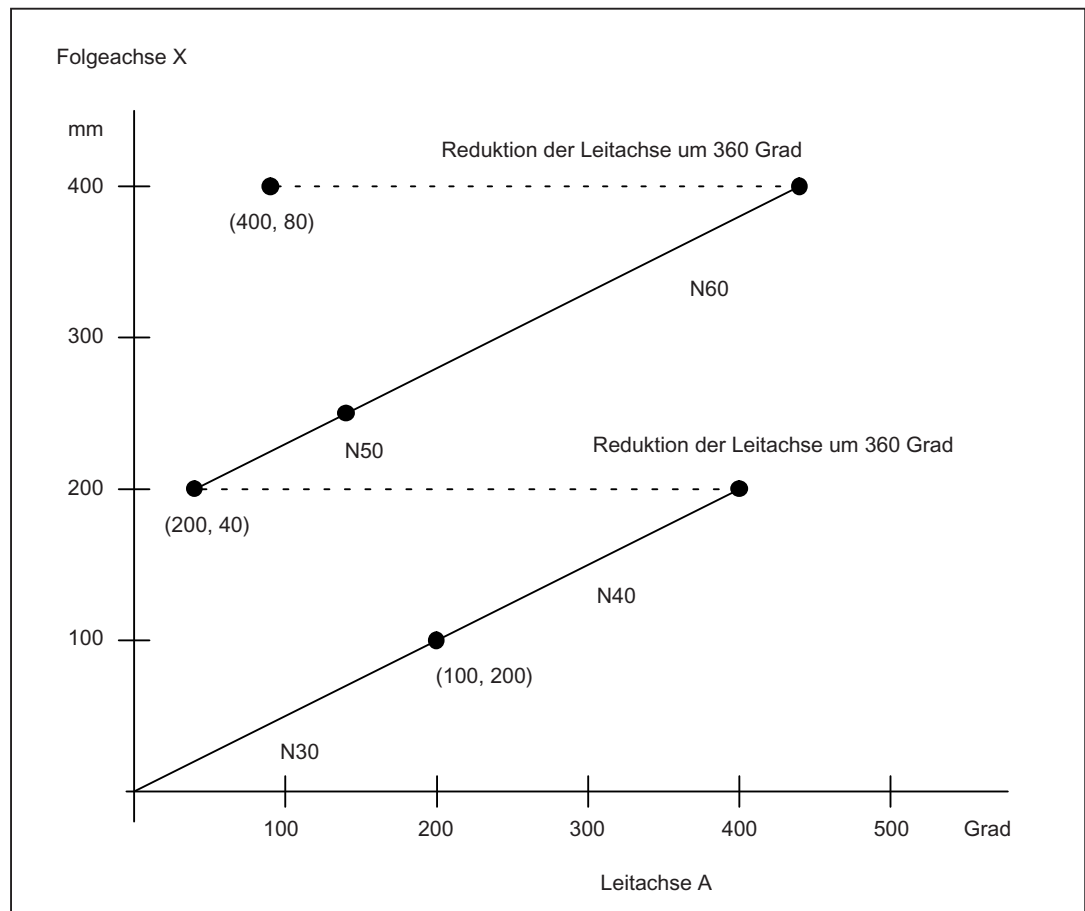


Bild 10-12 Beispiel: Modulareduzierte Rundachse an Linearachse
(...) Positionsanzeige für X, A

10.5.10 Verhalten bei POWER ON, RESET, ...

Power On

Bei Power On ist keine Kopplung aktiv. Koppelmodule sind nicht vorhanden.

RESET

Das Verhalten bei RESET ist für jedes Koppelmodul getrennt einstellbar (siehe `CPMRESET`). Die Kopplung kann aktiviert, deaktiviert oder der aktuelle Zustand erhalten bleiben.

Betriebsartenwechsel

Bei Betriebsartenwechsel bleibt die Kopplung aktiv. Lediglich in der Betriebsart JOG-REF erfolgt beim Referenzieren einer Folgeachse die Unterdrückung der Kopplung (keine Abwahl!).

Referenzpunktfahren

G74 der Folgeachse ist bei aktiver Kopplung nicht möglich. Es wird ein Alarm ausgegeben.

Wird die Betriebsart JOG-REF angewählt und die Folgeachse verfahren, so wird die Kopplung unterdrückt. Die Kopplung wird erst nach dem Verlassen der Betriebsart JOG-REF wieder ausgeführt.

SERUPRO

Der SERUPRO-Vorgang wird die Generische Kopplung simulieren und Werte zum Wiederanfahren liefern.

Bei axialen Kopplungen erfolgt die Simulation stets unter der Annahme von Sollwertkopplung, d. h. es wird bei Istwertkopplung während des SERUPRO-Vorgangs auf Sollwertkopplung umgeschaltet. Dies kann bedeuten, dass die Simulation nicht korrekt erfolgt.

Weitere Abweichungen zum realen Ablauf können durch die erhöhte Simulationsgeschwindigkeit und durch die aufgehobenen Achsdynamikbegrenzungen entstehen.

10.5.11 CP-SW-Limit-Überwachung

10.5.11.1 Funktion

Wozu dient die Funktion?

Die Funktion "CP-SW-Limit-Überwachung" verbessert das Bremsverhalten einer Folgeachse bzw. einer Folgespindel im Achsbetrieb beim Zufahren auf einen Software-Endschalter.

Vorteile:

- Das Überfahren des Software-Endschalters wird verhindert.
- Der Synchronlauf der Kopplung bleibt nach Möglichkeit erhalten.

Verfügbarkeit

Die Funktion "CP-SW-Limit-Überwachung" ist nur aktivierbar für Folgeachsen von:

- Generischen Kopplungen vom Typ "Freie Programmierbarkeit" (CPSETTYPE[FAx] = "CP")
- Kopplungen (Generische Kopplung mit CPSETTYPE[FAx] ungleich "CP", Mitschleppen, Elektronisches Getriebe, Leitwertkopplung oder Synchronspindel) mit maximal einer aktiven Leitachse/-spindel

In allen anderen Fällen (und bei Nichtaktivierung der "CP-SW-Limit-Überwachung") ist die bisherige Funktion der Software-Endschalterüberwachung wirksam (siehe Kapitel "Endschalter-Überwachung" im Funktionshandbuch Grundfunktionen, A3: Achsüberwachungen, Schutzbereiche).

Überwachung und Setzen der Bremse

Die Funktion "CP-SW-Limit-Überwachung" prüft in jedem IPO-Takt, ob die Bewegung der Folgeachse/-spindel für den folgenden IPO-Takt freigegeben werden kann, weil in jedem Fall noch rechtzeitig vor dem Software-Endschalter angehalten werden kann. Ist dies nicht der Fall, wird bei der Folgeachse ein "CP-SW-Limit-Stop" gesetzt, was ein unbedingtes richtungsspezifisches Bremsen an der Kontur bedeutet.

Übertragung der Bremse auf die Achsen

Das Setzen der Bremse bewirkt, dass der kopplungsunabhängige Bewegungsanteil (CMD- und CORR-Anteil) der Folgeachse gestoppt wird. Der kopplungsabhängige Bewegungsanteil (DEP-Anteil) der Folgeachse hingegen kann nur gebremst werden, indem die Leitachsen gebremst werden.

Bei Kopplungen (Generische Kopplung mit CPSETTYPE[FAx] ungleich "CP", Mitschleppen, Elektronisches Getriebe, Leitwertkopplung oder Synchronspindel) mit maximal einer aktiven Leitachse/-spindel wird die Bremse auch bei der (einzigen) aktiven Leitachse in Kollisionsrichtung (in Abhängigkeit vom Koppelfaktor) gesetzt.

Bei generischen Kopplungen vom Typ "Freie Programmierbarkeit" (CPSETTYPE[FAx] = "CP") erfolgt eine Übertragung der Bremse auf alle aktiven Leitachsen nur dann, wenn die Koppelleigenschaft CPMBRAKE[FAx] Bit 0 = 1 programmiert ist (siehe "Reaktion auf Stopp-Signale und -kommandos (CPMBRAKE) (Seite 526)").

Die Kollisionsrichtung ist leitachsspezifisch abhängig vom Koppelfaktor: Bei negativem Koppelfaktor dreht sich die Kollisionsrichtung um, bei Koppelfaktor "Null" erfolgt keine

Übertragung. Bei nichtlinearen Kopplungen (z. B. Kurventabelle) wird der Koppelfaktor aus der Steigung, die als lineare Näherung ermittelt wird, abgeleitet.

Hinweis

Die gesetzte Bremse einer Folgeachse kann nur Einfluß auf ihre Leitachsen haben, solange die Kopplung aktiv ist.

Bremsverhalten

Es wird unter Einhaltung des Koppelgesetzes mit der normalen Beschleunigungsrampe der Achsen in Richtung des Software-Endschalters an der Kontur gebremst.

Beim Bremsen werden der Haltezustand 75 "Bremsanforderung" und situationsabhängig die höherpriorien Haltezustände 22 "Warten: Spindelfreigabe fehlt", 12 "Warten auf Achs-/ Spindelfreigabe" bzw. 71 "Warten auf Freigabe Trafoachse" angezeigt.

Wenn die Folgeachse aufgrund des "CP-SW-Limit-Stops" steht und nicht mehr näher an den Software-Endschalter heranfahren kann, wird folgender Alarm angezeigt:

10625 "%?C{Kanal %1: %}Satz %3 Folgeachse/-spindel %2 mit CP-SW-Limit-Stop %4"

Bis zum "endgültigen" Stillstand kann so noch eine Entschärfung der Situation durch eine Bewegung der Folgeachse in die entgegengesetzte Richtung berücksichtigt werden.

Hinweis

Bei der "CP-SW-Limit-Überwachung" geht die Kopplung mit der Einhaltung des Synchronlaufs vor. Es ist daher nicht sichergestellt, dass die Achse punktgenau anhält, wenn das Koppelgesetz eingehalten werden soll. Wenn die Bremse auf die Leitachsen übertragen wird und die Interpolatoren wie gewünscht bremsen, kommt die Folgeachse vor dem Software-Endschalter zum Stillstand. Der maximale Abstand zum Software-Endschalter ergibt sich dann aus der aktuellen Maximalbeschleunigung der Folgeachse multipliziert mit dem Quadrat der Interpolationszykluszeit.

Freifahren

Das Freifahren ist Sache des Anwenders.

Zu Diagnosezwecken gibt es:

- den Haltezustand 75 "Bremsanforderung" (wenn eine Achse einen Fahrbefehl hat, aber aufgrund einer Bremsanforderung stoppt oder nicht fahren darf)
- und die Systemvariablen:
 - \$AA_BRAKE_STATE (aktueller Bremsstatus)
 - \$AA_BRAKE_CONDB (kontextsensitive Bedingungen zum Interpolator-Stopp im BKS)
 - \$AA_BRAKE_CONDM (kontextsensitive Bedingungen zum Interpolator-Stopp im MKS) einschließlich der daraus abgeleiteten BTSS-Variablen aaBrakeState, aaBrakeCondB und aaBrakeConDM.

Beispiel zum Freifahren:

1. Alarm 10625 quittieren.
2. In die Betriebsart JOG wechseln.
3. Leitachse mit den Verfahrtasten so verfahren, dass sich die Folgeachse vom Software-Endschalter weg bewegt.

10.5.11.2 Parametrierung

Aktivierung

Die Funktion "CP-SW-Limit-Überwachung" wird achsspezifisch aktiviert über das Maschinendatum:

MD30455 \$MA_MISC_FUNCTION_MASK (Achsfunktionen)

Bit	Wert	Bedeutung
11	0	CP-SW-Limit-Überwachung ist nicht aktiv.
	1	CP-SW-Limit-Überwachung ist aktiv.

10.5.11.3 Programmierung

Übertragung der Bremse auf die Leitachsen

Für generische Kopplungen vom Typ "Freie Programmierbarkeit" (CPSETTYPE[Fax] = "CP") kann über die Programmierung der Koppereigenschaft CPMBRAKE (siehe "Reaktion auf Stopp-Signale und -kommandos (CPMBRAKE) (Seite 526)") eingestellt werden, ob die durch die Funktion "CP-SW-Limit-Überwachung" ausgelöste Bremse der Folgeachse auch auf die Leitachsen übertragen werden soll.

10.5.11.4 Randbedingungen

Mögliche Fehlerquellen

Falls die Achsen nicht wie gewünscht bremsen, kann es folgenden Grund haben:

Mindestens eine Leitachse reagiert nicht auf die Bremse, weil sie ihrerseits eine Folgeachse ist (Kaskade) und die Bremse nicht auf ihre Leitachsen übertragen wird. Eine Bremse wird nur dann (rekursiv) auf die Leitachsen übertragen, wenn für die beteiligten Folgeachsen Folgendes gilt: CPSETTYPE[Fax] = "CP" und CPMBRAKE[Fax] Bit 0 = 1

Auch wenn alle Achsen wie gewünscht bremsen, können Konturverletzungen durch unterschiedliche Beschleunigungen und/oder Koppelfaktoren auftreten, wenn diese Werte nicht passend projiziert bzw. programmiert wurden.

Auch wenn alle bisherigen Voraussetzungen erfüllt sind, gibt es noch verschiedene Gründe, warum das Überfahren des Software-Endschalters durch die Folgeachse nicht verhindert werden kann:

- Die aktuelle Maximalbeschleunigung einer Leitachse ändert sich.
- Die aktuelle Maximalbeschleunigung der Folgeachse wird kleiner.
- Die Geschwindigkeit der Folgeachse ist beim Einschalten der Kopplung bzw. bei Ansprechen der CP-SW-Limit-Überwachung bereits so hoch, dass mit der aktuellen Maximalbeschleunigung nicht mehr rechtzeitig vorher angehalten werden kann.

Falls nicht mehr rechtzeitig angehalten werden kann, schlägt nach wie vor die bisherige Software-Endschalterüberwachung zu und bringt die Folgeachse beim Software-Endschalter zum Stehen.

Bremsverhalten bei Transformationen

Ist die zu bremsende Achse Ausgang einer Transformation und hat sie einen (MCS-)Fahrbefehl in Kollisionsrichtung, wird die Bremse in beide Richtungen auf alle Eingangsachsen der Transformation übertragen und es erfolgt ein Bahnstopp für diese Transformation. Dabei gibt es folgende Ausnahmen: Es gilt nicht für die unabhängigen Achsen der Transformation. Das sind bei TRANSMIT und TRAANG jeweils die in den Maschinendaten \$MC_TRAFO_AXES_IN_*[2] angegebenen Achsen. Bei den Transformationen erfolgt die Übertragung der Bremsen nur auf deren abhängige Eingangsachsen.

DRF-Verschiebung

Das Setzen eines CP-SW-Limit-Stop kann zum Abbruch einer DRF-Bewegung führen, wie das folgende Beispiel verdeutlicht:

Grundbewegung ist eine Bahnbewegung mit sehr kleiner Geschwindigkeit in positiver Richtung. Darauf wirkt eine größere DRF-Korrekturbewegung in entgegengesetzter Richtung. Damit ergibt sich ein Fahrbefehl in negativer Richtung. Werden die Bewegungen jetzt durch einen CP-SW-Limit-Stop angehalten, so führt dies bei Standardmaschinendaten zu einem Abbruch der DRF-Bewegung und nach dem Aufheben des Stopps wird nur die Bahnbewegung fortgesetzt.

NCU-Link

Bei NCU-übergreifenden Kopplungen auf Basis Lead-Link-Achse und Achscontainer muss folgende Einschränkung berücksichtigt werden:

Da die Leitachse auf einer anderen NCU interpoliert werden kann, kann die Übertragung der Bremse auf die NCU der Leitachse ein bis zwei IPO-Takte dauern.

10.5.11.5 Beispiele

Beispiel 1: Generische Kopplung vom Typ "Freie Programmierbarkeit" (CPSETTYPE[FAx] = "CP")

Projektierung:

MD26110 \$MA_POS_LIMIT_PLUS[AX2]=15 ;

Position des 1. Software-Endschalters der Folgeachse für die Verfahrbereichsgrenze in positiver Richtung

MD30455 \$MA_MISC_FUNCTION_MASK[AX2] = 'H800' ;

Bit 11 = 1 (CP-SW-Limit-Überwachung ist aktiviert)

Programmierung:

; Startpositionen von X und Y

N100 G0 X0 Y0

; Aktivierung der Kopplung FA=Y LA=X

; mit CPMBRAKE Bit 0 gesetzt (Übertragung der Bremse auf die Leitachse)

N120 CPON=Y CPSETTYPE="CP" CPLA=X CPLDEN=2 CPMBRAKE B_OR='H1'

N140 G4 F2

; Fahre Leitachse X

N200 G1 X=1000 F1000

Beispiel 2: Kopplungen mit maximal einer aktiven Leitachse/-spindel

Bei folgenden Aufrufen erfolgt die CP-SW-Limit-Überwachung einschließlich der Übertragung der Bremse auf die Leitachse, wenn MD30455 MA_MISC_FUNCTION_MASK[AX2] Bit 11 gesetzt ist:

TRAILON(Y,X,0.5) ; Definition und Einschalten der Kopplung der Mitschleppachse Y zur Leitachse X.

EGDEF(Y,X,1) ; Definition eines EG-Achsverbundes mit Sollwertkopplung von X auf Y (Folgeachse).

EGON(Y,"FINE",X,1,2) ; Einschalten der Kopplung.

10.5.12 Störverhalten

10.5.12.1 Schnellstopp

Funktion

Der Schnellstopp bewirkt ein Stillsetzen der Achse / Spindel ohne Rampe, d. h. es wird der Geschwindigkeitssollwert Null vorgegeben. Diese Vorgabe bewirkt ein Bremsen an der Stromgrenze. Die Reglerfreigabe bleibt erhalten.

Der Schnellstopp wird gesetzt bei:

- Stop A und Stop C (Safety Integrated)
- Alarmen mit Schnellstopp als projektiertem Bremsverhalten
- Erreichen eines Hardware-Endschalters und Schnellstopp als projektiertem Bremsverhalten:
MD36600 \$MA_BRAKE_MODE_CHOICE = 1

Umschaltung auf Istwertkopplung

Sobald der Schnellstopp einer Leitspindel einer generischen Kopplung gemeldet wird, werden zur Berechnung der Sollwerte die Istwerte dieser Leitspindel herangezogen.

Die Umschaltung auf Istwertkopplung erfolgt sprungfrei und ist solange aktiv, bis bei der Leitspindel sowohl die Reglerfreigabe als auch die Impulsfreigabe wieder vorhanden sind und keine Positionskorrektur mehr erfolgt. Erst wenn diese Bedingungen erfüllt sind, erfolgt die Sollwertberechnung wieder wie mit `CPLSETVAL` programmiert.

Hinweis

Ein Schnellstopp, der durch Erreichen des Hardware-Endschalters ausgelöst wurde, bewirkt keine Umschaltung auf Istwertkopplung.

Verhalten der Folgespindel

Wenn ein Schnellstopp bei einer Leitspindel erkannt wurde und die Folgespindel selbst keinen Schnellstopp durchführt, dann versucht die Folgespindel im Rahmen ihrer eingestellten Dynamik der Bewegung der Leitspindel zu folgen. Da Positionssynchronität hergestellt wird, kann es dabei zu einem Überschwingen der Folgeachse bezüglich der anzufahrenden Position kommen.

Der Beginn eines Schnellstopps bei einer Leitachse/-spindel wird NCU-übergreifend erkannt.

Hinweis

Beim Kopplungstyp Synchronspindel (`CPSETTYPE="COUP"`) wird beim einem Servo-Alarm ein gleichzeitiger Schnellstopp von Leit- und Folgespindel durchgeführt.

10.5.13 Nachführen der Synchronlaufabweichung

10.5.13.1 Grundlagen

Synchronlaufabweichung

Werkstückbearbeitungen, die sowohl an der Stirnvorder- als auch an der Stirnrückseite durchgeführt werden sollen, erfordern eine Werkstückübergabe an eine andere Werkstückaufnahmeeinrichtung (z. B. Gegenspindelfutter).

Bei der Werkstückübergabe von der Vorder- auf die Rückseitenbearbeitung kann bedingt durch das Schließen der Werkstückaufnahme ein Positionsversatz entstehen. Gründe dafür können kantige Werkstücke oder das Entstehen eines Drehimpulses beim schnellen Schließen der Werkstückaufnahmeeinrichtung (Futter) während der Bewegung sein. Erkennbar ist die Verspannung je nach Widerstandsfähigkeit des Werkstückes durch ein Ansteigen der Stromaufnahme beider an der Kopplung beteiligter Motoren und/oder durch eine Torsion des Werkstücks.

Je nach eingestellter Synchronlauf toleranz und Größe des Versatzes kann dies zum Rücksetzen folgender NC/PLC-Nahtstellensignale führen:

DB31, ... DBX98.1 (Synchronlauf grob) und/oder

DB31, ... DBX98.0 (Synchronlauf fein)

Bei der Sollwertkopplung werden die Positions- und Geschwindigkeitssollwerte entsprechend dem programmierten Koppelgesetz exakt berechnet und an die Regelungsbaugruppen ausgegeben. Geht man von gleichen Antrieben und einem starren Werkstück aus, dann wird sich eine Regelabweichung jeweils zur Hälfte als Soll-/Istwert-Differenz bei der Leit- und Folgespindel einstellen.

Funktion

Aufgabe der Funktion "Nachführen der Synchronlaufabweichung" ist es, den istwertseitig aufgezwungenen Positionsversatz zu erkennen und bei der Sollwertberechnung der Folgespindel zu korrigieren.

Voraussetzung

Voraussetzung für die Funktion ist eine über das Teil/Spannmittel geschlossene Kopplung.

Varianten

Es gibt zwei verschiedene Möglichkeiten zur Bestimmung der Synchronlaufabweichung:

1. Die Synchronlaufabweichung wird vom NC ermittelt (siehe "Synchronlaufabweichung messen (Seite 552)").
2. Der Abweichungswert ist bereits bekannt und wird vom Anwender direkt angegeben (siehe "Synchronlaufabweichung direkt angeben (Seite 555)").

In beiden Fällen wird der Abweichungswert anschließend als Korrekturwert in die Sollwertberechnung der Folgespindel eingerechnet.

Verfügbarkeit

Die Funktion "Nachführen der Synchronlaufabweichung" wurde für MKS-Kopplungen (CPFRS="MCS") entwickelt. Damit ist die Funktion auch beim Kopplungstyp "Synchronspindel" (CPSETTYPE="COUP") vorhanden.

Die Verfügbarkeit der Funktion ist wie bei anderen überlagerten Bewegungen (z. B. Differenzdrehzahl) optionsabhängig (siehe "Voraussetzungen (Seite 484)").

10.5.13.2 Synchronlaufabweichung messen

Die Steuerung misst die Differenz zwischen den Soll- und Istwertpositionen beim Synchronlauf der Folgespindel. Das Resultat ist der Korrekturwert, der in einer Systemvariablen gespeichert wird.

Voraussetzungen

Damit die Steuerung den Korrekturwert ermitteln kann, müssen folgende Voraussetzungen erfüllt sein:

- Voraussetzungen bei gesetztem Kopplungstyp "Synchronspindel" (CPSETTYPE="COUP"):
 - Die Kopplung hat genau eine Leitspindel (ist bei CPSETTYPE="COUP" erfüllt).
 - Der Koppelfaktor (Quotient aus CPLNUM und CPLDEN) ist 1 oder -1.
 - Der Folgewert wird aus der Sollposition ("DV") oder der Istposition ("AV") der Leitspindel abgeleitet.
 - Der sollwertseitige Synchronlauf muss erreicht sein:
DB31, ... DBX99.4 (Synchronisation läuft) = 0
 - Der sollwertseitige Synchronlauf darf nicht wieder abfallen.
 - Es darf keine überlagerte Bewegung (DB31, ... DBX98.4 = 0) vorhanden sein.
 - Eine Dynamikbegrenzung der Leitspindel ist erforderlich, damit eine Überforderung der Folgespindel ausgeschlossen wird.
- Voraussetzungen bei freier generischer Kopplung mit CPFRS="MCS":
 - Es werden projektierte Spindeln gekoppelt.
 - Die Kopplung hat genau eine Leitspindel.
 - Der Koppelfaktor (Quotient aus CPLNUM und CPLDEN) ist 1 oder -1.
 - Der Folgewert wird aus der Sollposition (CPLSETVAL="CMDPOS") oder der Istposition (CPLSETVAL="ACTPOS") der Leitspindel abgeleitet.
 - Der sollwertseitige Synchronlauf muss erreicht sein:
DB31, ... DBX99.4 (Synchronisation läuft) = 0
 - Der sollwertseitige Synchronlauf darf nicht wieder abfallen.
 - Es darf keine überlagerte Bewegung (DB31, ... DBX98.4 = 0) vorhanden sein.
 - Eine Dynamikbegrenzung der Leitspindel ist erforderlich, damit eine Überforderung der Folgespindel ausgeschlossen wird.

Hinweis

Dynamikbegrenzung der Leitspindel

Die Eigenschaft "Dynamikbegrenzung der Leitspindel" ist bei gesetztem Kopplungstyp "Synchronspindel" (CPSETTYPE="COUP") automatisch gegeben. Für andere Kopplungstypen liegt es in der besonderen Verantwortung des Anwenders/Maschinenherstellers, durch geeignete Maßnahmen eine dynamische Überlastung der Folgespindel zu vermeiden.

Aktivierung

Messen und Nachführen der Synchronlaufabweichung werden aktiviert, indem das folgende NC/PLC-Nahtstellensignal auf "1" gesetzt wird:

DB31, ... DBX31.6 (Synchronlauf nachführen)

Das Signal ist ausschließlich für die Folgespindel wirksam.

Hinweis

In den folgenden Fällen wird das Signal DB31, ... DBX31.6 (Synchronlauf nachführen) ignoriert:

- Achs-/Spindelsperre ist aktiv (DB31, ... DBX1.3 = 1).
- Programmtest ist angewählt.
- SERUPRO ist aktiv.

Wenn einer dieser Fälle eintritt während die Funktion "Nachführen der Synchronlaufabweichung" bereits aktiv ist, dann wird die Funktion "Nachführen der Synchronlaufabweichung" deaktiviert.

Zeitpunkt der Messung

Der Zeitpunkt der Messung und Berechnung des Korrekturwerts ist abhängig von der Einstellung von Bit 7 im Maschinendatum:

MD30455 \$MA_MISC_FUNCTION_MASK (Achsfunktionen)

Bit	Wert	Bedeutung
7	0	Der Korrekturwert wird ständig berechnet, solange das NC/PLC-Nahtstellensignal DB31, ... DBX31.6 (Synchronlauf nachführen) gesetzt ist und sollwertseitiger Synchronlauf vorhanden ist (zyklische Berechnung).
	1	Der Korrekturwert wird nur zum Zeitpunkt des Setzens des NC/PLC-Nahtstellensignals DB31, ... DBX31.6 (Synchronlauf nachführen) von 0 auf 1 berechnet (Flankenauswertung).

Hinweis

Wird mit einer zeitlichen Ausdehnung bei dem Abbau der Verspannung zwischen Leit- und Folgespindel gerechnet, dann sollte Bit 7 = 0 gesetzt sein. Das Nahtstellensignal ist dann zustandsgesteuert.

Die benötigte Zeit bei dem Abbau der Verspannung kann von verschiedenen Faktoren abhängig sein (z. B. K_V -Faktor der Lageregelung, Beschleunigungsvermögen der Motoren) und sollte experimentell ermittelt werden.

Messablauf

Das NC/PLC-Nahtstellensignal DB31, ... DBX31.6 (Synchronlauf nachführen) kann erst wirksam werden, wenn der sollwertseitige Synchronlauf erreicht ist:

DB31, ... DBX99.4 (Synchronisation läuft) = 0

Für die Dauer der Aktivierung von Signal DB31, ... DBX31.6 bzw. einmalig bei dessen aufsteigender Flanke (abhängig von der Einstellung von Bit 7 im MD30455; siehe Absatz "Zeitpunkt der Messung") werden die Istwerte der Folgespindel im IPO-Takt gelesen und die Differenz zur Sollposition ermittelt.

Die Differenz aus sollwertseitiger und istwertseitiger Synchronlaufposition ist der Korrekturwert. Dieser wird für die betreffende Folgespindel in folgender Systemvariablen gespeichert:

$\$AA_COUP_CORR[S<n>]$ (Folgespindel: Korrekturwert bei Synchronspindelkopplung)

Hinweis

Es sollte darauf geachtet werden, dass für die Dauer der Messung die Geschwindigkeit der Leit- und Folgeachse möglichst konstant ist und keine Beschleunigungssprünge auftreten.

Beispiel

Beim Einschalten der Kopplung der Synchronspindel [S2] ist zusätzlich ein Positionsversatz von 77 Grad programmiert:

```
CPON=S2 ... CPFPOS[S2]=AC(77)
```

Durch das Schließen der Werkstückaufnahmevorrichtung entsteht ein mechanischer Positionsversatz. Dadurch ergibt sich ein istwertseitiger Positionsversatz von 81 Grad.

Nach Aktivierung der Funktion "Nachführen der Synchronlaufabweichung" (DB31, ... DBX31.6 = 1) und nach Erreichen des sollwertseitigen Synchronlaufs (DB31, ... DBX99.4 = 0) wird der istwertseitige Positionsversatz ($\$VA_COUP_OFFS[S2] = 81$) mit dem sollwertseitigen Positionsversatz ($\$AA_COUP_OFFS[S2] = 77$) verglichen. Daraus resultiert ein Korrekturwert von 4 Grad, der in der Systemvariablen $\$AA_COUP_CORR[S2]$ gespeichert wird.

10.5.13.3 Synchronlaufabweichung direkt angeben

Wenn der Abweichungswert bekannt ist, dann kann dieser für die betreffende Folgespindel auch direkt in die Systemvariable \$AA_COUP_CORR geschrieben werden. Dies ist über Teileprogramm oder Synchronaktion möglich.

Hinweis

Es ist darauf zu achten, dass das Beschreiben der Systemvariablen erst nach der Herstellung der mechanischen Kopplung erfolgt. Anderenfalls kann sich beim Schließen des Futters erneut ein Versatz einstellen.

Voraussetzung

Damit das Beschreiben der Systemvariablen \$AA_COUP_CORR aus dem Teileprogramm oder Synchronaktionen möglich ist, muss für die betreffende Folgespindel mindestens einmal nach dem letzten Steuerungshochlauf eine generische MKS-Kopplung aktiviert worden sein.

10.5.13.4 Synchronlaufkorrektur

Wenn der Korrekturwert \$AA_COUP_CORR[S<n>] ungleich null ist und für die Folgespindel S<n> eine generische MKS-Kopplung aktiv war (durch CPFRS="MCS" oder CPSETTYPE="COUP"), dann wird folgendes NC/PLC-Nahtstellensignal gesetzt:

DB31, ... DBX103.0 (Synchronlaufkorrektur wird eingerechnet)

Der Korrekturwert wird im Koppelmodul in die Sollwertberechnung für die Folgespindel eingerechnet. Durch das Nachsetzen des Sollwerts um den Kopplungsversatz wird die Entspannung zwischen Leit- und Folgespindel erreicht.

Die Synchronlaufsignale werden durch Vergleich der Istwerte mit den korrigierten Sollwerten gebildet. Nach einem erfolgten Korrekturvorgang sollten die Synchronlaufsignale wieder anstehen:

DB31, ... DBX98.1 (Synchronlauf grob) und/oder

DB31, ... DBX98.0 (Synchronlauf fein)

Das Herausfahren des Korrekturwerts kann zusammen mit dem Erreichen der Synchronlaufsignale erfolgen, da die Funktion "Nachführen der Synchronlaufabweichung" in einer Verspannungssituation ja gerade dafür sorgt, dass der Synchronlauf besser erreicht wird. Das Beschleunigungsvermögen beim Herausfahren ist auf max. 10% der Maximalbeschleunigung und -geschwindigkeit reduziert.

Wenn \$AA_COUP_CORR[S<n>] vollständig herausgefahren ist, dann wird folgendes NC/PLC-Nahtstellensignal gesetzt:

DB31, ... DBX99.2 (Synchronlaufkorrektur herausgefahren)

Das gilt auch dann, wenn \$AA_COUP_CORR[S<n>] null ist und gar nichts herauszufahren ist.

Am Ende der Synchronlaufkorrektur muss das NC/PLC-Nahtstellensignal DB31, ... DBX31.6 (Synchronlauf nachführen) auf "0" zurückgesetzt werden, um die Steifigkeit der Kopplung wiederherzustellen.

Mit dem Zurücksetzen des Signals DB31, ... DBX31.6 oder dem Ausschalten der Kopplung (mit CPOF) wird der Korrekturwert nicht mehr verändert. Die Systemvariable \$AA_COUP_CORR[S<n>] liefert dann einen konstanten Wert.

Der Korrekturwert wird eingerechnet, solange er nicht durch Setzen der Systemvariablen \$AA_COUP_CORR[S<n>] auf "0" gelöscht wird, was spätestens nach Entfernung des Werkstücks aus der Spindel erfolgen muss.

Hinweis

Die Sollwertkorrektur durch die Systemvariable \$AA_COUP_CORR[S<n>] wirkt sich auf alle nachfolgenden Programmierungen der Folgespindel wie eine Positionsverschiebung aus, analog zu einer DRF-Verschiebung im MKS.

10.5.13.5 Diagnose der Synchronlaufkorrektur

Für Diagnosezwecke wird der aktuelle Wert von \$AA_COUP_CORR (Korrekturwert für Synchronlaufdifferenz nachführen) im Fenster "Service-Achse/Spindel" unter der Zeile "Positionsoffset zum Leitachse/-spindel-Sollwert" angezeigt.

Anhand der Systemvariablen \$AA_COUP_CORR_DIST (Restweg von \$AA_COUP_CORR) kann ermittelt werden, welcher Weg des Korrekturwerts noch herauszufahren ist.

10.5.13.6 Ablöschen einer Synchronlaufkorrektur

Varianten

Zum Ablöschen einer Synchronlaufkorrektur gibt es folgende Möglichkeiten:

- Beschreiben der Variablen \$AA_COUP_CORR[S<n>] mit dem Wert "0"
Die Synchronlaufkorrektur wird über eine Rampe mit reduziertem Beschleunigungsvermögen abgebaut (genauso wie beim Herausfahren eines Korrekturwerts).
- Ablöschen der Synchronlaufkorrektur über die PLC
Mit der aufsteigenden Flanke des NC/PLC-Nahtstellensignals: DB31, ... DBX31.7 (Synchronlaufkorrektur löschen) wird die Variable \$AA_COUP_CORR[S<n>] auf Null gesetzt und die Synchronlaufkorrektur wie folgt gelöscht:
 - Falls die Spindel im Drehzahlsteuerbetrieb ist, wird die Korrekturbewegung gestoppt. Danach wird die bestehende Synchronlaufkorrektur in die Sollposition übernommen.
 - In allen anderen Fällen wird die herausgefahrte Synchronlaufkorrektur genauso wieder zurückgefahren, wie es beim Setzen der Variablen \$AA_COUP_CORR[S<n>] auf Null geschieht.

Voraussetzungen

Voraussetzung für das Ablöschen einer Synchronlaufkorrektur ist, dass der Korrekturwert nicht gerade berechnet wird (siehe Kapitel "Synchronlaufabweichung messen (Seite 552)").

Ende des Löschvorgangs

Wenn der Löschvorgang beendet ist, wird das folgende NC/PLC-Nahtstellensignal gesetzt:

DB31, ... DBX99.2 (Synchronlaufkorrektur herausgefahren)

Wenn darüber hinaus das NC/PLC-Nahtstellensignal DB31, ... DBX103.0 (Synchronlaufkorrektur wird eingerechnet) zurückgesetzt ist, dann kann auch das NC/PLC-Nahtstellensignal DB31, ... DBX31.7 (Synchronlaufkorrektur löschen) zurückgesetzt werden.

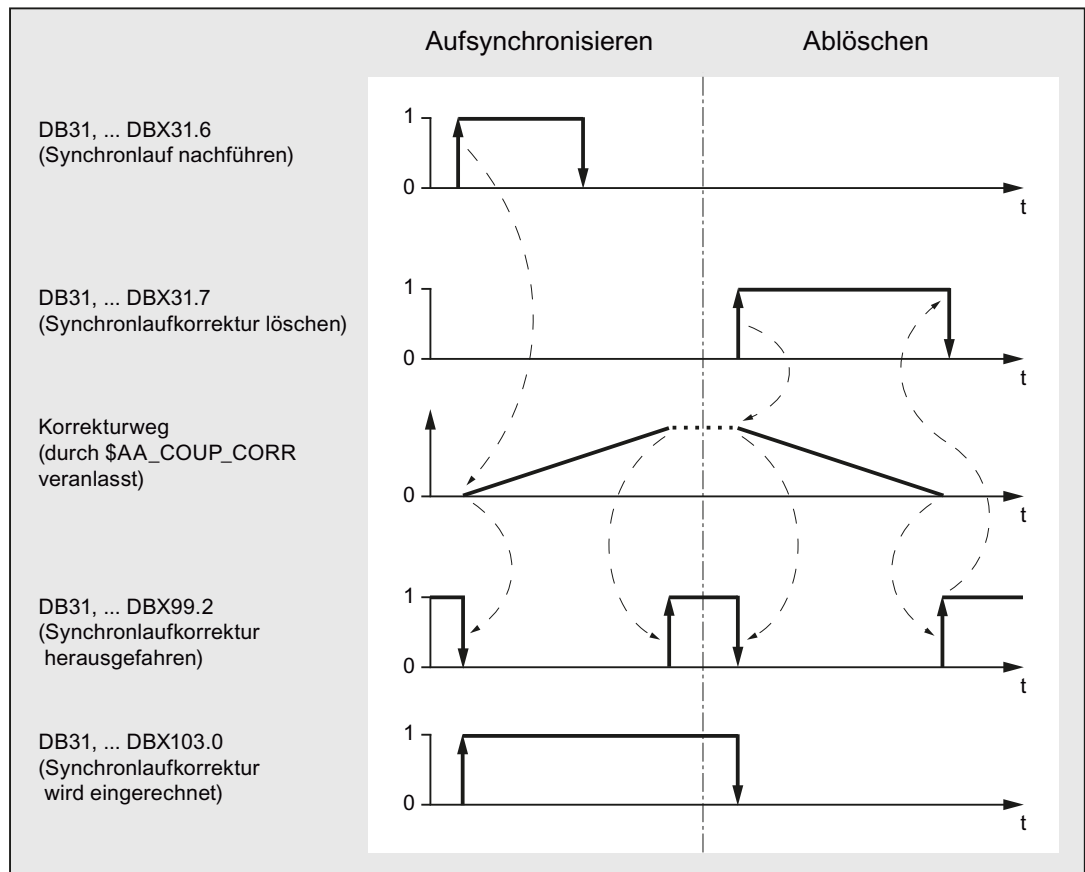


Bild 10-13 Zeitdiagramm zum Aufsynchronisieren und Ablöschen einer Synchronlaufkorrektur

Hinweis

Solange der Korrekturweg nicht komplett herausgefahren wurde und das NC/PLC-Nahtstellensignal DB31, ... DBX31.7 (Synchronlaufkorrektur löschen) nicht zurückgesetzt ist, hat das Schreiben von `$AA_COUP_CORR[S<n>]` keine Wirkung.

10.5.13.7 Randbedingungen

Mehrere Folgespindeln

Hat eine Leitspindel mehrere Folgespindeln, so kann jede Folgespindel getrennt voneinander mit dem axialen NC/PLC-Nahtstellensignal DB31, ... DBX31.6 (Synchronlauf nachführen) behandelt werden.

Beschreiben der Variablen \$AA_COUP_CORR

Das Beschreiben der Systemvariablen \$AA_COUP_CORR aus dem Teileprogramm oder Synchronaktionen wirkt erst dann, wenn mindestens einmal eine generische MKS-Kopplung zu der betreffenden Achse/Spindel aktiviert worden ist.

Korrekturwert

Wenn der Korrekturwert \$AA_COUP_CORR sowohl über Teileprogramm / Synchronaktion geschrieben als auch aufgrund Aktivierung der Funktion "Nachführen der Synchronlaufabweichung" (DB31, ... DBX31.6 = 1) ermittelt wird, dann ist immer das zuletzt aufgetretene Ereignis wirksam.

Ablöschen der Synchronlaufkorrektur

Während des Ablöschens der Synchronlaufkorrektur hat das Schreiben des Korrekturwerts \$AA_COUP_CORR (über Teileprogramm oder Synchronaktion bzw. bei der Berechnung) keine Wirkung.

Verhalten bei Kanal-/BAG-Reset

Die Synchronlaufkorrektur wird bei Kanal-/BAG-Reset nicht gelöscht und bleibt erhalten.

Verhalten bei Referenzpunktfahren und Nullmarkensynchronisation

Beim Referenzpunktfahren und Nullmarkensynchronisation bei Spindeln wird die Synchronlaufkorrektur automatisch abgelöscht.

Während des Referenzpunktfahrens/Nullmarkensynchronisation darf die Systemvariable \$AA_COUP_CORR nicht gesetzt werden und somit auch keine Messung der Synchronlaufabweichung erfolgen.

Desweiteren muss das Herausfahren der Synchronlaufkorrektur beendet sein, bevor das Referenzpunktfahren/Nullmarkensynchronisation gestartet wird.

Verhalten bei einer Unterbrechung

Bei einer Unterbrechung (z. B. Not-Halt) wird die Synchronlaufkorrektur automatisch abgelöscht, die bestehende Synchronlaufkorrektur in die Sollposition übernommen und das NC/PLC-Nahtstellensignal DB31, ... DBX99.2 (Synchronlaufkorrektur herausgefahren) gesetzt.

Wenn nach dem Aufheben der Unterbrechung die generische MKS-Kopplung noch aktiv ist und das NC/PLC-Nahtstellensignal DB31, ... DBX31.6 (Synchronlauf nachführen) gesetzt ist, gilt Folgendes:

- Wenn das Signal pegelgetriggert ist (MD30455 \$MA_MISC_FUNCTION_MASK, Bit 7 = 0), wird die Synchronlaufabweichung gemessen und nach \$AA_COUP_CORR geschrieben, die Sollwerte werden entsprechend korrigiert.
- Wenn das Signal flankengetriggert ist (MD30455 \$MA_MISC_FUNCTION_MASK, Bit 7 = 1) und die Synchronlaufabweichung gemessen werden soll, muss eine erneute aufsteigende Flanke des NC/PLC-Nahtstellensignal DB31, ... DBX31.6 (Synchronlauf nachführen) gebracht werden.

10.5.14 Beispiele

10.5.14.1 Programmierbeispiele

Direktes Ein- / Ausschalten mit einer Leitachse

Ein Koppelmodul mit der Folgeachse X2 und der Leitachse X1 wird angelegt und aktiviert. Der Koppelfaktor ist 2.

```
CPON=(X2) CPLA[X2]=(X1) CPLNUM[X2,X1]=2
...
CPOF=(X2) ; Mit CPOF wird die Kopplung deaktiviert und das
angelegte Koppelmodul gelöscht.
```

Direktes Ein- / Ausschalten mit zwei Leitachsen

Ein Koppelmodul mit der Folgeachse X2 und den Leitachsen X1 und Z wird angelegt und aktiviert. Der Koppelfaktor bzgl. Leitachse X1 ist 2, der Koppelfaktor bzgl. Leitachse Z ist 3.

```
CPON=(X2) CPLA=(X1) CPLNUM[X2,X1]=2 CPLA=(Z) CPLNUM[X2,Z]=3
...
CPOF=(X2) ; Mit CPOF wird die Kopplung deaktiviert und das
angelegte Koppelmodul gelöscht.
```

Selektives Ausschalten mit zwei Leitachsen

Ein Koppelmodul mit der Folgeachse Y und den Leitachsen X und Z wird angelegt und aktiviert. Der Koppelfaktor bzgl. Leitachse X ist 2, der Koppelfaktor bzgl. Leitachse Z ist 1.2.

```
CPON=(X2) CPLA[X2]=(X1) CPLNUM[X2,X1]=2 CPLA[X2]=(Z) CPLNUM[X2,Z]=1.2
...
CPOF=(X2) CPLA[X1]=(Z) ; Mit CPOF wird die Kopplung mit der Leitachse
Z deaktiviert, die Kopplung mit der Leitachse
X1 bleibt aktiv. Das angelegte Koppelmodul
bleibt angelegt.
```

Selektives Ein- / Ausschalten mit drei Leitachsen

Ein Koppelmodul mit der Folgeachse X2 und den Leitachsen X1, Z und A wird angelegt.

N10 CPDEF=(X2) CPLA[X2]=(X1) CPLA[X2]=(Z) CPLA[X2]=(A)	
N20 CPON=(X2)	; Alle Leitachsen werden aktiv, d. h. alle liefern gemäß Koppelgesetz einen Positionsbeitrag (Kopplungsbeitrag) zur Achse X2.
N30 CPOF=(X2)	; Alle Leitachsen werden deaktiviert.
N40 CPLON[X2]=(X1)	; Leitachse X1 wird aktiviert, nur diese liefert einen Kopplungsbeitrag. Die Leitachsen Z und A bleiben deaktiviert.
N50 CPLON[X2]=(A)	; Leitachse X1 bleibt aktiv, Leitachse A wird aktiviert, X1 und A liefern Kopplungsbeiträge (→ selektives Einschalten wirkt additiv, der Zustand der anderen Leitachsen bleibt erhalten).
N60 CPLON[X2]=(Z) CPLOF[X2]=(A)	; Leitachse Z wird aktiviert, Leitachse A deaktiviert. Es sind jetzt die Leitachsen X1 und Z aktiv.
N70 CPLOF[X2]=(X1)	; Leitachse X1 wird deaktiviert. Leitachse Z bleibt aktiv.

Definition / Löschen eines Koppelmoduls

Mit CPDEF wird ein Koppelmodul mit der Folgeachse X2 und den Leitachsen X1 und Z angelegt. Die Kopplung wird nicht aktiviert. Die Folgeachse X2 befolgt das Koppelgesetz nicht!

```
CPDEF=(X2) CPLA[X2]=(X1) CPLNUM[X2,X1]=2 CPLA[X2]=(Z) CPLNUM[X2,Z]=3
...
```

Die Aktivierung kann mit CPON erfolgen, die Deaktivierung mit CPOF.

Nach erfolgter Deaktivierung der Koppelbeziehung zu allen Leitachsen kann das Koppelobjekt gelöscht werden. Der reservierte Speicher wird freigegeben:

```
CPDEL=(X2)
```

10.5.14.2 Anpasszyklus anpassen

Ziel

Das Mitschleppen im Maschinenkoordinatensystem soll mit dem bestehenden Kopplungskommando TRAILON möglich sein. Dazu wird der Anpasszyklus für TRAILON mit der Kopplungseigenschaft "Koordinatenbezug" (CPFRS) ergänzt.

Vorgehen

1. Anpasszyklus cycle700 aus dem Verzeichnis "CST" in das Verzeichnis "CMA" kopieren.
2. Anpasszyklus cycle700 um folgenden Eintrag ergänzen:
CPFRS[_FA]="MCS"
3. Änderung im Zyklus kommentieren (z. B. durch Anwenderversionsnummer und Änderungsdatum).
4. Zyklus speichern.

```

;*CHANGE : 07.01.02.00 Mar 08, 2006
; $PATH=/_N_CST_DIR/_N_CYCLE700_SPF
; ANWENDER V1.1 Mar 22, 2006
; classic TRAILON(FA,LA,Factor)
PROC CYCLE700(AXIS_CPF=NO_AXIS, AXIS_CPL=NO_AXIS, REAL_CPLF=1) IPTRLOCK SBLOF DISPLOF ICYCOF
;* IF _CPLF==0 GOTOF _CPOF
CPLON[_CPF]=(_CPL) CPSETTYPE[_CPF]="TRAIL" CPLNUM[_CPF,_CPL]=(_CPLF) CPFRS[_CPF]="MCS"
RET
_CPOF:
IF ($P_TECCYCLE==TRUE)
  IF ($AA_CPSETTYPE[_CPF]<>"TRAIL") GOTOF _EXIT
ELSE
  IF ($PA_CPSETTYPE[_CPF]<>"TRAIL") GOTOF _EXIT
ENDIF
IF _CPL==NO_AXIS
  CPSETTYPE[_CPF]="TRAIL" CPOF=(_CPF)
ELSE
  CPSETTYPE[_CPF]="TRAIL" CPLOF[_CPF]=(_CPL)
ENDIF
_EXIT: RET

;* SysFehler 500918 deshalb IF Satz auskommentiert

```

Bild 10-14 Cycle700 nach Anpassung. Die Änderungen sind durch einen farbigen Balken gekennzeichnet.

10.6 Dynamikverhalten der Folgeachse

10.6.1 Parametrierte Dynamikgrenzen

Die Dynamik der Folgeachse wird durch folgende Maschinendatenwerte begrenzt:

MD32000 \$MA_MAX_AX_VELO (Maximale Achsgeschwindigkeit)

MD32300 \$MA_MAX_AX_ACCEL (Maximale Achsbeschleunigung)

10.6.2 Programmierte Dynamikgrenzen

10.6.2.1 Programmierung (VELOLIMA, ACCLIMA)

Dynamikgrenzen reduzieren oder überhöhen

Die über MD32000 und MD32300 angegebenen Dynamikbegrenzungen der Folgeachse (FA) können mit Sprachbefehlen aus dem **Teileprogramm** heraus reduziert oder überhöht werden:

Befehl	Bedeutung
VELOLIMA [FA]	Maximale Achsgeschwindigkeit reduzieren oder überhöhen
ACCLIMA [FA]	Maximale Achsbeschleunigung reduzieren oder überhöhen

Die bei der Programmierung von VELOLIMA [FA] und ACCLIMA [FA] angegebenen Werte sind **Prozentwerte**. Sie legen fest, mit welchem Anteil die parametrisierten Dynamikgrenzen (MD32000 und MD32300) berücksichtigt werden sollen:

Wertebereich	Bedeutung
$1 \leq \text{Wert} < 100$	bewirkt eine Reduzierung der Dynamikgrenze
$100 < \text{Wert} \leq 200$	bewirkt eine Überhöhung der Dynamikgrenze

Die Dynamikgrenzen für die Geschwindigkeit und die Beschleunigung der Folgeachse berechnen sich dann wie folgt:

Maximale Achsgeschwindigkeit = MD32000 \$MA_MAX_AX_VELO * VELOLIMA[FA]

Maximale Achsbeschleunigung = MD32300 \$MA_MAX_AX_ACCEL * ACCLIMA[FA]

Hinweis

Die Reduzierung / Überhöhung wirkt auf die gesamte Dynamik der Achse, also auf die Summe der Achsanteile aus Überlagerung und Kopplung.

Programmierung in Synchronaktionen

Die Möglichkeit der Programmierung von VELOLIMA [FA] und ACCLIMA [FA] in **Synchronaktionen** ist vom Kopplungstyp abhängig:

Kopplungstyp	Teileprogramm	Synchronaktionen
Tangentiale Nachführung	x	
Mitschleppen	x	x
Leitwertkopplung	x	x
Elektronisches Getriebe	x	
Synchronspindel	x	
Generische Kopplung	x	x

Synchronisation zwischen Folge- und Leitachsen

Durch das eingestellte Beschleunigungsverhalten und die eingestellten Dynamikkorrekturen wird die Zeitdauer für die Synchronisation zwischen Folge- und Leitachsen bei Beschleunigungsvorgängen wie folgt verändert:

Dynamikkorrektur	Wirkung
Dynamikreduktion	Verlängert die Synchronlaufdifferenz. Die Überwachung von Leit- zu Folgewert kann eine längere Zeit den zulässigen Bereich überschreiten.
Dynamiküberhöhung	Verkürzt die Synchronlaufdifferenz. Die Überwachung von Leit- zu Folgewert kann eine kürzere Zeit den zulässigen Bereich überschreiten.


Hinweis

Der Anwender hat die technologische Synchronisation zwischen Bearbeitung und Synchronlaufdifferenz wieder herzustellen.

Beschleunigungsmodus

Für die Folgeachse steht nur **BRISKA**, d. h. sprungförmige Achsbeschleunigung, zur Verfügung. Die Beschleunigungsmodi **SOFTA** und **DRIVEA** sind für die beschriebenen Folgeachsen nicht verfügbar.

Darüber hinaus besteht die Möglichkeit, den Lageregler als PI-Regler zu konfigurieren.

 VORSICHT
Anwendungsfehler
Diese Möglichkeit ist nur in Verbindung mit Servo-Trace und mit regeltechnischem Fachwissen einsetzbar.

Literatur:

- Inbetriebnahmehandbuch CNC: NC, PLC, Antrieb
- Funktionshandbuch Grundfunktionen; Geschwindigkeiten, Soll-/Istwertsysteme, Regelung (G2)

POWER ON

Bei POWER ON werden die Werte für **VELOLIMA** und **ACCLIMA** auf 100% initialisiert.

Betriebsartenwechsel

Beim Betriebsartenwechsel AUTO → JOG behalten die Dynamikkorrekturen ihre Gültigkeit.

RESET

Die Gültigkeiten der Dynamikkorrekturen (VELOLIMA und ACCLIMA) nach RESET sind abhängig von der Einstellung im kanalspezifischen Maschinendatum:

MD22410 \$MC_F_VALUES_ACTIVE_AFTER_RESET (F-Funktion über RESET hinaus wirksam)

Wert	Bedeutung
0	Die Werte für VELOLIMA[FA] und ACCLIMA[FA] werden nach RESET auf 100% gesetzt.
1	Die zuletzt programmierten Werte für VELOLIMA[FA] und ACCLIMA[FA] sind auch nach RESET wirksam.

Dieses Verhalten gilt auch für Dynamikkorrekturen, die durch statische Synchronaktionen gesetzt wurden. Soll dies trotz MD22410 = 0 nicht der Fall sein, muss per IDS-Synchronaktion ein erneutes bzw. ständiges Schreiben der Dynamikkorrektur erfolgen.

Literatur:

Funktionshandbuch Synchronaktionen

10.6.2.2 Beispiele

Elektronisches Getriebe

Die Achse 4 wird über eine EG-Kopplung an X gekoppelt. Das Beschleunigungsvermögen der Folgeachse wird auf 70% der maximalen Beschleunigung begrenzt. Die maximal zulässige Geschwindigkeit wird auf 50% der maximalen Geschwindigkeit begrenzt. Nach dem Einschalten wird die maximal zulässige Geschwindigkeit wieder auf 100% gesetzt.

```

...
N120 ACCLIMA[AX4]=70
N130 VELOLIMA[AX4]=50 ; Reduzierte Geschwindigkeit
N150 EGON (AX4, "FINE", X, 1, 2)
N200 VELOLIMA[AX4]=100 ; Volle Geschwindigkeit
...
    
```

Leitwertkopplung

Die Achse 4 wird über eine Leitwertkopplung an X gekoppelt. Das Beschleunigungsvermögen der Folgeachse wird auf 80% der maximalen Beschleunigung begrenzt.

```

...
N120 ACCLIMA[AX4]=80 ; 80%
N130 LEADON (AX4, X, 2) ; Kopplung einschalten
...
    
```


Leitwertkopplung mit Synchronaktion

Die Achse 4 wird mittels Leitwertkopplung an X gekoppelt. Das Beschleunigungsverhalten wird per statische Synchronaktion 2 ab Position 100 auf 80% begrenzt.

```

...
N120 IDS=2 WHENEVER $AA_IM[AX4] > 100 DO ACCLIMA[AX4]=80
N130 LEADON (AX4, X, 2)
...

```

10.6.2.3 Systemvariablen

Für die Achstypen Geometrieachse, Kanalachse, Maschinenachse und Spindel stehen im Teileprogramm und in Synchronaktionen folgende lesbaren Systemvariablen zur Verfügung:

Bezeichner	Datentyp	Bedeutung	Einheit
Im Vorlauf			
\$PA_ACCLIMA[n]	REAL	Mit ACCLIMA[AX] gesetzte Beschleunigungskorrektur	%
\$PA_VELOLIMA[n]	REAL	Mit VELOLIMA[AX] gesetzte Geschwindigkeitskorrektur	%
Im Hauptlauf			
\$AA_ACCLIMA[n]	REAL	Mit ACCLIMA[AX] gesetzte Beschleunigungskorrektur	%
\$AA_VELOLIMA[n]	REAL	Mit VELOLIMA[AX] gesetzte Geschwindigkeitskorrektur	%

Hinweis

Das Lesen der Hauptlaufvariablen löst implizit Vorlaufstopp aus.

10.7 Allgemeine Randbedingungen

Hinweis

Antrieboptimierung

An einem Antriebsgerät SINAMICS S120 können maximal 3 Antriebe gleichzeitig optimiert bzw. vermessen (Drehzahlregleroptimierung/Funktionsgenerator) werden. Es wird daher dringend empfohlen bei einer Kopplung mit mehr als 3 gleichzeitig gekoppelten Antrieben, diese auf mehrere Antriebsgeräte zu verteilen.

Hinweis

Satzsuchlauf bei aktiver Kopplung

Es wird empfohlen bei aktiver Kopplung für einen Satzsuchlauf ausschließlich den Suchlauftyp 5, "Satzsuchlauf über Programmtest" (SERUPRO), zu verwenden.

10.8 Datenlisten

10.8.1 Maschinendaten

10.8.1.1 NC-spezifische Maschinendaten

Nummer	Bezeichner: \$MN_	Beschreibung
11410	SUPPRESS_ALARM_MASK	Maske zur Unterdrückung spezieller Alarmausgaben
11415	SUPPRESS_ALARM_MASK_2	Maskierung von Alarmausgaben
11660	NUM_EG	Anzahl der möglichen Elektronischen Getriebe
11750	NCK_LEAD_FUNCTION_MASK	Funktionen zur Leitwertkopplung
11752	NCK_TRAIL_FUNCTION_MASK	Funktionen zum Mitschleppen
18400	MM_NUM_CURVE_TABS	Anzahl Kurventabellen (SRAM)
18402	MM_NUM_CURVE_SEGMENTS	Anzahl Kurvensegmente (SRAM)
18403	MM_NUM_CURVE_SEG_LIN	Anzahl linearer Kurvensegmente (SRAM)
18404	MM_NUM_CURVE_POLYNOMS	Anzahl Kurventabellenpolynome (SRAM)
18406	MM_NUM_CURVE_TABS_DRAM	Anzahl Kurventabellen im DRAM
18408	MM_NUM_CURVE_SEGMENTS_DRAM	Anzahl Kurvensegmente im DRAM
18409	MM_NUM_CURVE_SEG_LIN_DRAM	Anzahl linearer Kurvensegmente (DRAM)
18410	MM_NUM_CURVE_POLYNOMS_DRAM	Anzahl Kurventabellenpolynome im DRAM
18450	MM_NUM_CP_MODULES	Maximal zulässige Anzahl der CP-Koppelmodule
18452	MM_NUM_CP_MODUL_LEAD	Maximal zulässige Anzahl der CP-Leitwerte

10.8.1.2 Kanal-spezifische Maschinendaten

Nummer	Bezeichner: \$MC_	Beschreibung
20110	RESET_MODE_MASK	Festlegung der Steuerungsgrundstellung nach Hochlauf und RESET/Teileprogrammende
20112	START_MODE_MASK	Festlegung der Steuerungsgrundstellung nach Hochlauf und bei RESET
22620	START_MODE_MASK_PRT	Festlegung der Steuerungs-Grundstellung bei speziellen Starts
22621	ENABLE_START_MODE_MASK_PRT	Freischaltung von MD22620
20900	CTAB_ENABLE_NO_LEADMOTION	Kurventabellen mit Sprung der Folgeachse
20905	CTAB_DEFAULT_MEMORY_TYPE	Default Speichertyp für Kurventabellen
21300	COUPLE_AXIS_1	Projektierung Synchronspindel paar

10.8.1.3 Achs-/Spindel-spezifische Maschinendaten

Nummer	Bezeichner: \$MA_	Beschreibung
30130	CTRL_OUT_TYPE	Ausgabeart des Sollwerts
30132	IS_VIRTUAL_AX	Achse ist virtuelle Achse
30455	MISC_FUNCTION_MASK	Achsfunktionen
35040	SPIND_ACTIVE_AFTER_RESET	Eigener Spindel-RESET
37160	LEAD_FUNCTION_MASK	Funktionen zur Leitwertkopplung
37200	COUPLE_POS_TOL_COARSE	Schwellwert für "Synchronlauf grob"
37202	COUPLE_POS_TOL_COARSE_2	Zweite Synchronlaufüberwachung: Schwellwert für "Synchronlauf grob"
37210	COUPLE_POS_TOL_FINE	Schwellwert für "Synchronlauf fein"
37212	COUPLE_POS_TOL_FINE_2	Zweite Synchronlaufüberwachung: Schwellwert für "Synchronlauf fein"
37220	COUPLE_VELO_TOL_COARSE	Geschwindigkeitstoleranz "grob"
37230	COUPLE_VELO_TOL_FINE	Geschwindigkeitstoleranz "fein"
37500	ESR_REACTION	Reaktions-Definition beim Erweiterten Stillsetzen und Rückziehen
37550	EG_VEL_WARNING	Schwellwert Geschwindigkeits-Warnschwelle
37560	EG_ACC_TOL	Schwellwert für das Signal "Achse beschleunigt"

10.8.2 Settingdaten

10.8.2.1 Achs-/Spindel-spezifische Settingdaten

Nummer	Bezeichner: \$SC_	Beschreibung
43100	LEAD_TYPE	Festlegung des Leitwerttyps
43102	LEAD_OFFSET_IN_POS	Verschiebung des Leitwerts bei Kopplung zu dieser Achse
43104	LEAD_SCALE_IN_POS	Skalierung des Leitwerts bei Kopplung zu dieser Achse
43106	LEAD_OFFSET_OUT_POS	Verschiebung des Funktionswerts der Kurventabelle
43108	LEAD_SCALE_OUT_POS	Skalierung des Funktionswerts der Kurventabelle

10.8.3 Systemvariablen

Elektronisches Getriebe EG und Leitwertkopplung

Bezeichner	Bedeutung
\$AA_EG_ACTIVE	Koppelung für die Leitachse b ist aktiv, d. h. eingeschaltet
\$AA_EG_AX	Name für n-te Leitachse
\$AA_EG_DENOM	Nenner des Koppelfaktors für die Leitachse b
\$AA_EG_NUMERA	Zähler des Koppelfaktors für die Leitachse b
\$AA_EG_NUMLA	Anzahl der mit EGDEF definierten Leitachsen
\$AA_EG_SYN	Synchronposition der Leitachse b
\$AA_EG_SYNFA	Synchronposition der Folgeachse a
\$AA_EG_TYPE	Art der Kopplung für die Leitachse b
\$AA_IN_SYNC[FA]	Synchronisationszustand der Folgeachse
\$AA_LEAD_P	aktueller Leitwert - Position (moduloreduziert).
\$AA_LEAD_P_TURN	aktueller Leitwert - Positionsanteil, der durch Moduloreduktion verloren geht.
\$AA_LEAD_SP	simulierter Leitwert - Position in MKS
\$AA_LEAD_SV	simulierter Leitwert - Geschwindigkeit
\$AA_LEAD_V	aktueller Leitwert - Geschwindigkeit
\$AA_SYNC	Kopplungszustand der Folgeachse bei Leitwertkopplung.
\$P_EG_BC	Satzwechselkriterium für EG-Aktivierungsaufrufe: EGON, EGONSYN. WAITC => sofort, Synchronlauf fein bzw. grob und sollwertseitiger Synchronlauf.
\$VA_EG_SYNCDIFF	Synchronlaufdifferenz

Generische Kopplung

Bezeichner	Bedeutung
\$AA_ACCLIMA	Mit ACCLIMA gesetzte Beschleunigungskorrektur (HL)
\$AA_COUP_ACT	Kopplungstyp einer Folgeachse/-spindel
\$AA_COUP_CORR	Folgespindel - Korrekturwert für Synchronlaufdifferenz nachführen
\$AA_COUP_CORR_DIST	Restweg von \$AA_COUP_CORR
\$AA_COUP_OFFS	Sollwertseitiger Positions-Offset
\$AA_CPACTFA	Achsname der aktiven Folgeachse
\$AA_CPACTLA	Achsname der aktiven Leitachse
\$AA_CPBC	Satzwechselkriterium
\$AA_CPDEFLA	Achsname der definierten Leitachse
\$AA_CPFACT	Kopplungsart der Folgeachse/-spindel
\$AA_CPFCDPT	Achssollposition über alle Koppelanteile
\$AA_CPFCDVT	Achssollgeschwindigkeit über alle Koppelanteile
\$AA_CPFMOF	Verhalten der Folgeachse beim Ausschalten
\$AA_CPFMON	Verhalten der Folgeachse beim Einschalten
\$AA_CPFMSON	Synchronisationsmodus

Bezeichner	Bedeutung
\$AA_CPFERS	Bezugssystem der Kopplung
\$AA_CPLCMDP	Achssollpositionsanteil der Leitachse
\$AA_CPLCMDV	Achssollgeschwindigkeitsanteil der Leitachse
\$AA_CPLCTID	Tabellenummer der aktiven Kurventabelle
\$AA_CPLDEN	Nenner des Koppelfaktors
\$AA_CPLNUM	Zähler des Koppelfaktors
\$AA_CPLSETVAL	Kopplungsbezug der Leitachse
\$AA_CPLSTATE	Zustand der Kopplung
\$AA_CPSYNCOV	Schwellwert für den Positionssynchronlauf "Grob" (HL)
\$AA_CPSYNCOV	Schwellwert für den Geschwindigkeitssynchronlauf "Grob" (HL)
\$AA_CPSYNFIP	Schwellwert für den Positionssynchronlauf "Fein" (HL)
\$AA_CPSYNFIV	Schwellwert für den Geschwindigkeitssynchronlauf "Fein" (HL)
\$AA_CPLINSC	Skalierfaktor für den Eingangswert der Leitachse (HL)
\$AA_CPLINTR	Verschiebewert für den Eingangswert der Leitachse (HL)
\$AA_CPLOUTSC	Skalierfaktor für den Ausgangswert der Kopplung (HL)
\$AA_CPLOUTTR	Verschiebewert für den Ausgangswert der Kopplung (HL)
\$AA_CPLTYPE	Kopplungsart
\$AA_CPMRESET	Zustand der Kopplung nach RESET
\$AA_CPMSTART	Zustand der Kopplung nach Programmstart
\$AA_CPNACTFA	Anzahl der aktiven Folgeachsen
\$AA_CPNACTLA	Anzahl der aktiven Leitachsen
\$AA_CPNDFFLA	Anzahl der definierten Leitachsen
\$AA_CPSETTYPE	Voreingestellter Kopplungstyp
\$AA_EG_ACTIVE	Koppelung für die Leitachse b ist aktiv
\$AA_EG_AX	Name für n-te Leitachse
\$AA_EG_BC	Satzwechselkriterium
\$AA_EG_DENOM	Nenner des Koppelfaktors
\$AA_EG_NUMERA	Zähler des Koppelfaktors
\$AA_EG_NUM_LA	Anzahl der mit EGDEF definierten Leitachsen
\$AA_EG_SYN	Synchronposition der Leitachse
\$AA_EG_SYNFA	Synchronposition der Folgeachse
\$AA_EG_TYPE	Art der Kopplung
\$AA_IN_SYNC[FA]	Synchronisationszustand der Folgeachse
\$AA_JERKLIMA	Mit JERKLIMA gesetzte Ruckkorrektur (HL)
\$AA_LEAD_SP	Simulierter Leitwert - Position bei LEAD
\$AA_LEAD_SV	Simulierter Leitwert - Geschwindigkeit bei LEAD
\$AA_LEAD_P_TURN	Aktueller Leitwert - Positionsanteil, der durch Moduloreduktion verloren geht
\$AA_LEAD_P	Aktueller Leitwert - Position (moduloreduziert)
\$AA_LEAD_V	Aktueller Leitwert - Geschwindigkeit
\$AA_SYNC	Kopplungszustand der Folgeachse
\$AA_SYNCDIFF[FA]	Synchronlaufdifferenz sollwertseitig

Bezeichner	Bedeutung
\$AA_SYNCDIFF_STAT[FA]	Status der Synchronlaufdifferenz sollwertseitig
\$AA_TYP/TYPE	Achstyp
\$AA_VELOLIMA	Mit VELOLIMA gesetzte Geschwindigkeitskorrektur (HL)
\$PA_ACCLIMA	Mit ACCLIMA gesetzte Beschleunigungskorrektur (VL)
\$PA_CPFACT	Kopplungsart der Folgeachse/-spindel
\$PA_CPFPOSSTAT	Gültigkeit der Synchron- und Stopp-Position
\$PA_CPSYNCOV	Schwellwert für den Positionssynchronlauf "Grob" (VL)
\$PA_CPSYNCOV	Schwellwert für den Geschwindigkeitssynchronlauf "Grob" (VL)
\$PA_CPSYNFIP	Schwellwert für den Positionssynchronlauf "Fein" (VL)
\$PA_CPSYNFIV	Schwellwert für den Geschwindigkeitssynchronlauf "Fein" (VL)
\$PA_CPLINSC	Skalierfaktor für den Eingangswert der Leitachse (VL)
\$PA_CPLINTR	Verschiebewert für den Eingangswert der Leitachse (VL)
\$PA_CPLOUTSC	Skalierfaktor für den Ausgangswert der Kopplung (VL)
\$PA_CPLOUTTR	Verschiebewert für den Ausgangswert der Kopplung (VL)
\$PA_CPSETTYPE	Voreingestellter Kopplungstyp
\$PA_JERKLIMA	Mit JERKLIMA gesetzte Ruckkorrektur (VL)
\$PA_VELOLIMA	Mit VELOLIMA gesetzte Geschwindigkeitskorrektur (VL)
\$VA_COUP_OFFS[S2]	Istwertseitiger Positions-Offset der Synchronspindel
\$VA_EG_SYNCDIFF	Synchronlaufdifferenz
\$VA_EG_SYNCDIFF_S	Synchronlaufdifferenz mit Vorzeichen
\$VA_SYNCDIFF[FA]	Synchronlaufdifferenz istwertseitig
\$VA_SYNCDIFF_STAT[FA]	Status der Synchronlaufdifferenz
\$P_COUP_OFFS[S2]	Programmierter Positions-Offset der Synchronspindel
\$P_EG_BC	Satzwechselkriterium

Folgeachsdynamik

Bezeichner	Bedeutung
\$AA_ACCLIMA	Mit ACCLIMA gesetzte Beschleunigungskorrektur im Hauptlauf
\$AA_VELOLIMA	Mit VELOLIMA gesetzte Geschwindigkeitskorrektur im Hauptlauf
\$PA_ACCLIMA	Mit ACCLIMA gesetzte Beschleunigungskorrektur im Vorlauf
\$PA_VELOLIMA	Mit VELOLIMA gesetzte Geschwindigkeitskorrektur im Vorlauf

10.8.4 Signale

10.8.4.1 Signale an Achse/Spindel

Signalname	SINUMERIK 840D sl	SINUMERIK 828D
Vorschub-Override	DB31, ... DBX0.0-7	DB380x.DBB0
Achsensperre	DB31, ... DBX1.3	DB380x.DBX1.3
Reglerfreigabe	DB31, ... DBX2.1	DB380x.DBX2.1
Handrad aktivieren	DB31, ... DBX4.0-2	DB380x.DBX4.0/1
Vorschub Halt	DB31, ... DBX4.3	DB380x.DBX4.3
Freigabe Folgeachsüberlagerung	DB31, ... DBX26.4	DB380x.DBX5002.4
Folgespindel synchronisieren	DB31, ... DBX31.4	DB380x.DBX5007.4
Synchronisation sperren	DB31, ... DBX31.5	DB380x.DBX5007.5
Synchronlauf nachführen	DB31, ... DBX31.6	DB380x.DBX5007.6
Synchronlaufkorrektur löschen	DB31, ... DBX31.7	DB380x.DBX5007.7

10.8.4.2 Signale von Achse/Spindel

Signalname	SINUMERIK 840D sl	SINUMERIK 828D
Begrenzung DiffDrehzahl	DB31, ... DBX83.1	DB390x.DBX2001.1
Spindel im Sollbereich DiffDrehzahl	DB31, ... DBX83.5	DB390x.DBX2001.5
Drehzahlgrenze überschritten Summendrehzahl	DB31, ... DBX83.6	DB390x.DBX2001.6
Istdrehrichtung rechts Summendrehzahl	DB31, ... DBX83.7	DB390x.DBX2001.7
Synchronbetrieb	DB31, ... DBX84.4	DB390x.DBX2002.4
Synchronlauf fein	DB31, ... DBX98.0	DB390x.DBX5002.0
Synchronlauf grob	DB31, ... DBX98.1	DB390x.DBX5002.1
Istwertkopplung	DB31, ... DBX98.2	DB390x.DBX5002.2
Überlagerte Bewegung	DB31, ... DBX98.4	DB390x.DBX5002.4
Geschwindigkeitswarnschwelle	DB31, ... DBX98.5	DB390x.DBX5002.5
Beschleunigungswarnschwelle	DB31, ... DBX98.6	DB390x.DBX5002.6
Leitspindel aktiv	DB31, ... DBX99.0	DB390x.DBX5003.0
Folgespindel aktiv	DB31, ... DBX99.1	DB390x.DBX5003.1
Synchronlaufkorrektur herausgefahren	DB31, ... DBX99.2	DB390x.DBX5003.2
Folgeachse beschleunigt	DB31, ... DBX99.3	DB390x.DBX5003.3
Synchronisation läuft	DB31, ... DBX99.4	DB390x.DBX5003.4
Maximale Geschwindigkeit erreicht	DB31, ... DBX99.5	DB390x.DBX5003.5
Maximale Beschleunigung erreicht	DB31, ... DBX99.6	DB390x.DBX5003.6
Synchronlaufkorrektur wird eingerechnet	DB31, ... DBX103.0	DB390x.DBX5007.0
Synchronlauf 2 fein	DB31, ... DBX103.4	DB390x.DBX5007.4
Synchronlauf 2 grob	DB31, ... DBX103.5	DB390x.DBX5007.5

R3: Erweitertes Stillsetzen und Rückziehen

11.1 Kurzbeschreibung

Die Funktion "Erweitertes Stillsetzen und Rückziehen", im weiteren Verlauf mit ESR bezeichnet, bietet die Möglichkeit, in Fehlersituationen prozessabhängig flexibel zu reagieren:

- **Erweitertes Stillsetzen**
Soweit es die spezifische Fehlersituation erlaubt, werden alle für das Erweiterte Stillsetzen freigegebenen Achsen geordnet stillgesetzt.
- **Rückziehen**
Das sich im Eingriff befindliche Werkzeug wird schnellstmöglich vom Werkstück zurückgezogen.
- **Generatorbetrieb (SINAMICS Antriebsfunktion "Vdc-Regelung")**
Bei Unterschreitung eines parametrierbaren Wertes der Zwischenkreisspannung, z.B. bei Ausfall der Netzspannung, wird die für das Rückziehen benötigte elektrische Energie durch Rückspeisen der Bremsenergie eines dafür vorgesehenen Antriebs (Generatorbetrieb) erzeugt.

ESR und Aktive Kopplungen

Während des Stillsetzens und Rückziehens bleiben aktive Kopplungen für eine parametrierbare Zeit erhalten.

Rückziehen auf einer Bahn

Neben dem rein axialen Rückziehen, kann auch eine Gerade als Rückzugsbahn programmiert werden.

11.2 Steuerungsgeführtes ESR - nur 840D sl

11.2.1 Grundlagen

Durch die Funktion "Erweitertes Stillsetzen/Rückziehen (ESR)" erfolgt ein definiertes, zeitlich verzögertes Stillsetzen sowie Rückziehen von dafür freigegebenen Achsen zum schnellen Trennen von Werkzeug und Werkstück in bestimmten programmierbaren Systemzuständen.

Damit die für die Rückzugsbewegung erforderliche Energie in den beteiligten Antrieben auch bei Netzausfall zur Verfügung steht, kann über die SINAMICS S120 Antriebsfunktion "Vdc-Regelung" eine oder mehrere Antriebe als "Generatoren" parametrierbar werden. Ihre

Bewegungsenergie wird bei Netzausfall dann zur Aufrechterhaltung der Zwischenkreisspannung verwendet, um die NC-geführte Rückzugsbewegung zu ermöglichen.

Hinweis

Ausführliche Informationen zur SINAMICS S120 Antriebsfunktion "Vdc-Regelung" finden sich in:

Literatur

Funktionshandbuch SINAMICS S120 Antriebsfunktionen

NC-geführte Reaktionen

Die Funktion stellt folgende NC-geführte Reaktionen zur Verfügung:

- "Erweitertes Stillsetzen":
Programmierbares definiertes, zeitlich verzögertes, bahnbezogenes Stillsetzen von Verfahrbewegungen
- "Rückziehen":
Schnellstmögliches Herausfahren aus der Bearbeitungsebene in eine sichere Rückzugsposition zum Trennen von Werkzeug und Werkstück

Die Reaktionen sind unabhängig voneinander. Rückzugsvorgänge und temporäre Fortführung von vor dem Stillsetzen bestehenden Achskopplungen können so projektiert werden, dass sie zeitlich parallel ausgeführt werden. Dabei kann ein Antrieb im Generatorbetrieb (Vdc-Regelung) die Spannung des Zwischenkreises aufrechterhalten.

Zusammenwirken der NC-geführten Reaktionen

Die NC-geführten Reaktionen werden über die **kanalspezifische** Systemvariable \$AC_ESR_TRIGGER ausgelöst.

Mit \$AC_ESR_TRIGGER wird ein interpolatorisches Stillsetzen auf der Bahn bzw. Kontur möglich. Das NC-geführte Rückziehen wird dazu zeitsynchron durch die Rückzugsachsen im Kanal durchgeführt.

Für ESR freigegebene Achsen dürfen nur genau einem Kanal zugeordnet sein und nicht zwischen Kanälen getauscht werden.

Bei NC-geführtem Stillsetzen wird eine bestehende Verfahrbewegung ebenso wie eine aktive elektronische Kopplung über eine einstellbare Zeit (MD21380 \$MC_ESR_DELAY_TIME1) aufrecht erhalten, auch wenn ein Alarm mit Bewegungsstopp ansteht. Nach Ablauf der parametrisierten Zeit wird auf der programmierten Bahn bis zum Stillstand abgebremst.

Um auch außerhalb der Betriebsart AUTOMATIK ein Rückziehen durchführen zu können, wird die Auslösung dieser Funktionalität an die Systemvariable \$AC_ESR_TRIGGER geknüpft. Durch \$AC_ESR_TRIGGER ausgelöstes Rückziehen ist gegen Mehrfach-Rückzug verriegelt.

11.2.2 Antriebsautarke Reaktionen

Generatorbetrieb

Der Generatorbetrieb ist eine Antriebsfunktion. Das Antriebsgerät SINAMICS S120 bietet dabei die Möglichkeit, über die Funktion "Vdc-Regelung" den Zwischenkreisverband auf Unterspannung zu überwachen. Bei Unterschreitung eines einstellbaren Spannungswerts wird dann der dafür vorgesehene Antrieb in den Generatorbetrieb geschaltet. Dabei wird die Bewegungsenergie des Antriebs zur Stützung der Zwischenkreisspannung verwendet, um NC-seitig ein geordnetes Stillsetzen und Rückziehen der in Bewegung befindlichen Achsen zu ermöglichen.

Hinweis

Ist eine Achse zum Zeitpunkt, da sie in den Generatorbetrieb geschaltet wird, in Lageregelung, können zusätzliche Alarmer auftreten.

Literatur

Ausführliche Informationen zur SINAMICS S120 Antriebsfunktion "Vdc-Regelung" siehe:
Funktionshandbuch SINAMICS S120 Antriebsfunktionen

11.2.3 Überwachung der Zwischenkreisspannung

Grenzwerte der Zwischenkreisspannung

Der Zwischenkreis (ZK) wird auf die im nachfolgenden Bild dargestellten Grenzwerte überwacht:

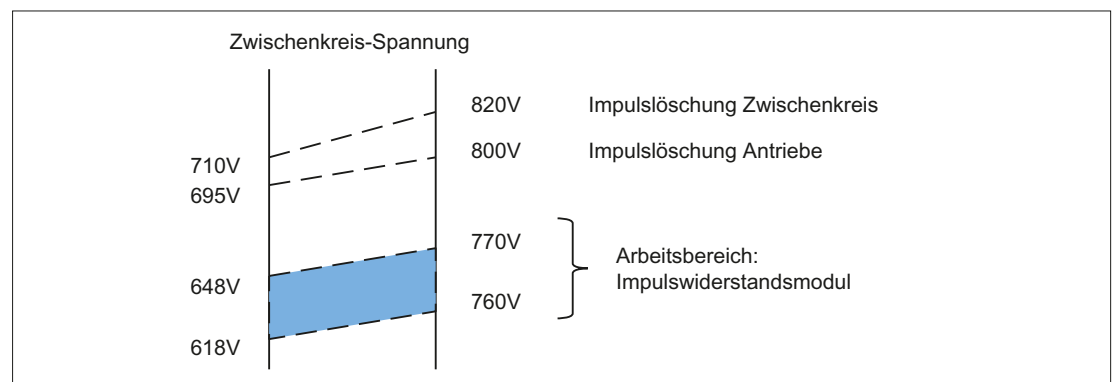


Bild 11-1 Grenzwerte der Zwischenkreisspannung

Bei bestimmten Spannungspegeln werden die Impulse der Antriebe und des Zwischenkreises gelöscht, was zum Austrudeln der Antriebe führt. Wird dieses Verhalten nicht gewünscht, kann durch Verwendung eines Widerstandsmoduls die überschüssige Energie abgeleitet werden.

Der Arbeitsbereich des Widerstandsmoduls (im Bild hervorgehoben dargestellt) liegt unterhalb der kritischen Spannungspegel ist.

Hinweis

Die Impulsleistung des Widerstandsmoduls ist größer als die Leistung der Einspeisung.

Überwachung der Zwischenkreisunterspannung

Die Zwischenkreisspannung kann auf einen im Antrieb parametrierbaren Grenzwert überwacht werden:

- p1248 (Zwischenkreisspannung Spannungsschwelle unten)

Bei Unterschreitung des Grenzwertes wird im Meldewort (MELDW) des PROFIdrive-Telegramms folgendes Signal gesetzt:

- MELDW.Bit4 = 1 (VDC_min Regler aktiv (Uzk < p1248))

Das Signal kann in der NC zum Auslösen der ESR-Reaktionen verwendet werden.

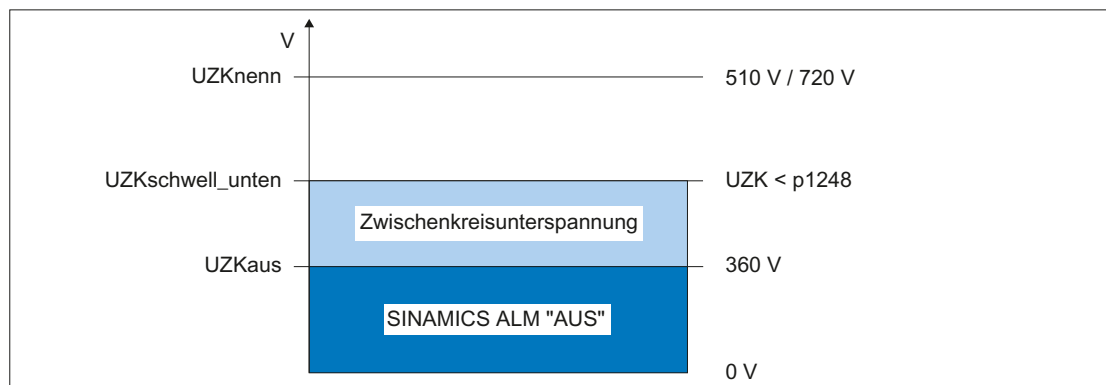


Bild 11-2 Überwachung der Zwischenkreisspannung

11.2.4 Erweitertes Stillsetzen

Achsspezifische Parametrierung

Das NC-geführte Erweiterte Stillsetzen wird für eine Maschinenachse aktiviert über Maschinendatum:

MD37500 \$MA_ESR_REACTION[<Achse>] = 22

Hinweis

Bahnachsen

Ist das NC-geführte Erweiterte Stillsetzen für eine **Bahnachse** parametrier, überträgt sich das entsprechende Verhalten auch auf alle anderen **Bahnachsen** des Kanals.

Leit- und Folgeachsen

Eine Folgeachse eines Elektronischen Getriebes folgt während des Erweiterten Stillsetzens den Leitachsen entsprechend dem parametrieren/programmierten Bewegungsgesetz. Demzufolge ist für die Folgeachse während der Zeiten MD21380 \$MC_DELAY_TIME1 und MD21381 \$MC_ESR_DELAY_TIME2 kein eigenständiges Bremsen möglich. MD21380 \$MC_DELAY_TIME1MD21381 \$MC_ESR_DELAY_TIME2Voraussetzung für ein bestimmungsgemäß ablaufendes ESR ist, dass alle erforderlichen Freigabesignale gesetzt sind und bleiben.

Kanalspezifische Parametrierung

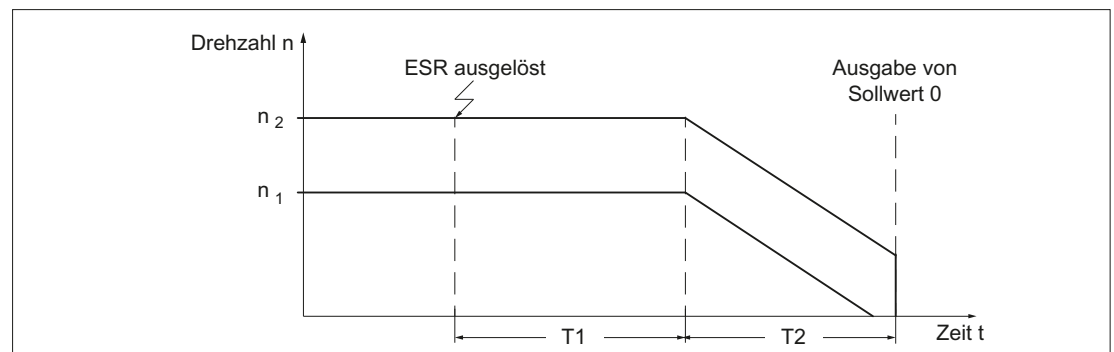
Der zeitliche Ablauf innerhalb eines Kanals wird über folgende Maschinendaten parametrier:

- MD21380 \$MC_ESR_DELAY_TIME1[<Achse>] (Verzögerungszeit ESR-Achsen)
- MD21381 \$MC_ESR_DELAY_TIME2[<Achse>] (ESR-Zeit für interpolatorisches Bremsen)

Zeitlicher Ablauf

Nach dem Auslösen von ESR, wird die Achse weiter mit der aktuellen Istdrehzahl verfahren. Nach Ablauf der Verzögerungszeit (..._TIME1) wird die Achse interpolatorisch abgebremst.

Für das interpolatorische Bremsen steht maximal die dafür eingestellte Zeit (..._TIME2) zur Verfügung. Nach Ablauf dieser Zeit wird für die Achse Sollwert 0 ausgegeben und Nachfuhrbetrieb aktiviert.



- T1 MD21380 \$MC_ESR_DELAY_TIME1 (Verzögerungszeit ESR-Achsen)
T2 MD21381 \$MC_ESR_DELAY_TIME2 (ESR-Zeit für interpolatorisches Bremsen)
 n_1 Istdrehzahl, bei der innerhalb von T2 bis zum Stillstand gebremst werden konnte
 n_2 Istdrehzahl, bei der innerhalb von T2 nicht bis zum Stillstand gebremst werden konnte \Rightarrow Schnellbremsung

Hinweis

Die Summe aus T1 und T2 sollte aus Sicherheitsgründen einen Maximalwert von ca. 1 sec nicht überschreiten.

Voraussetzung

Damit ESR durchgeführt wird, muss mindestens eine der an der Verfahrbewegung beteiligten Achsen als NC-geführte Rückzugs- oder Stillsetzachse parametrierbar sein: MD37500
\$MA_ESR_REACTION > 20

Für Achsen, die nicht als NC-geführte Rückzugs- oder Stillsetzachse parametrierbar sind, erfolgt sofort mit Beginn des Erweiterten Stillsetzens (\$AC_ESR_TRIGGER = 1) Schnellbremsen mit anschließendem Nachführen.

Die Bearbeitung aller Kommandos, insbesondere die, die ein Stoppen der Achsen zur Folge haben (z. B. Reset, Stop, Stopall), sowie der Standard-Alarmreaktionen STOPBYALARM und NOREADY, wird um die Summe der parametrierbaren Zeiten verzögert:

Verzögerungszeit = MD21380 \$MC_ESR_DELAY_TIME1 + MD21381
\$MC_ESR_DELAY_TIME2

Ein NC-geführtes Stillsetzen wirkt auch in Zusammenhang mit der Funktion "Elektronisches Getriebe" (siehe Kapitel "Elektronisches Getriebe (EG) (Seite 455)"). Es beinhaltet die (selektive) Umschaltung des elektronischen Getriebes auf Istwertkopplung bei gestörten Leitachsen sowie die Aufrechterhaltung der Verfahrbewegung und der Freigaben während der Verzögerungszeit für ESR-Achsen: MD21380 \$MC_ESR_DELAY_TIME1

11.2.5 Rückziehen

11.2.5.1 Funktion

Das Schnellabheben auf die mit POLF definierte Position wird mit dem modalen Befehl LFPOS programmiert.

Die mit LFPOS bzw. POLF parametrierbare Rückzugsbewegung der mit POLFMASK oder POLFMLIN programmierten Achsen ersetzt die für diese Achsen im NC-Programm programmierten Verfahrbewegungen.

Bezüglich der Rückzugsbewegung gilt:

- Die mit POLFMASK angegebenen Achsen werden unabhängig voneinander auf die mit POLF angegebenen Positionen gefahren.
- Die mit POLFMLIN angegebenen Achsen werden im **linearen** Zusammenhang auf die mit POLF angegebenen Positionen gefahren.

Die Erweiterte Rückzugsbewegung (d. h. durch \$AC_ESR_TRIGGER ausgelöstes LIFTFAST/LFPOS) ist **nicht unterbrechbar** und kann nur durch Not-Aus vorzeitig beendet werden.

Die Geschwindigkeits- und Beschleunigungsbegrenzungen der am Rückzug beteiligten Achsen werden während der Rückzugsbewegung überwacht. Die Abhebewegung erfolgt mit BRISK, d. h. ohne Ruckbegrenzung.

Auslösen des Rückzugs

Ausgelöst wird die Rückzugsbewegung `LIFTFAST` in einem Kanal, wenn in diesem Kanal eine Achse vorhanden ist, für welche die Rückzugsbewegung freigegeben ist (`$AA_ESR_ENABLE == 1`) und der Trigger für den Schnellrückzug (`$AC_ESR_TRIGGER = 1`) gesetzt wird.

Voraussetzung

- Im NC-Programm muss eine Rückzugsposition programmiert worden sein.
- Für die Rückzugsbewegung müssen die Freigabesignale gesetzt sein und bleiben.

11.2.5.2 Parametrierung: Maschinendaten

Aktivierung

Mit dem Maschinendatum wird das NC-geführte Rückziehen für eine Achse aktiviert:

MD37500 `$MA_ESR_REACTION = 21`

Verzögerungszeiten

Für den Rückzug können zwei verschiedene Rückzugszeiten parametrierbar werden:

- MD21380 `$MC_ESR_DELAY_TIME1 = <Verzögerungszeit: Verfahrbewegung>`
- MD21381 `$MC_ESR_DELAY_TIME2 = <Verzögerungszeit: Interpolatorisches Bremsen>`

Für Durchführung der Rückzugsbewegung steht die Summe der beiden Verzögerungszeiten zur Verfügung. Nach Ablauf der Gesamt-Verzögerungszeit wird für die Rückzugachsen Schnellbremsung mit Sollwert 0 mit anschließendem Umschalten in den Nachführbetrieb ausgelöst.

Spiegeln

Mit dem Maschinendatum wird eingestellt, ob bei aktiver Funktion "Spiegeln" (DB31, ... DBX97.2) die Rückzugsrichtung beim "Schnellabheben von der Kontur" (`LIFTFAST`) gespiegelt werden soll:

MD21202 `$MC_LIFTFAST_WITH_MIRROR = TRUE`

Die Spiegelung der Rückzugsrichtung bezieht sich nur auf die Richtungskomponenten senkrecht zur Werkzeugrichtung.

Stoppverhalten

Das Stoppverhalten der Abhebbewegung wird kanalspezifisch mit folgendem Maschinendatum eingestellt:

MD21204 `$MC_LIFTFAST_STOP_COND, <Bit> = <Wert>`

Bit	Wert	Beschreibung
0		Reaktion auf das achsspezifische NC/PLC-Nahtstellensignal DB31, ... DBB4.3 (Vorschub Halt) bzw. kontextsensitiver Interpolatorstopp.
	0	Stopp der Rückzugsbewegung bei achsspezifischem Vorschub-Halt bzw. kontextsensitivem Interpolatorstopp
	1	kein Stopp der Rückzugsbewegung bei achsspezifischem Vorschub-Halt bzw. kontextsensitivem Interpolatorstopp

Bit	Wert	Beschreibung
1		Reaktion auf das kanalspezifische NC/PLC-Nahtstellensignal DB21, ... DBB6.0 (Vorschubsperrere)
	0	Stopp der Rückzugbewegung bei Vorschub-Sperre im Kanal
	1	kein Stopp der Rückzugbewegung bei Vorschub-Sperre im Kanal

Hinweis**Einfluss der Nahtstellensignale**

- Das achsspezifische NC/PLC-Nahtstellensignal DB31, ... DBB4.3 (Vorschub Halt) beeinflusst die gesamte Rückzugbewegung. Alle mit `POLFMASK` und `POLFMLIN` definierten Achsbewegungen werden gestoppt.
- Das kanalspezifische NC/PLC-Nahtstellensignal DB21, ... DBB7.3 (NC-Stop), hat keinen Einfluss auf die Rückzugbewegung.

Siehe auch

Schnellabheben im linearen Zusammenhang der Achsen (Seite 611)

11.2.5.3 Parametrierung: Systemvariable**Achsspezifische Funktionsfreigabe**

Die Freigabe zum Rückziehen einer Achse erfolgt achsspezifisch über die Systemvariable:

```
$AA_ESR_ENABLE[<Achse>] = 1
```

Kanalspezifisches Auslösen

Das Auslösen der Rückzugsbewegung erfolgt kanalspezifisch über die Systemvariable:

```
$AC_ESR_TRIGGER = 1
```

Das Rückziehen erfolgt dann in allen Kanalachsen für die gilt:

```
$AA_ESR_ENABLE[<Achse>] == 1 UND MD37500 $MA_ESR_REACTION[<Achse>] == 21  
ODER 22
```

11.2.5.4 Programmierung (POLF, POLFA, POLFMASK, POLFMLIN)**Syntax**

```
POLF (<Achse>) = <Position>  
POLFA (<Achse>, <Typ>, <Position>)  
POLFA (<Achse>, <Typ>)  
POLFMASK (<Achse_1>, <Achse_2>, ...)  
POLFMLIN (<Achse_1>, <Achse_2>, ...)
```


Bedeutung

POLF:	Adresse zur Angabe der Zielposition der Rückzugsachse POLF ist modal wirksam.		
	<Achse>:	Kanalachsname der Rückzugsachse	
	<Position>:	Rückzugsposition	
		Typ: REAL Für Geometrieachse erfolgt das Rückziehen im WKS. Für alle anderen Kanalachsen im MKS. Bei gleichen Bezeichnern für Geometrie- und Kanal-/Maschinenachse wird im WKS zurückgezogen.	
POLFA:	Vordefinierter Unterprogrammaufruf zur Angabe der Rückzugsposition von Einzelachsen		
	<Achse>:	Kanalachsname der Rückzugsachse	
	<Typ>:	Positionsangabemodus	
		Typ: INT	
		Wert: 0	Positionswert als ungültig markieren
		1	Positionswert ist absolut
	2	Positionswert ist inkrementell (Distanz)	
Hinweis: Ist eine Achse keine Einzelachse oder fehlt der Typ bzw. Typ=0, dann wird ein entsprechender Alarm gemeldet.			
<Position>:	Rückzugsposition (s. o.) (optional) Hinweis: Der Positionswert wird auch mit Typ=0 übernommen. Nur ist dieser Wert dann als ungültig markiert und muss für das Rückziehen neu programmiert werden.		
POLFMASK:	Vordefinierter Unterprogrammaufruf zur Auswahl der Achsen, die nach Auslösung des Schnellabhebens unabhängig voneinander zurückgezogen werden sollen.		
	<Achse_1>, ...:	Namen der Achsen, die beim Schnellabheben auf ihre mit POLF definierten Positionen fahren sollen. Alle angegebenen Achsen müssen sich im gleichen Koordinatensystem befinden.	
	POLFMASK () ohne Angabe einer Achse deaktiviert das Schnellabheben für alle Achsen, die unabhängig voneinander zurückgezogen wurden.		
POLFMLIN:	Vordefinierter Unterprogrammaufruf zur Auswahl der Achsen, die nach Auslösung des Schnellabhebens im linearen Zusammenhang zurückgezogen werden sollen.		
	<Achse_1>, ...:	s. o.	
	POLFMLIN () ohne Angabe einer Achse deaktiviert das Schnellabheben für alle Achsen, die im linearen Zusammenhang zurückgezogen wurden.		

Hinweis

Bevor über POLFMASK oder POLFMLIN das Schnellabheben auf eine feste Position freigegeben werden kann, muss für die ausgewählten Achsen eine Position mit POLF programmiert worden sein.

Hinweis

Werden Achsen nacheinander mit POLFMASK, POLFMLIN oder POLFMLIN, POLFMASK freigegeben, gilt für die jeweilige Achse immer die letzte Festlegung.

Hinweis

Die mit POLF programmierten Positionen und die Aktivierung durch POLFMASK oder POLFMLIN werden bei Teileprogrammstart gelöscht. Das heißt, der Anwender muss in jedem Teileprogramm die Werte für POLF und die selektierten Achsen in POLFMASK bzw. POLFMLIN neu programmieren.

Hinweis

Wenn bei Verwendung der Kurzformen POLFA nur der Typ geändert wird, dann muss der Anwender sicherstellen, dass entweder die Rückzugsposition oder der Rückzugsweg einen sinnvollen Wert enthält. Insbesondere sind die Rückzugsposition und der Rückzugsweg nach Warmstart neu zu setzen.

Beispiel: Rückziehen einer Einzelachse

Programmcode	Kommentar
MD37500 \$MA_ESR_REACTION[AX1]=21	; NC-geführtes Rückziehen.
...	
\$AA_ESR_ENABLE[AX1]=1	
POLFA (AX1,1,20.0)	; AX1 wird die axiale Rückzugsposition 20.0 (absolut) zugeteilt.
\$AA_ESR_TRIGGER[AX1]=1	; Ab hier beginnt der Rückzug.

11.2.5.5 Randbedingungen

Allgemeine Randbedingungen

Ausnahmen

Rückziehen bzw. Schnellabheben wird **nicht** bei folgende Achsen ausgeführt:

- Achsen die nicht fest einem Kanal zugeordnet sind
- Achsen die sich im drehzahlgesteuerten Betrieb befinden (Spindeln)
- Achsen die als positionierende Spindeln (SPOS/SPOSA) interpoliert werden

Modulorundachsen

Modulorundachsen verhalten sich beim Schnellabheben folgendermaßen:

- Bei inkrementeller Programmierung der Zielposition wird diese ohne Modulkorrektur angefahren.
- Bei absoluter Programmierung wird die Zielposition unter Verwendung von Modulkorrekturen zeitoptimal angefahren. Dies entspricht weitestgehend einer Positionierung auf kürzestem Weg.

Ausnutzung der Achsdynamik

Die Rückzugsbewegung wird unter Ausnutzung der maximalen Beschleunigung und Geschwindigkeit der beteiligten Achsen in `POLFLIN` linear interpoliert.

Paralleles Rückziehen

Im **Kanal** kann jeweils nur **ein lineares** Rückziehen erfolgen. D. h. es können im Kanal nicht mehrere Achsverbände gebildet werden, die im linearen Zusammenhang ihre Rückzugspositionen anfahren.

Parallel zum linearen Rückziehen können zusätzliche Achsen mittels `POLFMASK` eine unabhängige, achsspezifische Abhebbewegung auf ihre programmierte Rückzugsposition durchführen.

Konkurrierendes Rückziehen

Werden Achsen sowohl in `POLFMASK` als auch in `POLFMLIN` verwendet, so ist zu beachten, dass immer der zuletzt programmierte Zustand im Falle einer Rückzugsbewegung aktiv ist. D. h., eine Achse die zuvor mit `POLFMLIN` aktiviert wurde, wird nach der Programmierung in `POLFMASK` aus dem linearen Zusammenhang entfernt und die Rückzugsbewegung würde jetzt als unabhängige Bewegung erfolgen (siehe Beispiele im Kapitel "Schnellabheben im linearen Zusammenhang der Achsen (Seite 611)").

Wirksamkeit der Rückzugsparameter

Für die Rückzugsbewegung sind die zum **Zeitpunkt der Auslösung** gültigen Parameter bestimmend. Ändert sich einer dieser Parameter (`POLF`, `POLFMASK`, `POLFMLIN`, `Frame...`) während der Rückzugsbewegung (z. B. durch Satzwechsel), so hat diese Änderung keinen Einfluss auf die bereits gestartete Rückzugsbewegung.

Randbedingungen: Bahnachsen

Alternatives Erweitertes Stillsetzen

Ist für eine Bahnachse `ESR_REACTION=21` (NC-geführter Rueckzug) projektiert und mit `$AC_ESR_ENABLE=1` freigegeben, wird für alle Bahnachsen für die kein `ESR_REACTION=21` projektiert bzw. freigegeben ist, `ESR_REACTION=22` (erweitertes Stillsetzen) aktiv.

Schnellabheben ohne freigegebene Rückzugsbewegung

Ist für eine Achse `ESR_REACTION=21` projektiert und mit `$AC_ESR_ENABLE=1` freigegeben, aber z.B. mit `POLFMASK` keine Rückzugsbewegung freigegeben, wird für diese Achse `ESR_REACTION=22` (erweitertes Stillsetzen) aktiv.

Verhalten bei unterschiedlichen Freigaben

Für die Bahnachsen X und Y ist "NC-geführter Rückzug" projektiert:

- MD37500 \$MA_ESR_REACTION[X] = 21
 - MD37500 \$MA_ESR_REACTION[Y] = 21
1. Für beide Bahnachsen ist "Erweitertes Stillsetzen und Rückziehen" und die Rückzugbewegung freigeben:
 - \$AA_ESR_ENABLE(X) = 1
 - \$AA_ESR_ENABLE(Y) = 1
 - POLFMASK(X,Y)

X und Y führen beim Schnellabheben die programmierte Rückzugbewegung aus.

2. Nur für eine Bahnachse ist "Erweitertes Stillsetzen und Rückziehen", aber für beide Bahnachsen die Rückzugbewegung freigeben:
 - \$AA_ESR_ENABLE(X) = 0
 - \$AA_ESR_ENABLE(Y) = 1
 - POLFMASK(X,Y)

Für Achse X wird beim Schnellabheben aufgrund des Bahnzusammenhangs trotz fehlender Freigabe "erweitertes Stillsetzen" entsprechend ESR_REACTION = 22 ausgeführt.

Achse Y führt die programmierte Rückzugbewegung aus.

3. Für beide Bahnachsen ist "Erweitertes Stillsetzen und Rückziehen", aber nur für eine Bahnachse die Rückzugbewegung freigeben:
 - \$AA_ESR_ENABLE(X) = 1
 - \$AA_ESR_ENABLE(Y) = 1
 - POLFMASK(Y)

Für Achse X wird beim Schnellabheben aufgrund des Bahnzusammenhangs trotz fehlender Freigabe der Rückzugbewegung "erweitertes Stillsetzen" entsprechend ESR_REACTION = 22 ausgeführt.

Achse Y führt die programmierte Rückzugbewegung aus.

11.2.6 Auslösequellen

Die ESR-Auslösequellen müssen anwenderspezifisch durch Auswertung von Systemvariablen unterschieden werden. Dafür stehen alle Systemvariablen zur Verfügung, die in **Synchronaktionen** lesbar sind.

Allgemeine Auslösequellen

- Digitale Eingänge der NCU-Baugruppe oder das steuerungsinterne, rücklesbare Abbild digitaler Ausgänge: \$A_IN, \$A_OUT
- Kanalzustand: \$AC_STAT

- NC/PLC-Nahtstellensignale
- Von PLC geschriebene Systemvariable: \$A_DBB, \$A_DBW, \$A_DBD
Die Verwendung dieser von PLC geschriebenen Systemvariablen wird bei **zeitkritischen** Signalen **nicht** empfohlen, da hier die PLC-Zykluszeit in die Gesamt-Reaktionszeit eingeht. Als Einflussmöglichkeit der PLC auf den Ablauf bzw. die Freigabe des Erweiterten Stillsetzens und Rückziehens ist dieser Weg aber dennoch sinnvoll. Weiterhin ist eine solche Einknüpfung von PLC-Zuständen sinnvoll, soweit diese **ausschließlich** von der PLC vorgegeben werden (z. B. Not-Halt, Reset-Taste, Stopp-Taste)
- Sammelmeldungen von Alarmen: \$AC_ALARM_STAT

Axiale Auslösequellen

- \$VA_SYNCDIFF[<Folgeachse>] (Synchronlaufdifferenz istwertseitig)
- \$AA_ESR_STAT (Axiales Rückmeldewort: ESR Status)

Alarmreaktionen

Bei aktivem ESR verzögern sich die Alarmreaktionen NOREADY und STOPBYALARM um einen IPO-Takt. Als Hinweis für die Verzögerung wird der selbstlöschende Alarm angezeigt: Alarm 21600 "Überwachung für ESR aktiv".

Hinweis

Die Anzeige des Alarms kann unterdrückt werden durch:

MD11410 \$MN_SUPPRESS_ALARM_MASK, Bit16 = 1

11.2.7 Verknüpfungslogik: Quellen-/Reaktionsverknüpfung

Die flexiblen Verknüpfungsmöglichkeiten der **statischen Synchronaktionen** können genutzt werden, um aufgrund von Quellen bestimmte Reaktionen auszulösen. Die Verknüpfung aller relevanten Quellen mit Hilfe statischer Synchronaktionen liegt in den Händen des Anwenders/Maschinenherstellers. Er kann die Quellen-Systemvariablen als Ganzes oder mit Hilfe von Bit-Masken auch selektiv auswerten und hieran seine gewünschten Reaktionen knüpfen. Die statischen Synchronaktionen sind in allen Betriebsarten wirksam. Eine ausführliche Beschreibung der Verwendung der Synchronaktionen finden Sie in:

Literatur:

- Funktionshandbuch Synchronaktionen
- Programmierhandbuch Arbeitsvorbereitung (Synchronaktionen, Systemvariablen)

Die Verknüpfung axialer Quellen mit globalen bzw. kanalspezifischen Quellen werden bei Bedarf z. B. mit Hilfe von \$AA_TYP (Achstyp) variabel konfiguriert.

11.2.8 Aktivierung

Option

Die Funktion "Erweitertes Stillsetzen und Rückziehen" ist eine Option.

Achsspezifische Funktionsfreigabe (\$AA_ESR_ENABLE)

Die achsspezifische Funktionsfreigabe erfolgt über die Systemvariable:

`$AA_ESR_ENABLE[<Achse>] = 1`

Achsspezifische Freigabe zum Erweiterten Stillsetzen

Die Freigabe einer Achse für das Erweiterte Stillsetzen erfolgt mit:

`MD37500 $MA_ESR_REACTION[Achse] = 22`

Achsspezifische Freigabe zum Rückziehen

Die Freigabe einer Achse für das Rückziehen erfolgt mit:

`MD37500 $MA_ESR_REACTION[Achse] = 21`

Kanalspezifisches Auslösen (\$AC_ESR_TRIGGER)

ESR wird kanalspezifische durch Setzen folgender Systemvariable ausgelöst:

`$AC_ESR_TRIGGER = 1`

ESR erfolgt dann in allen Kanalachsen für die gilt:

`$AA_ESR_ENABLE[<Achse>] == 1 UND MD37500 $MA_ESR_REACTION[Achse] == 21
ODER 22`

11.2.9 Projektierungshilfe für ESR

Spannungsausfall

Für folgende Baugruppen muss bei Ausfall der Netzspannung die Stromversorgung durch geeignete Maßnahmen mindesten bis zum Stillsetzen der Achsen sichergestellt werden:

- SINUMERIK 840D sl NCU 7x0
- SINUMERIK NCU-Peripheriebaugruppen
- SIMATIC PLC-Peripheriebaugruppen
- SINAMICS Antriebssystem S120 (Booksize)

Zwischenkreisenergie

Die bei Ausfall der Netzspannung im Zwischenkreis der Antriebsgeräte noch zur Verfügung stehende Energie berechnet sich zu:

$$E = 1/2 * C * (UZK_{\text{warn}}^2 - UZK_{\text{min}}^2)$$

E: Energie in Wattsekunden [Ws]

C: Gesamtkapazität des Zwischenkreises in Farad [F]

UZK_{warn}: Zwischenkreisspannung ab der Unterspannung erkannt wird.
SINAMICS Parameter: r0296 + p0279

UZK_{min}: Untergrenze der Zwischenkreisspannung für einen sicheren Betrieb, mit Berücksichtigung der motorspezifischen EMK.

Hinweis $UZK_{\text{min}} > UZK_{\text{aus}}$

Diese Energie steht mindestens für die Zeit t_{min} zur Verfügung:

$$t_p = E / P_n * \eta$$

E: Energie in Wattsekunden [Ws]

t_p: Pufferzeit in Millisekunden [ms]

P_n: Leistung in Kilowatt [kW]

η: Wirkungsgrad des Antriebsgerätes

Beispiel

Annahmen:

- P_n = 16 kW, Einspeisung (ALM) mit 3 AC 380 V
- C = C_{ist} - 20% = 20.000 μF - 20% = 16.000 μF = 16 * 10⁻³ F, zur Sicherheit wird eine um 20% geringere effektiv zur Verfügung stehende Kapazität angenommen.
- UZK_{warn} = 550 V
- UZK_{min} = 350 V

$$E = 1/2 * 16 * 10^{-3} \text{ F} * ((550 \text{ V})^2 - (350 \text{ V})^2) = 1440 \text{ Ws}$$

Diese Energie steht für eine Zeit t_{min} z. B. für eine Rückzugsbewegung zur Verfügung:

- E = 1440 Ws
- P_n = 16 kW
- η = 0,9 (Annahme)

$$t_{\text{Pmin}} = 1440 \text{ Ws} / 16 \text{ kW} * 0,9 = 81 \text{ ms}$$

In der nachfolgenden Tabelle sind die Werte für verschieden SINAMICS Einspeisungen (ALM) aufgeführt. Hierbei sind nominale (C_{max}) und minimale (C_{min}) Kapazität berücksichtigt.

Die zur Verfügung stehende Gesamtkapazität des Zwischenkreises setzt sich, unter Berücksichtigung der jeweiligen maximalen Kapazität C_{max} (Ladegrenze), aus den Kapazitäten der Einspeisung (ALM), der Motormodule, sowie eventuell vorhandener Kondensatormodule (Capacitor Module) zusammen.

Die in der Tabelle für den minimalen Energieinhalt (E_{\min} bei C_{\min}) angesetzte Kapazität berücksichtigt eine Bauteiletoleranz von -20 % (schlechtester Fall).

Hinweis

Es wird empfohlen, zur Ermittlung der zur Verfügung stehenden Gesamtkapazität des Zwischenkreises, das Projektierungs-Tool SIZER zu verwenden.

Tabelle 11-1 SINAMICS Einspeisungen (ALM): nominale und minimale Pufferzeiten

ALM P_{\max} [kW]	max. Kapazität C_{\max} [μ F]	Energie-Inhalt E_{\max} bei C_{\max} [Ws]	Energie-Inhalt E_{\min} bei C_{\min} [Ws]	Pufferzeit $t_{P_{\max}}$ bei P_{\max} [ms]	Pufferzeit $t_{P_{\min}}$ bei P_{\max} [ms]
16	20000	1800	1440	101,25	81,00
36	20000	1800	1440	45,00	36,00
55	20000	1800	1440	29,46	23,56
80	20000	1800	1440	20,25	16,20
120	20000	1800	1440	13,50	10,80

Energiebilanz

Bei der Projektierung der Rückzugsbewegung muss in jedem Fall eine Energiebilanz aufgestellt werden, um Aussagen darüber zu erhalten, ob auf zusätzliche Kondensatormodule oder eine Generatorachse mit entsprechend dimensionierter Schwungmasse verzichtet werden kann.

Stillsetzen als Energielieferant

Ab dem ca. dritten Interpolatortakt erfolgen Änderungen der Drehzahlsollwerte der projektierten Stillsetz- oder Rückzugsachsen. Nach dieser Zeit beginnt die Bremsphase (wenn kein antriebsautarkes Stillsetzen in dieser Achse projektiert ist).

Sobald der Bremsvorgang einsetzt, steht die dabei freiwerdende Energie für die Rückzugsbewegung zur Verfügung. Über eine Energiebilanz muss sichergestellt werden, dass die kinetische Energie der bremsenden Achsen für den Rückzug ausreicht.

Anhand der Energiebilanz kann festgestellt werden, welche Interpolatortaktzeit maximal eingestellt werden kann, um einen sicheren Notrückzug durchführen zu können.

Beispiel

Bei einer 16 kW-Einheit unter maximaler Last und minimaler Zwischenkreiskapazität soll ein Notrückzug ohne Generatorbetrieb möglich sein. Dafür darf die Interpolatortaktzeit theoretisch höchstens 4,86 ms betragen, d. h. in diesem Fall können maximal 4 ms eingestellt werden.

Bei Bedarf muss auf eine leistungsfähigere NCU ausgewichen werden, um optimale Bedingungen zu erreichen.

Generatorbetrieb

Für Fälle, in denen die Energie des Zwischenkreises nicht für ein sicheres Rückziehen (mindestens 3 IPO-Takten) ausreicht, kann über die SINAMICS Funktion "Vdc-Regelung" für einen Antrieb Generatorbetrieb projiziert werden. Hierbei wird die mechanische Energie einer Achse in den Zwischenkreis zurückgespeist.

Die in einer Achse gespeicherte Energie berechnet sich zu:

$$E = 1/2 * J_{ges} * \omega^2$$

J_{ges} gesamtes Massenträgheitsmoment [kg*m²]

ω Winkelgeschwindigkeit zum Zeitpunkt der Umschaltung auf Generatorbetrieb [s⁻¹]

Diese Energie wird mit einem Wirkungsgrad von ca. 90% in den Zwischenkreis zurückgespeist:

Es wird empfohlen bei Verwendung von Einspeisungen hoher Leistung (55, 80, 120 kW) für den Generatorbetrieb eine eigene Achse mit zusätzlicher Schwungmasse vorzusehen. Für diese Achse muss nach dem Beschleunigen auf Nenn Drehzahl nur noch die Energie zur Kompensation der Reibungsverluste aufgebracht werden.

Prinzipiell kann aber jede beliebige Achse für den Generatorbetrieb verwendet werden, sofern diese nicht direkt am Erweiterten Stillsetzen und Rückziehen oder an Kopplungen beteiligt ist, die gezielt aufrecht erhalten werden müssen.

Um zu verhindern, dass bei Einsetzen des Generatorbetriebs die Zwischenkreisspannung zu groß wird (Grenzwert: Impulslöschung Antrieb) und als Folge davon für die Achsen Impulslöschung ausgelöst wird, sind ausreichend dimensionierte Impulswiderstandsmodule oder die SINAMICS Funktion "Vdc-Regelung, Überspannungsüberwachung" zu verwenden.

11.2.10 Steuerungsverhalten

11.2.10.1 Achsverhalten in Abhängigkeit der Freigaben

Systematik

Ist eine Achse als NC-geführte Rückzugsachse projiziert:

MD37500, \$MA_ESR_REACTION[<Achse>] = 21

müssen für die Durchführung der Rückzugsbewegung im ESR-Fall folgende Freigaben vorliegen:

- Axiale ESR-Freigabe über Systemvariable: \$AA_ESR_ENABLE[<Achse>] = 1
- Axiale Rückzugsfreigabe über Teileprogrammbefehl: POLFMASK (<Achse>)

Liegt eine der beiden Freigaben nicht vor, verhält sich die Achse im ESR-Fall entsprechend der folgenden Tabelle:

Freigaben	Reaktion im ESR-Fall
<ul style="list-style-type: none"> • \$AA_ESR_ENABLE[<Achse>] = 1 • Keine Freigabe über POLFMASK 	Für die Achse wird implizit NC-geführten Stillsetzen, entsprechend MD 37500, \$MA_ESR_REACTION[<Achse>] = 22, aktiv.
<ul style="list-style-type: none"> • \$AA_ESR_ENABLE[<Achse>] = 0 • Freigabe über POLFMASK (<Achse>) 	<p>Achse ist Bahnachse: Die ESR-Reaktion wird für die Achse implizit auf NC-geführtes Stillsetzen geändert. Das Verhalten entspricht dann einer Projektierung von MD 37500, \$MA_ESR_REACTION[<Achse>] = 22</p> <p>Achse ist Zusatzachse: Die Achse wird mit Schnellstopp stillgesetzt.</p>

Beispiele für Achsen mit Bahnzusammenhang

Für die nachfolgenden Beispiele werden zwei Bahnachsen X und Y angenommen, die als NC-geführte Rückzugsachsen projektiert und für die Rückzugspositionen programmiert sind:

- MD37500, \$MA_ESR_REACTION[X] = 21
- MD37500, \$MA_ESR_REACTION[Y] = 21
- POLF[X]=<Rückzugposition> LFPOS
- POLF[Y]=<Rückzugposition> LFPOS

Ausgangssituation	Reaktion im ESR-Fall
<ul style="list-style-type: none"> • \$AA_ESR_ENABLE[X] = 1 • \$AA_ESR_ENABLE[Y] = 1 • POLFMASK (X, Y) • Verfahrbewegung prog.: Y 	Rückzugsbewegung in X und Y.
<ul style="list-style-type: none"> • \$AA_ESR_ENABLE[X] = 0 • \$AA_ESR_ENABLE[Y] = 1 • POLFMASK (X, Y) • Verfahrbewegung prog.: X und Y 	<p>X hat keine ESR-Freigabe. Da X Bahnachse ist, wird die ESR-Reaktion für X implizit auf 22 (Stillsetzen) geändert. X fährt deshalb entsprechend der programmierten Verfahrbewegung weiter.</p> <p>Y führt Rückzugsbewegung aus.</p> <p>Nach Ablauf der Verzögerungszeit für ESR-Achsen MD21380, \$MC_ESR_DELAY_TIME1 wird X mit Bahnbeschleunigung gestoppt.</p>
<ul style="list-style-type: none"> • \$AA_ESR_ENABLE[X] = 1 • \$AA_ESR_ENABLE[Y] = 1 • POLFMASK (Y) • Verfahrbewegung prog.: X 	<p>X hat keine Rückzugsfreigabe (POLFMASK). X führt keine Rückzugsbewegung aus. Da X Bahnachse ist, wird die ESR-Reaktion für X implizit auf 22 (Stillsetzen) geändert. X fährt deshalb entsprechend der programmierten Verfahrbewegung weiter.</p> <p>Y führt Rückzugsbewegung aus.</p> <p>Nach Ablauf der Verzögerungszeit für ESR-Achsen MD21380, \$MC_ESR_DELAY_TIME1 wird X mit Bahnbeschleunigung gestoppt.</p>
<ul style="list-style-type: none"> • \$AA_ESR_ENABLE[X] = 1 • \$AA_ESR_ENABLE[Y] = 1 • POLFMASK (X) • Verfahrbewegung prog.: X und Y 	<p>Y hat keine Rückzugsfreigabe (POLFMASK). Y führt keine Rückzugsbewegung aus. Da Y Bahnachse ist, wird die ESR-Reaktion für Y implizit auf 22 (Stillsetzen) geändert. Y fährt deshalb entsprechend der programmierten Verfahrbewegung weiter.</p> <p>X führt Rückzugsbewegung aus.</p> <p>Nach Ablauf der Verzögerungszeit für ESR-Achsen MD21380, \$MC_ESR_DELAY_TIME1 wird Y mit Bahnbeschleunigung gestoppt.</p>

Beispiele für Achsen ohne Bahnzusammenhang

Für die nachfolgenden Beispiele wird eine Bahnachsen X und eine Kommandoachse B angenommen, die jeweils als NC-geführte Rückzugsachse projiziert sind:

- MD37500, \$MA_ESR_REACTION[X] = 21
- MD37500, \$MA_ESR_REACTION[B] = 21

Ausgangssituation	Reaktion im ESR-Fall
<ul style="list-style-type: none"> • \$AA_ESR_ENABLE[X] = 1 • \$AA_ESR_ENABLE[B] = 0 • POLFMASK (X) • Verfahrbewegung prog.: X und B 	<p>B hat keine ESR-Freigabe. Da B keine Bahnachse ist, wird B sofort mit Schnellstopp angehalten.</p> <p>X führt Rückzugsbewegung aus.</p>
<ul style="list-style-type: none"> • \$AA_ESR_ENABLE[X] = 1 • \$AA_ESR_ENABLE[B] = 1 • POLFMASK (X) • Verfahrbewegung prog.: X und B 	<p>B hat keine Rückzugsfreigabe (POLFMASK). Da B aber ESR-Freigabe hat, wird die ESR-Reaktion für B implizit auf 22 (Stillsetzen) geändert. B fährt deshalb entsprechend der programmierten Verfahrbewegung weiter.</p> <p>Y führt Rückzugsbewegung aus.</p> <p>Nach Ablauf der Verzögerungszeit für ESR-Achsen MD21380, \$MC_ESR_DELAY_TIME1 wird B mit axialer Beschleunigung gestoppt.</p>

11.2.10.2 POWER OFF/POWER ON

Wird die anwenderspezifische Rückzugslogik innerhalb von Bewegungssynchronaktionen programmiert, ist zu beachten, dass diese im Hochlauf der Steuerung (POWER ON) noch nicht aktiv sind. Eine Rückzugslogik, die nach POWER ON aktiv sein soll, muss daher in einem von der PLC gestarteten ASUP liegen oder ereignisgesteuert aufgerufen werden.

Literatur

Ausführliche Informationen zu ereignisgesteuerten Programmaufrufen siehe:

Funktionshandbuch Grundfunktionen; BAG, Kanal, Programmbetrieb, Reset-Verhalten (K1), Kapitel: Ereignisgesteuerte Programmaufrufe

11.2.10.3 Betriebsartenwechsel, NC-Stop, Reset

Für die anwenderspezifische Rückzugslogik können **statische** Synchronaktionen (Schlüsselwort IDS) verwendet werden, da die folgenden Systemzustände bzw. Zustandswechsel auf diese keinen Einfluss haben:

- Betriebsartenwechsel, z. B. DB11 DBX0.0 (AUTOMATIK)
- NC-Stop, z. B. DB21, ... DBX7.3 (NC-Stopp)
- Reset, z. B. DB21, ... DBX7.7 (Reset)

Hinweis

NC-Stop und Kommandoachsen

Das Verfahren von Kommandoachsen wird bei NC-Stop abgebrochen.

Hinweis

Reset und Programmbefehle POLF/POLFMASK

Bei Reset werden die programmierten absoluten Rückzugspositionen (POLF) und die Freigaben der Rückzugsachsen (POLFMASK) nicht gelöscht.

11.2.10.4 Teileprogrammstart, NC-Start

Damit beim Start eines Teileprogramms ein definierter Ausgangszustand vorliegt, werden die programmierten absoluten Rückzugspositionen und die Freigaben der Rückzugsachsen mit dem Start des Teileprogramms gelöscht.

Hinweis

Rückzugspositionen und Freigaben

Beim Start eines Teileprogramms müssen die absoluten Rückzugspositionen (POLF) und die Freigaben der Rückzugsachsen (POLFMASK) entsprechend den aktuellen Anforderungen neu programmiert werden.

11.2.10.5 Alarmverhalten

Das Alarmverhalten ist abhängig von der Achse in welcher ein Fehler auftritt:

- Fehler in einer Achse außerhalb des Achsverbandes eines Elektronischen Getriebes: Diese Achse schaltet "normal" ab. Erweitertes Stillsetzen und Rückziehen arbeiten "ungestört" weiter bzw. werden durch solche Fehler ausgelöst.
- Fehler in einer Leitachse (LA): selektive Umschaltung auf Istwertkopplung bereits während des Stillsetzens, ansonsten wie zuvor.
- Fehler in einer Folgeachse (FA):
 - Rückziehen durchführen: Rückzugsachse darf nicht Folgeachse sein, d. h. kein Konflikt.
 - Stillsetzen durchführen: Es kann sein, dass die Folgeachse unkontrolliert reagiert. Die Rettung von Werkstück/Werkzeug muss dann dem Rückziehen überlassen werden, das Stillsetzen soll aber zumindest keine weiteren Störungen in den Ablauf einbringen.

- Fehler in der Rückzugsachse: es gibt keinen Rückzug.
- Not-Halt
Ein Not-Halt ist aus Steuerungssicht kein Fehlerfall, sondern wird wie jedes andere Steuersignal behandelt. Not-Halt bricht aus Sicherheitsgründen die Interpolation und alle Fahrbewegungen ab und löst durch Wegnahme der Reglerfreigaben auch die elektronische Kopplung auf.
Bei Anwendungsfällen, in denen es auch bei Not-Halt auf die Aufrechterhaltung der Kopplung und Fahrbewegungen ankommt, muss dieser Not-Halt von der **PLC** entsprechend lange **verzögert** werden, bis die geforderten NC- bzw. antriebsseitigen Reaktionen ablaufen können.
Als Rückmeldesignal an die PLC steht das folgende Nahtstellensignal zur Verfügung: NST "ESR-Reaktion ist ausgelöst" DB31, ... DBX98.7

Wenn während der aktiven Phase des ESR ein Alarm auftritt mit kanalübergreifender NOREADY-Reaktion (d. h. NOREADY | NCKREACTIONVIEW | BAGREACTIONVIEW), dann hat das zur Folge, dass in **allen** Kanälen ESR ausgelöst wird.

11.2.10.6 Satzsuchlauf, REPOS

Erweitertes Stillsetzen und Rückziehen hat auf Satzsuchlauf- oder REPOS-Bewegungen keinen Einfluss.

11.3 Antriebsautarkes ESR

11.3.1 Grundlagen

Funktion

Antriebsautarkes Erweiterte Stillsetzen und Rückziehen (ESR) ermöglicht die schnelle Trennung von Werkstück und Werkzeug unabhängig von der übergeordneten Steuerung (NC).

Im Antrieb können dazu folgende achsspezifischen Funktionen projiziert werden:

- Generatorbetrieb
- Erweitertes Stillsetzen
- Rückziehen

Die antriebsautarken Reaktionen werden in Fehlersituationen automatisch ausgelöst. Das Auslösen der antriebsautarken Reaktionen kann auch anwenderspezifisch über Teileprogramme oder Synchronaktionen von der übergeordneten Steuerung aus erfolgen.

Da die Stillsetz- und Rückzugsbewegungen des antriebsautarken ESR rein achsspezifisch erfolgen, werden dabei im Gegensatz zum steuerungsgeführten ESR keine Kopplungen berücksichtigt.

Antriebsautarkes ESR kann auch in Kombination mit dem steuerungsgeführten ESR eingesetzt werden. Dadurch können Stillsetz- und Rückzugsbewegungen auch dann in den

11.3 Antriebsautarkes ESR

Antrieben erfolgen, wenn diese von der Steuerung, z. B. durch Kommunikationsausfall, nicht mehr vorgegeben werden können.

Hinweis

DRIVE-CLiQ

Um bei eventuell auftretenden DRIVE-CLiQ Fehlern die Rückwirkung auf die Funktionen des antriebsautarken ESR möglichst gering zu halten, wird im Rahmen von ESR empfohlen, die beteiligten Motormodule nicht über eine Linientopologie, sondern direkt an der Control Unit (CU) anzuschließen.

Link-Achsen

Die Funktion "Antriebsautarkes Erweitertes Stillsetzen und Rückziehen (ESR)" kann nicht im Zusammenhang mit Link-Achsen verwendet werden.

Voraussetzungen

Folgende Voraussetzungen müssen gegeben sein:

- SINAMICS und SINUMERIK Softwareversion: ≥ V4.4
- Antriebsobjekt Servo, PROFIdrive Telegramm: 102 - 199
- Antriebsobjekt Servo, Funktionsmodul "ESR" aktiv
Das Funktionsmoduls "Erweitertes Stillsetzen und Rückziehen (ESR)" muss über den Antriebsassistenten von SINUMERIK Operate aktiviert werden. Zur Beschreibung des Antriebsassistenten siehe: SINUMERIK Operate Onlinehilfe
- Antriebsobjekt Servo, Generatorbetrieb (Vdc-Regelung) in einem Antrieb aktiv
- Antriebsobjekt Control Unit, PROFIdrive Telegramm: 390, 391
- Stützung der 24V Spannungsversorgung der Baugruppen über CSM oder USV

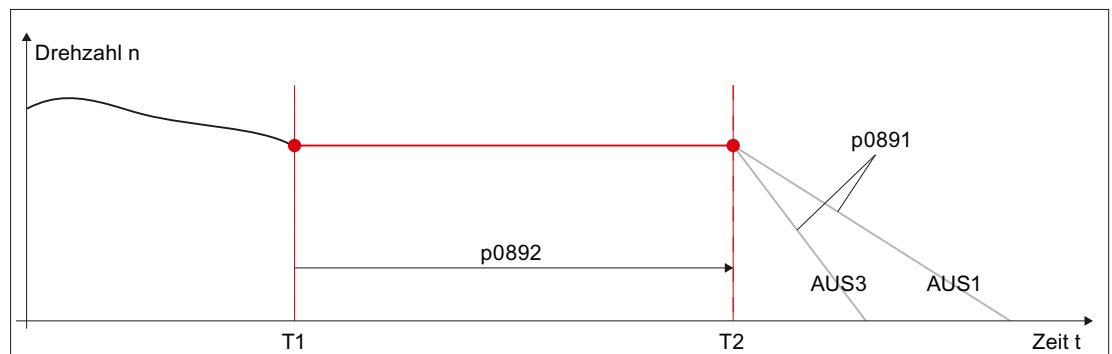
11.3.2 Stillsetzen im Antrieb projektieren

Das antriebsautarke Stillsetzen wird über folgende Antriebsparameter projiziert:

Parameter	Beschreibung	
p0888	ESR: Konfiguration	
	Wert	Bedeutung
	1	Erweitertes Stillsetzen (antriebsautark)

Parameter	Beschreibung	
p0891	ESR: Ausrampe	
	Wert	Bedeutung
	0	AUS3 (default) Der Antrieb wird durch sofortige Vorgabe von $n_{\text{soll}} = 0$ an der AUS3-Rücklauf- rampe (p1135: AUS3 Rücklaufzeit) abgebremst.
1	AUS1 Der Antrieb wird durch sofortige Vorgabe von $n_{\text{soll}} = 0$ an der Hochlaufgeber- Rücklauf-rampe (p1121: Hochlaufgeber Rücklaufzeit) abgebremst.	

Parameter	Beschreibung
p0892	ESR: Zeitstufe
	Der Antrieb fährt über die projektierte Zeit mit dem beim Fehleraustritt anliegenden Drehzahlsollwert konstant weiter.



T1 Zeitpunkt zu dem das Stillsetzen ausgelöst wurde

T2 Zeitpunkt nach Ablauf der in $p0892$ projektierten Zeit

AUS3/AUS1 Bremsrampe in Abhängigkeit von $p0891$

Bild 11-3 Verhalten bei antriebsautarkem Stillsetzen

Rückmeldung

Der Status des Stillsetzens wird an die Steuerung zurückgemeldet (siehe Kapitel "Rückmeldung des ESR-Status (Seite 600)").

Literatur

Eine ausführliche Beschreibung der Antriebsparameter findet sich in:

SINAMICS S120/S150 Listenhandbuch

11.3.3 Rückziehen im Antrieb projektieren

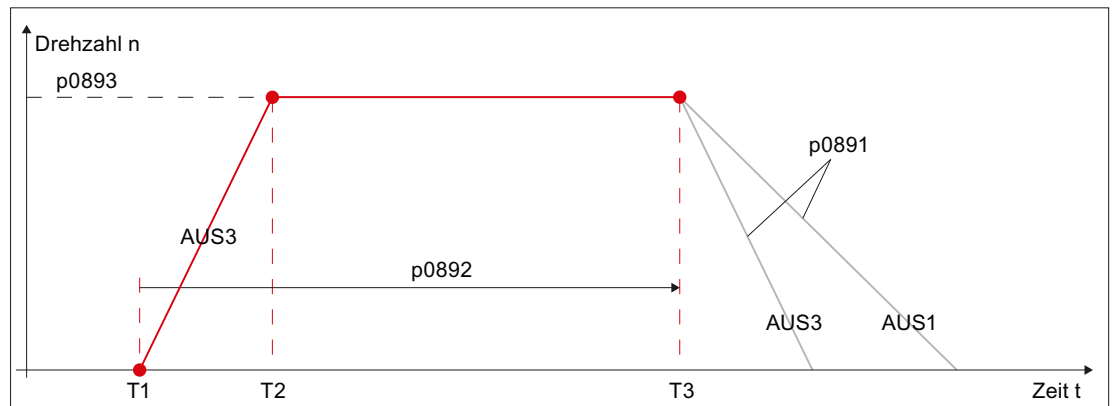
Das antriebsautarke Rückziehen wird über folgende Antriebsparameter projektiert:

Parameter	Beschreibung	
p0888	ESR: Konfiguration	
	Wert	Bedeutung
	2	Erweitertes Rückziehen (antriebsautark)

Parameter	Beschreibung	
p0891	ESR: Ausrampe	
	Wert	Bedeutung
	0	AUS3 (default) Der Antrieb wird durch sofortige Vorgabe von n_soll = 0 an der AUS3-Rücklauf- rampe (p1135: AUS3 Rücklaufzeit) abgebremst.
1	AUS1 Der Antrieb wird durch sofortige Vorgabe von n_soll = 0 an der Hochlaufgeber- Rücklauf-rampe (p1121: Hochlaufgeber Rücklaufzeit) abgebremst.	

Parameter	Beschreibung
p0892	ESR: Zeitstufe
	Der Parameter gibt die Zeit an, die in Summe für das Anfahren der in p0893 angege- benen Drehzahl und anschließender Konstantfahrt vergeht. Danach erfolgt je nach Parametrierung in p0891 eine AUS1 oder AUS3 Rampe.

Parameter	Beschreibung
p0893	ESR: Drehzahl
	Der Parameter gibt die Drehzahl (Rückzugsdrehzahl) an, die bei Auslösung des Rück- ziehens über eine AUS3 Rampe angefahren wird.



T1 Zeitpunkt zu dem Rückziehen ausgelöst wurde

T2 Zeitpunkt zu dem die in p0893 projektierte Rückzugsdrehzahl erreicht ist

T3 Zeitpunkt nach Ablauf der in p0892 projektierten Zeit

Bild 11-4 Verhalten bei antriebsautarkem Rückziehen

Rückmeldung

Der Status des Rückziehens wird an die Steuerung zurückgemeldet (siehe Kapitel "Rückmeldung des ESR-Status (Seite 600)").

Literatur

Eine ausführliche Beschreibung der Antriebsparameter findet sich in:

SINAMICS S120/S150 Listenhandbuch

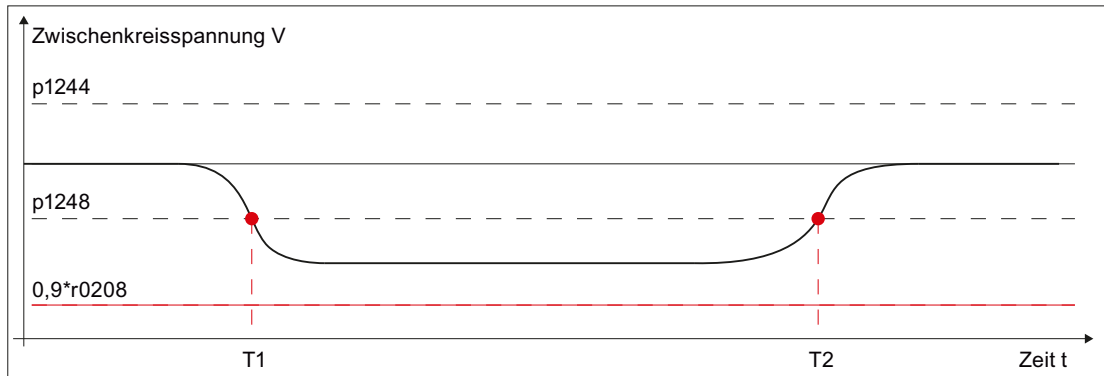
11.3.4 Generatorbetrieb im Antrieb projektieren

Der Generatorbetrieb für das antriebsautarke ESR wird über folgende Antriebsparameter projiziert:

Parameter	Beschreibung	
p0888	ESR: Konfiguration	
	Wert	Bedeutung
	3	Generatorbetrieb (Vdc-Regler)

Parameter	Beschreibung	
p1240	Vdc-Regler oder Vdc-Überwachung Konfiguration	
	Wert	Bedeutung
	2	Beim Erreichen der unteren Zwischenkreisspannungsschwelle (p1248) wird bei ESR der Motor abgebremst, um dessen kinetische Energie zur Pufferung des Zwischenkreises zu verwenden.

Parameter	Beschreibung
p1248	Zwischenkreisspannung Schwelle unten Einstellung der unteren Schwelle für die Zwischenkreisspannung. Diese Schwelle wird bei p1240 = 2 als Begrenzungssollwert für den Vdc_min-Regler verwendet.



- p1244 Zwischenkreisspannung Schwelle oben
- p1248 Zwischenkreisspannung Schwelle unten
- r0208 Leistungsteil Netzennennspannung.
Sinkt die Zwischenkreisspannung unter $0,9 \cdot r0208$ erfolgt Impulslöschung
- T1 Zeitpunkt zu dem der Generatorbetrieb ausgelöst wurde
- T2 Zeitpunkt zu dem der Generatorbetrieb beendet wurde

Bild 11-5 Antriebsaut_Generatorbetrieb_projektieren

Generator Minimaldrehzahl

Über den Antriebsparameter p2161 wird die Untergrenze der Motordrehzahl der Generatorachse projiziert:

Parameter	Beschreibung
p2161	Drehzahlschwellwert 3 Drehzahlschwellwert für die Meldung: $ n_{ist} < \text{Drehzahlschwellwert } 3$

Rückmeldung

Der Status des Generatorbetriebs wird an die Steuerung zurückgemeldet (siehe Kapitel "Rückmeldung des ESR-Status (Seite 600)").

Literatur

Eine ausführliche Beschreibung der Antriebsparameter und der findet Vdc-Regelung sich in:

- SINAMICS S120/S150 Listenhandbuch
- SINAMICS S120 Funktionshandbuch; Kapitel "Servoregelung" > "Vdc-Regelung"

11.3.5 ESR über Systemvariable freigeben

Die Freigabe der über die Antriebsparameter projektierten ESR-Reaktionen einer Achse muss anwenderspezifisch in einem Teileprogramm/Synchronaktion über folgende achsspezifische Systemvariable programmiert werden:

\$AA_ESR_ENABLE[<Achse>]	
Wert	Bedeutung
1	ESR-Reaktionen im Antrieb freigegeben
0	ESR-Reaktionen im Antrieb gesperrt

11.3.6 ESR über Systemvariable auslösen

Das Auslösen aller über Antriebsparameter projektierten und über \$AA_ESR_ENABLE[<Achse>] freigegebenen antriebsautarken ESR-Reaktionen muss anwenderspezifisch in einem Teileprogramm/Synchronaktion über folgende NC-spezifische Systemvariable programmiert werden:

\$AN_ESR_TRIGGER	
Wert	Bedeutung
1	freigegebene antriebsautarke ESR-Reaktionen auslösen
0	freigegebene antriebsautarke ESR-Reaktionen nicht auslösen

Hinweis

Das Trigger-Signal muss im Teileprogramm/Synchronaktion mindestens 2 IPO-Takte lang anstehen, um sicher über das PROFIdrive-Telegramm zum Antrieb übertragen zu werden.

Abbildung im Antriebsgerät

Die Systemvariable \$AN_ESR_TRIGGER wird auf den Antriebsparameter p0890.0 abgebildet. Die Verschaltung des Antriebsparameter p0890.0 mit der entsprechenden Information im Telegramm 390 bzw. 391 an die Control Unit erfolgt automatisch bei Inbetriebnahme der ESR-Reaktionen durch das Antriebsgerät.

Literatur

Eine ausführliche Beschreibung der Antriebsparameter findet sich in:

SINAMICS S120/S150 Listenhandbuch

11.3.7 Rückmeldung des ESR-Status

Die Rückmeldung des aktuellen ESR-Status erfolgt vom Antrieb an die Steuerung über das Meldewort (MELDW) des zyklischen PROFIdrive-Telegramms. In der Steuerung ist der Status über Systemvariable und NC/PLC-Nahtstellensignale lesbar.

Systemvariable

Die Systemvariable können im Teileprogramm/Synchronaktion ausgewertet werden, um z.B. das antriebsautarke ESR auszulösen (siehe Kapitel "ESR über Systemvariable auslösen (Seite 599)") oder die im Antrieb ausgelösten ESR-Reaktionen zu quittieren (siehe Kapitel "ESR-Reaktionen quittieren (Seite 600)").

Signalzusammenhang

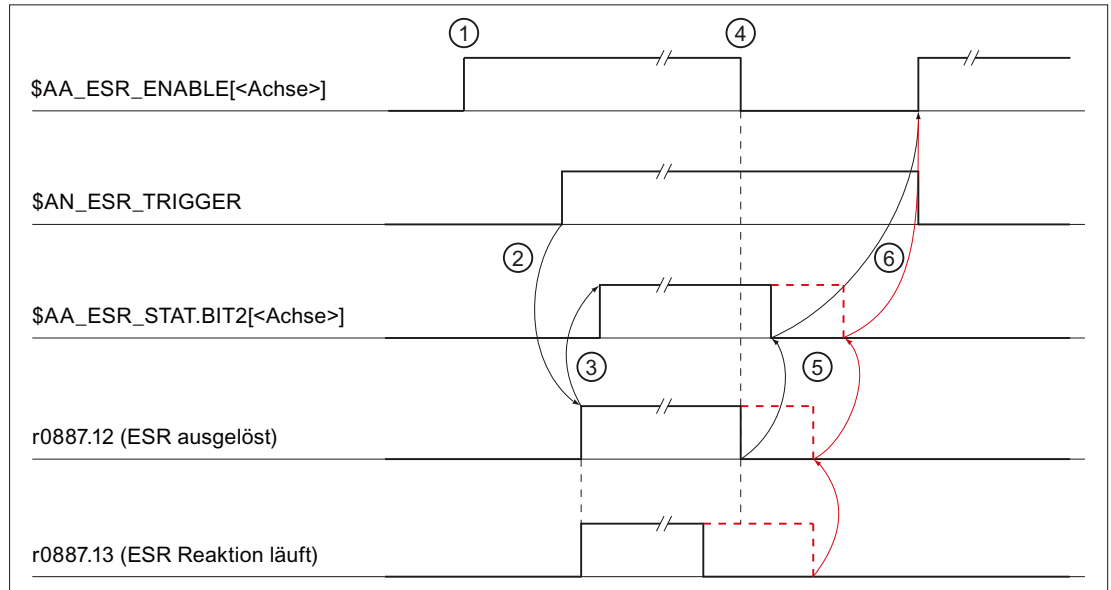
MELDW	Systemvariable	NC/PLC-Nahtstellen-signal	Bedeutung
MELDW.Bit 9 (r0887.12)	\$AA_ESR_STAT.Bit 2	DB31, ... DBX95.2	ESR: Reaktion ausgelöst oder Generatorbetrieb aktiv (r0887.12)
MELDW.Bit 4 (r0056.15)	\$AA_ESR_STAT.Bit 3	DB31, ... DBX95.1	ESR: Zwischenkreisunterspannung (p1248)
MELDW.Bit 2 (r2199.0)	\$AA_ESR_STAT.Bit 4	DB31, ... DBX95.3	ESR, Generatorbetrieb: Minimaldrehzahl unterschritten (p2161)

11.3.8 ESR-Reaktionen quittieren

Das Quittieren der achsspezifischen antriebsautarken ESR-Reaktionen muss anwenderspezifisch in einem Teileprogramm/Synchronaktion programmiert werden. Nach Abschluss der ESR-Reaktion im Antrieb muss in der Systemvariablen \$AA_ESR_ENABLE ein Flankenwechsel: 1→0→1 erzeugt werden. Damit die antriebsautarken ESR-Reaktionen erneut ausgelöst werden können, muss auch die Systemvariable \$AN_ESR_TRIGGER wieder auf den Wert 0 zurückgesetzt werden.

Systemvariable und Antriebsparameter

Das folgende Bild zeigt den Zusammenhang zwischen Systemvariablen und Antriebsparametern beim Auslösen und Quittieren der ESR-Reaktionen.



- ① NC: Freigabe der ESR-Reaktion über $\$AA_ESR_ENABLE = 1$ (achsspezifisch)
- ② NC: Auslösen der ESR-Reaktionen über $\$AN_ESR_TRIGGER = 1$
- ③ Antrieb: Das Auslösen der ESR-Reaktion wurde erkannt ($CU_STW.2 == 1$) \Rightarrow
 $r0887.12 = 1$ UND $r0887.13 = 1$
 Rückmeldung an die NC über $\$AA_ESR_STAT.BIT2 = 1$ (achsspezifisch)
- ④ NC: Quittieren der ESR-Reaktion durch $\$AA_ESR_ENABLE = 0$ (achsspezifisch)

Schwarz: Die ESR-Reaktion läuft zum Zeitpunkt ④ nicht mehr ($r0887.13 == 0$) \Rightarrow

- ⑤ Antrieb: "ESR ausgelöst" zurücksetzen: $r0887.12 = 0$
 Rückmeldung an die NC über $\$AA_ESR_STAT.BIT2 = 0$ (achsspezifisch)
- ⑥ NC: Die ESR-Reaktion kann erneut freigegeben werden \Rightarrow
 $\$AA_ESR_ENABLE = 1$ UND $\$AN_ESR_TRIGGER = 0$

Rot: Die ESR-Reaktion läuft zum Zeitpunkt ④ noch ($r0887.13 == 1$) \Rightarrow

- ⑤ Antrieb: "ESR ausgelöst" wird erst dann rückgesetzt ($r0887.12 = 0$), wenn die ESR-Reaktion im Antrieb beendet ist: $r0887.13 == 0$
 Rückmeldung an die NC über $\$AA_ESR_STAT.BIT2 = 0$ (achsspezifisch)
- ⑥ siehe ⑥ oben

Bild 11-6 Ablauf: Antriebsautarke ESR-Reaktionen auslösen und quittieren

11.3.9 ESR im Teileprogramm projektieren

Die Projektierung der antriebsautarken ESR-Funktionen "Stillsetzen" und "Rückziehen" kann durch die im Folgenden beschriebenen Befehle vom Teileprogramm aus verändert werden.

11.3.9.1 Stillsetzen (ESRS)

Syntax

ESRS (<Achse_1>, <Stillsetzzeit_1>[, ..., <Achse_n>, <Stillsetzzeit_n>])

Bedeutung

ESRS:	Mit der Funktion können die Antriebsparameter bezüglich der antriebsautarken ESR-Funktion "Stillsetzen" verändert werden. Besonderheiten: <ul style="list-style-type: none"> • Muss alleine im Satz stehen • Löst Vorlaufstopp aus • Kann nicht in Synchronaktionen verwendet werden • Maximal 5 Achsen pro Funktionsaufruf programmierbar; n = 5 	
<Achse_n>:	Schreibt für die angegebene Achse den Antriebsparameter p0888 (Konfiguration): p0888 = 1	
	Datentyp:	AXIS
	Wertebereich:	Kanalachsname
<Stillsetzzeit_n>:	Schreibt für die unter <Achse_n> angegebene Achse den Antriebsparameter p0892 (Zeitstufe): p0892 = <Stillsetzzeit_n>	
	Einheit:	s
	Datentyp:	REAL
	Wertebereich:	0.00 - 20.00

11.3.9.2 Rückziehen (ESRR)

Syntax

ESRR (<Achse_1>, <Rückzugsweg_1>, <Rückzugsgeschwindigkeit_1>[, ..., <Achse_n>, <Rückzugsweg_n>, <Rückzugsgeschwindigkeit_n>])

Bedeutung

ESRR:	Mit der Funktion können die Antriebsparameter bezüglich der antriebsautarken ESR-Funktion "Rückziehen" verändert werden. Besonderheiten: <ul style="list-style-type: none"> • Muss alleine im Satz stehen • Löst Vorlaufstopp aus • Kann nicht in Synchronaktionen verwendet werden • Maximal 5 Achsen pro Funktionsaufruf programmierbar; n = 5 	
<Achse_n>:	Schreibt für die angegebene Achse den Antriebsparameter p0888 (Konfiguration): p0888 = 2	
	Datentyp:	AXIS
	Wertebereich:	Kanalachsname
<Rückzugsgeschwindigkeit_n>:	Schreibt für die unter <Achse_n> angegebene Achse den Antriebsparameter p0893 (Rückzugsdrehzahl): p0893 = (<Rückzugsgeschwindigkeit> anhand der aktuell wirksamen Gesamtübersetzung umgerechnet in die entsprechende Rückzugsdrehzahl)	
	Einheit:	mm/min, inch/min, Grad/min (abhängig von der Einheit der Achse)
	Datentyp:	REAL
	Wertebereich:	0.00 - MAX
<Rückzugsweg_n>:	Schreibt für die unter <Achse_n> angegebene Achse den Antriebsparameter p0892 (Rückzugsdauer): p0892 = <Rückzugsweg_n> / <Rückzugsgeschwindigkeit_n>	
	Einheit:	mm, inch, Grad (abhängig von der Einheit der Achse)
	Datentyp:	REAL
	Wertebereich:	MIN - MAX

Hinweis

Rückzugsdrehzahl

Es ist sicher zu stellen, dass der Antrieb die programmierte Rückzugsdrehzahl (siehe Parameter <Rückzugsgeschwindigkeit_n>) innerhalb der Rückzugsdauer (siehe Parameter <Rückzugsweg_n>) erreichen kann. Es sind dazu die Einstellungen in folgenden Antriebsparametern zu prüfen:

- p1082[0...n] (Maximaldrehzahl)
- p1135[0...n] (AUS3 Rücklaufzeit)

Rückzugsweg

Die Umrechnung des Rückzugswegs in eine Rückzugsdauer für Antriebsparameter p0892 erfolgt unter der Annahme, dass die Rückzugsbewegung im Stillstand der Achse ausgelöst wird (siehe "Bild 11-5 Antriebsaut_Generatorbetrieb_projektieren (Seite 598)").

Erfolgt das Auslösen der Rückzugsbewegung während einer Verfahrbewegung der Achse, weicht durch die Überlagerung der Rückzugsbewegung mit der Verfahrbewegung der tatsächlich verfahrene Rückzugsweg vom programmierten Rückzugsweg ab.

Abhängigkeiten

Die programmierten Werte für den Rückzugsweg und die Rückzugsgeschwindigkeit beziehen sich auf die Lastseite. Vor dem Schreiben der Antriebsparameter werden diese auf die Motorseite umgerechnet. Für die Umrechnung gilt die zum Ausführungszeitpunkt der Funktion in der NC wirksame Übersetzung. Ist für eine Achse das antriebsautarke ESR freigegebenen ($\$AA_ESR_ENABLE[<Achse>] = 1$), ist eine Änderung der axialen Übersetzung nicht zulässig.

Hinweis

Änderung der Übersetzung

Ist für eine Achse das antriebsautarke ESR nicht freigegeben ($\$AA_ESR_ENABLE[<Achse>] = 0$), kann die axiale Übersetzung geändert werden. Es liegt dabei in der alleinigen Verantwortung des Anwenders/Maschinenherstellers, dass nach einer Änderung der Übersetzung der Rückzugsweg und die Rückzugsgeschwindigkeit entsprechend angepasst werden.

Siehe auch

Generatorbetrieb im Antrieb projektieren (Seite 597)

11.3.9.3 Randbedingungen

Konsistente Parameteränderung

Damit die Antriebsparameter konsistent wirksam werden, muss im Teileprogramm vor den Funktionen `ESRS (...)` und `ESRR (...)` die antriebsautarken ESR-Freigaben weggenommen werden.

Beispiel:

Programmcode	Kommentar
N100 <code>\$AA_ESR_ENABLE[X1]=0</code>	; ESR-Freigaben wegnehmen
N110 <code>\$AA_ESR_ENABLE[Z1]=0</code>	
N120 <code>\$AA_ESR_ENABLE[Y1]=0</code>	
N130 <code>ESRR(X1,20.0,10000.0,Z1,-15.0,20000.0)</code>	; ESR-Parameter ändern
N140 <code>ESRS(Y1,0.5,B1,0.5)</code>	
N150 <code>\$AA_ESR_ENABLE[X1]=1</code>	; ESR-Freigaben setzen
N160 <code>\$AA_ESR_ENABLE[Z1]=1</code>	
N170 <code>\$AA_ESR_ENABLE[Y1]=1</code>	

Bei Ausführung der Funktionen `ESRS (...)` und `ESRR (...)` wird die weitere Abarbeitung des Teileprogramms so lange angehalten, bis das Schreiben der Antriebsparameter quittiert worden ist. Als Rückmeldung für den Anwender wird während dieser Zeit die Meldung: "Warten auf Antriebsparameter von <Funktion>" angezeigt.

Satzsuchlauf mit Berechnung

Während Satzsuchlauf mit Berechnung werden die Funktionen `ESRS (...)` und `ESRR (...)` aufgesammelt und im Aktionssatz ausgeführt.

Satzsuchlauf mit Berechnung im Modus "Programmtest" (SERUPRO)

Während SERUPRO werden die Funktionen `ESRS (...)` und `ESRR (...)` sofort ausgeführt.

Reset-Verhalten

Die mit den Funktionen `ESRS (...)` und `ESRR (...)` geschriebenen Parameterwerte werden im Hochlauf des Antriebs nach Aus/Einschalten des Antriebs oder Antriebs-Warmstart mit den im Antrieb gespeicherten Parameterwerten überschrieben.

Das Speichern der Antriebsparameter kann über die HMI-Bedienoberfläche oder im Antrieb vorgenommen werden. Über die HMI-Bedienoberfläche:

- Bedienbereich "Inbetriebnahme" > "Maschinendaten" > "Antriebs-MD" > "Speichern/Reset" > "Speichern" > "Antriebsobjekt" **oder** "Antriebsgerät" **oder** "Antriebssystem"

Steuerungshochlauf (POWER ON)

Soll das antriebsautarke ESR bereits nach dem Hochlauf der Steuerung (POWER ON) aktiv sein, muss die Programmierung und Freigabe innerhalb eines von der PLC aus gestarteten ASUP oder über den PROGEVENT-Mechanismus erfolgen.

Literatur

Ausführliche Informationen zu ereignisgesteuerten Programmaufrufen finden sich in:

Funktionshandbuch Grundfunktionen, BAG, Kanal, Programmbetrieb, Reset-Verhalten (K1); Kapitel: Ereignisgesteuerte Programmaufrufe

11.3.10 ESR und Safety Integrated (840D sl)**Verzögerungszeiten**

Damit die antriebsautarken ESR-Reaktionen im Zusammenhang mit Safety Integrated vor den Safety-Reaktionen von STOP E, STOP F oder Kommunikationsausfall ausgeführt werden können, müssen entsprechende Verzögerungszeiten bezüglich der Safety-Reaktionen in der Steuerung und im Antrieb projektiert werden.

Die Projektierung der Verzögerungszeiten in der Steuerung erfolgt über folgende Maschinendaten:

- MD36954 \$MA_SAFE_STOP_SWITCH_TIME_E (Übergangszeit STOP E auf sicheren Stillstand)
- MD36955 \$MA_SAFE_STOP_SWITCH_TIME_F (Übergangszeit STOP F auf STOP B)
- MD36961 \$MA_SAFE_VELO_STOP_MODE (Stopp-Reaktion sichere Geschwindigkeit)
- MD36963 \$MA_SAFE_VELO_STOP_REACTION (Stopp-Reaktion sichere Geschwindigkeit)
- MD10089 \$MN_SAFE_PULSE_DIS_TIME_BUSFAIL (Wartezeit Impulslöschung bei Busausfall)

Die Projektierung der Verzögerungszeiten im Antrieb erfolgt über folgende Antriebsparameter:

- p9554 (Übergangszeit von STOP E auf "Sicheren Betriebshalt" (SOS))
 - p9555 (Übergangszeit STOP F auf STOP B)
 - p9561 (SG Stopp-Reaktion)
 - p9563 (SG-spezifisch Stopp-Reaktion)
 - p9580 (Wartezeit, nach der bei Busausfall die sichere Impulslöschung durchgeführt wird)
 - p9697 (Verzögerungszeit für die Impulslöschung nach Busausfall)
 - p9897 (Verzögerungszeit für die Impulslöschung nach Busausfall)
-

Hinweis

Durch die Safety Integrated Inbetriebnahmefunktion "SI-Daten kopieren" werden die Antriebsparameter automatisch mit den entsprechenden NC-Maschinendaten abgeglichen.

Antriebsautarke ESR-Reaktionen auslösen

Die Rückmeldung der aktuell im Antrieb aktiven Safety Stopp-Reaktion an die Steuerung erfolgt über folgende achsspezifische Systemvariable:

- \$VA_STOPSI[<Achse>] (aktuelle Stopp-Reaktion)

Durch Auswertung der Systemvariable in einem Teileprogramm/Synchronaktion kann anwenderspezifisch auf die jeweilige Stopp-Reaktion der Achse, z.B. durch Auslösen der antriebsautarken und/oder steuerungsgeführten ESR-Reaktionen über die Systemvariable \$AN_ESR_TRIGGER und/oder \$AC_ESR_TRIGGER bzw. \$AA_ESR_TRIGGER, reagiert werden.

Literatur

Eine ausführliche Beschreibung zur Projektierung und des Zusammenwirkens von antriebsautarken ESR-Reaktionen und Safety Integrated-Funktionen findet sich in:

Literatur

Funktionshandbuch SINUMERIK Safety Integrated

11.3.11 ESR und Safety Integrated (828D)

Keine Rückmeldung der Safety Stopp-Reaktion

Im Rahmen von SINUMERIK 828D wird die aktuell im Antrieb aktive Safety Stopp-Reaktion **nicht** an die Steuerung zurückgemeldet. Im Zusammenhang mit Safety Integrated können daher nur die antriebsautarken ESR-Funktionen verwendet werden.

Die vom Antrieb ausgelösten ESR-Reaktionen können auf Steuerungsseite über die achsspezifische Systemvariable \$AA_ESR_STAT (siehe Kapitel "Rückmeldung des ESR-Status (Seite 600)") ermittelt und anwenderspezifisch im Teileprogramm/Synchronaktion weitere ESR-Reaktionen über \$AN_ESR_TRIGGER ausgelöst werden..

Von Safety-Reaktionen unbeeinflusste ESR-Reaktionen

Damit die antriebsautarken ESR-Reaktionen im Zusammenhang mit Safety Integrated unbeeinflusst von parallel ablaufenden Safety-Reaktionen aufgrund von STOP E, STOP F oder Kommunikationsausfall ausgeführt werden können, müssen im Antrieb entsprechende Verzögerungszeiten bezüglich der Safety-Reaktionen **und** die entsprechenden Safety Trigger-Quellen für das Auslösen der antriebsautarken ESR-Reaktionen projektiert werden.

Verzögerungszeiten

Die Projektierung der Verzögerungszeiten im Antrieb erfolgt über folgende Antriebsparameter:

- p9554 (Übergangszeit von STOP E auf "Sicheren Betriebshalt" (SOS))
- p9555 (Übergangszeit STOP F auf STOP B)
- p9580 (Wartezeit, nach der bei Busausfall die sichere Impulslöschung durchgeführt wird)
- p9697 (Verzögerungszeit für die Impulslöschung nach Busausfall)

Safety Trigger-Quellen

Durch BICO-Verschaltung des Antriebsparameters p0890 (BI: ESR Trigger) mit den Safety-Reaktionen STOP E, STOP F und Kommunikationsausfall können zusätzlich die ESR-Reaktionen ausgelöst werden:

Parameter	BICO-Verschaltung mit:	Auswirkung: ESR-Reaktionen auslösen bei
p0890[1]	r9721[15]	Safety STOP E
p0890[2]	r9723[1]	Safety STOP F
p0890[3]	r9723[2]	Safety Kommunikationsausfall

Literatur

Eine ausführliche Beschreibung der Antriebsparameter und der BICO-Technik findet sich in:

- SINAMICS S120/S150 Listenhandbuch
- SINAMICS S120 Funktionshandbuch; Kapitel "Grundlagen der Antriebstechnik" > "BICO-Technik: Verschalten von Signalen"

11.4 Randbedingungen

Transformation

Ist das NC-geführte Erweiterte Stillsetzen und Rückziehen für **eine** an einer aktiven Transformation beteiligten Achse aktiviert, muss für **alle anderen** an der Transformation beteiligten Achsen ebenfalls Erweitertes Stillsetzen und Rückziehen aktiviert werden. Ansonsten kann es bei einer Aktivierung des Erweiterten Stillsetzens und Rückziehens zum Auflösen des Transformationsverbundes kommen.

Antriebskomponenten

Damit die projektierten Stillsetz- oder Rückzugsreaktionen ausgeführt werden können, müssen die beteiligten Antriebskomponenten voll funktionsfähig sein. Achsspezifische Servo- oder Antriebsalarmler, die den Ausfall bzw. Funktionsstörungen einer dieser Komponenten beschreiben, melden damit implizit, dass die projektierten Stillsetz- und Rückzugsreaktionen der betroffenen Achse nicht mehr oder nur teilweise verfügbar sind.

Bewegungssynchronaktionen

Bewegungssynchronaktionen werden im Interpolatortakt abgearbeitet. Eine Erhöhung des Interpolatortaktes, z.B. durch eine große Anzahl von aktiven Bewegungssynchronaktionen, führt daher zu einem größeren Zeitraster für das Auswerten der Auslösebedingungen und das Auslösen der Stillsetz- und Rückzugsreaktionen.

11.5 Beispiele

11.5.1 NC-geführte Reaktionen

Aufgabe

Die Achse A soll als Generator zur Verfügung stehen, Achse X soll im Fehlerfall mit maximaler Geschwindigkeit um 10 mm zurückziehen. Die Achsen Y und Z sollen um 100 ms verzögert stillgesetzt werden, damit die Rückzugsachse X ausreichend Zeit hat, die mechanische Kopplung aufzuheben.

Das Beispiel erfolgt unter Verwendung der NC-geführten Reaktionen.

Es werden nur die wesentlichen Teile der Gesamtfunktionalität aufgezeigt.

Voraussetzungen

Folgende Funktionen müssen zur Verfügung stehen:

- Erweitertes Stillsetzen und Rückziehen (ESR)
- Statische Synchronaktionen
- Asynchrone Unterprogramme (ASUP)

Einstellungen

Maschinendaten

Achsspezifische Betriebsart des ESR

- MD37500 \$MA_ESR_REACTION[X] = 21 (NC-geführte Rückzugsachse)
- MD37500 \$MA_ESR_REACTION[Y] = 22 (NC-geführte Stillsetzachse)
- MD37500 \$MA_ESR_REACTION[Z] = 22 (NC-geführte Stillsetzachse)

Konfiguration: NC-geführtes Erweitertes Stillsetzen

- MD21380 \$MC_ESR_DELAY_TIME1 = 0.1 (Fortdauer der Bahn-Interpolation in Sekunden)
- MD21381 \$MC_ESR_DELAY_TIME2 = 0.04 (Bremsdauer in Sekunden)

Programmierung

Programmcode	Kommentar
LFPOS	Achsspez. Abheben auf ein Position
POLF[X]=IC(10)	Rückzugposition, inkrementell
POLFMASK(X)	Rückziehen freigeben
; Freigabe des "Erweiterten Stillsetzen und Rückziehens"	
\$AA_ESR_ENABLE[X] = 1	; Achse X
\$AA_ESR_ENABLE[Y] = 1	; Achse Y
\$AA_ESR_ENABLE[Z] = 1	; Achse Z

Triggerbedingungen und statische Synchronaktionen

Beispiel 1

Triggerbedingung ist das Auftreten von Alarmen, die Nachführbetrieb auslösen:

Programmcode

```
IDS=02 WHENEVER ($AC_ALARM_STAT B_AND 'H2000') > 0 DO $AC_ESR_TRIGGER=1
```

Beispiel 2

Triggerbedingung ist das Ansprechen der ELG-Synchronlaufüberwachung, wenn z. B. Y als ELG-Folgeachse definiert und die max. zulässige Synchronlaufabweichung 100 mm ist:

Programmcode

```
IDS=03 WHENEVER $VA_EG_SYNCDIFF[Y] > 0.1 DO $AC_ESR_TRIGGER=1
```

11.5.2 Rückziehen während Gewindeschneiden

Während des Gewindeschneiden soll auf einen Fehler/Unterbrechung mit Rückziehen der Achse X auf die unter POLF angegebene Position reagiert werden. Die Achse Z verfährt bis zu ihrem Stillsetzen normal weiter.

Programmcode	Kommentar
N10 G0 G90 X200 Z0 S200 M3	;
N20 G0 G90 X170	;
N30 POLF[X]=210	; Rückzugsposition Achse X, absolut
N40 LFPOS	; Rückziehen über POLF/POLFMASK EIN
N50 POLFMASK(X)	; Freigabe Rückziehen Achse X
N60 LFON	; Schnelrückzug für Gewindeschneiden EIN
	; Gewindeschneiden
N70 G33 X100 I10	; Rückzugsreaktion, axial: X (abs.)
N80 X130 Z-45 K10	; dto.
N90 X155 Z-128 K10	; dto.
N100 X145 Z-168 K10	; dto.
N110 X120 I10	; dto.
N120 G0 Z0 LFOF	; Schnelrückzug für Gewindeschneiden AUS
N130 POLFMASK()	; Sperren Rückziehen alle Achsen

11.5.3 Schnellabheben über ASUP und schnellen Eingang

ASUP-Aktivierung durch den Programmbefehl LIFTFAST (Schnellabheben) über den schnellen Eingang 1:

Programmcode	Kommentar
N10 SETINT (1) PRIO=1 ABHEB_Y LIFTFAST	; ASUP-Aktivierung, schneller Eingang 1
N20 LFPOS	; Rückziehen über POLF/POLFMASK EIN
N30 POLF[X]=19.5 POLF[Y]=33.3	; Rückzugspositionen Achsen X Y, absolut
N40 POLF[Z]=100	; Rückzugsposition Achse Z, absolut
N50 X0 Y0 G0	;
N60 POLFMASK(X, Y)	; Freigabe Rückziehen Achse X Y
N70 Z100 G1 F1000	; Rückzugsreaktion, axial: X, Y (abs.)
N80 POLFMASK(Z)	; Sperren Rückziehen Achse X Y und
	; Freigabe Rückziehen Achse Z
N90 Y10	; Rückzugsreaktion, axial: Z (abs.)
N100 POLFMASK()	; Sperren Rückziehen alle Achsen

11.5.4 Schnellabheben absolut und inkrementell

Rückziehen auf absolute Positionen und um einen inkrementellen Weg:

Programmcode	Kommentar
N10 \$AA_ESR_ENABLE[X]=1	; Freigabe Rueckziehen, Achse X
N20 \$AA_ESR_ENABLE[Z]=1	; Freigabe Rueckziehen, Achse Z
N30 \$AA_ESR_ENABLE[Y]=1	; Freigabe Rueckziehen, Achse Y
N40 LFPOS	; Rückziehen über POLF/POLFMASK EIN
N50 POLF[X]=IC(3.0) POLF[Y]=-4.0	; Rückzugsposition Achse X, inkrementell
	; Rückzugsposition Achse Y, absolut
N60 POLF[Z]=100	; Rückzugsposition Achse Z, absolut
N70 G0 X0 Y0 Z0	;
N80 POLFMASK(X, Y)	; Freigabe Rückziehen Achse X Y
N90 Z100 G1 F1000	; Rückzugsreaktion, axial: X (ink.), Y (abs.)
N100 POLF[X]=10	; Rückzugsposition Achse X, absolut
N110 Y0 G1 F1000	; Rückzugsreaktion, axial: X, Y (abs.)
N120 POLFMASK(Z)	; Sperren Rückziehen Achse X Y und
	; Freigabe Rückziehen Achse Z
N130 Y10	; Rückzugsreaktion, axial: Z (abs.)
N140 POLFMASK()	; Sperren Rückziehen alle Achsen

11.5.5 Schnellabheben im linearen Zusammenhang der Achsen

Rückziehen im linearen Zusammenhang, absolut und inkrementell

Programmcode	Kommentar
N10 \$AA_ESR_ENABLE[X]=1	; Freigabe Rueckziehen, Achse X
N20 \$AA_ESR_ENABLE[Y]=1	; Freigabe Rueckziehen, Achse Y
N30 \$AA_ESR_ENABLE[Z]=1	; Freigabe Rueckziehen, Achse Z
N40 LFPOS	; Rückziehen über POLF/POLFMASK EIN
N50 POLF[X]=IC(3.0) POLF[Z]=-4.0	; Rückzugsposition Achse X, inkrementell
	; Rückzugsposition Achse Z, absolut
N60 POLF[Y]=100	; Rückzugsposition Achse Z, absolut
N70 X0 Y0 Z0 G0	; ---
N80 POLFMLIN(X, Y)	; Freigabe Rückziehen Achse X Y, linearer
	; Zusammenhang
N85 POLFMASK(Z)	; Freigabe Rückziehen Achse Z
N90 Z100 G1 F1000	; Rückzugsreaktion:
	; - linearer Zusammenh.: X (ink.), Y (abs.)
	; - axial: Z (abs.)
N95 POLF[X]=10	; Rückzugsposition Achse X, absolut
N100 Y0 G1 F1000	; Rückzugsreaktion:
	; - linearer Zusammenh.: X (abs.), Y (abs.)
	; - axial: Z (abs.)

11.6 Datenlisten

Programmcode	Kommentar
N110 POLFMLIN()	; Sperren Rückziehen Achse X Y
N120 Y10	; Rückzugsreaktion, axial: Z (abs.)
N130 POLFMASK()	; Sperren Rückziehen alle Achsen

11.6 Datenlisten

11.6.1 Maschinendaten

11.6.1.1 Kanal-spezifische Maschinendaten

Nummer	Bezeichner: \$MC_	Beschreibung
21204	LIFTFAST_STOP_COND	Stoppverhalten beim Schnellabheben
21380	ESR_DELAY_TIME1	Verzögerungszeit (STOPBYALARM, NOREAD) für ESR-Achsen
21381	ESR_DELAY_TIME2	Zeit für interpolatorisches Bremsen für ESR-Achsen

11.6.1.2 Achs-/Spindel-spezifische Maschinendaten

Nummer	Bezeichner: \$MA_	Beschreibung
37500	ESR_REACTION	Reaktions-Definition beim Erweiterten Stillsetzen und Rückziehen

11.6.2 Systemvariablen

Bezeichner	Bedeutung
\$A_DBB	Datenbyte von/an PLC lesen/schreiben
\$A_IN	digitaler Eingang
\$A_OUT	digitaler Ausgang
\$AA_ESR_ENABLE[<Achse>]	Freigabe "Erweitertes Stillsetzen und Rückziehen"
\$AA_TYP[<Achse>]	Achstyp
\$AA_ESR_TRIGGER[<Achse>]	ESR auslösen für PLC kontrollierte Achse (Einzelachse)
\$AC_ALARM_STAT	Alarmstatus im Kanal
\$AC_ESR_TRIGGER	ESR auslösen, kanalspezifisch
\$AC_STAT	Kanalzustand

11.6.3 Signale

11.6.3.1 Signale an Kanal

Signalname	SINUMERIK 840D sl	SINUMERIK 828D
Vorschubsperr	DB21, ... DBX6.0	

11.6.3.2 Signale an Achse/Spindel

Signalname	SINUMERIK 840D sl	SINUMERIK 828D
Vorschub Halt	DB31, ... DBX4.3	

11.6.3.3 Signale von Achse/Spindel

Signalname	SINUMERIK 840D sl	SINUMERIK 828D
ESR: Zwischenkreisunterspannung (p1248)	DB31, ... DBX95.1	DB390x.DBX4003.1
ESR: Reaktion ausgelöst / Generatorbetrieb aktiv (r0887.12)	DB31, ... DBX95.2	DB390x.DBX4003.2
ESR, Generatorbetrieb: Minimaldrehzahl unterschritten (p2161)	DB31, ... DBX95.3	DB390x.DBX4003.3

S9: Sollwertumschaltung - nur 840D sl

12.1 Kurzbeschreibung

Funktion

Die Funktion "Sollwertumschaltung" wird benötigt, wenn nur ein Motor zum Antrieb mehrerer Achsen/Spindeln benutzt werden soll. Z. B. bei Fräsköpfen, bei denen der Spindelmotor sowohl zum Antrieb des Werkzeugs als auch zur Orientierung des Fräskopfs verwendet wird.

Voraussetzungen

Damit eine Sollwertumschaltung durchgeführt wird, müssen folgende Voraussetzungen erfüllt sein:

- Alle an der Sollwertumschaltung beteiligten Achsen sind im Stillstand
- Am PROFIBUS steht kein Lebenszeichenfehler und keine Störung an
- Es ist keine der folgenden Funktionen in der Achse mit Antriebskontrolle aktiv:
 - Referenzpunktfahren
 - Messen
 - Fahren auf Festanschlag
 - Funktionsgenerator
 - Stern-/Dreieckumschaltung
 - Parametersatzumschaltung

Lageregelkreis

Während der Sollwertumschaltung wird der Antriebsstrang und damit auch der Lageregelkreis getrennt. Um Instabilitäten zu vermeiden, wird die Umschaltung nur im Stillstand und mit gelöschten Reglerfreigaben durchgeführt.

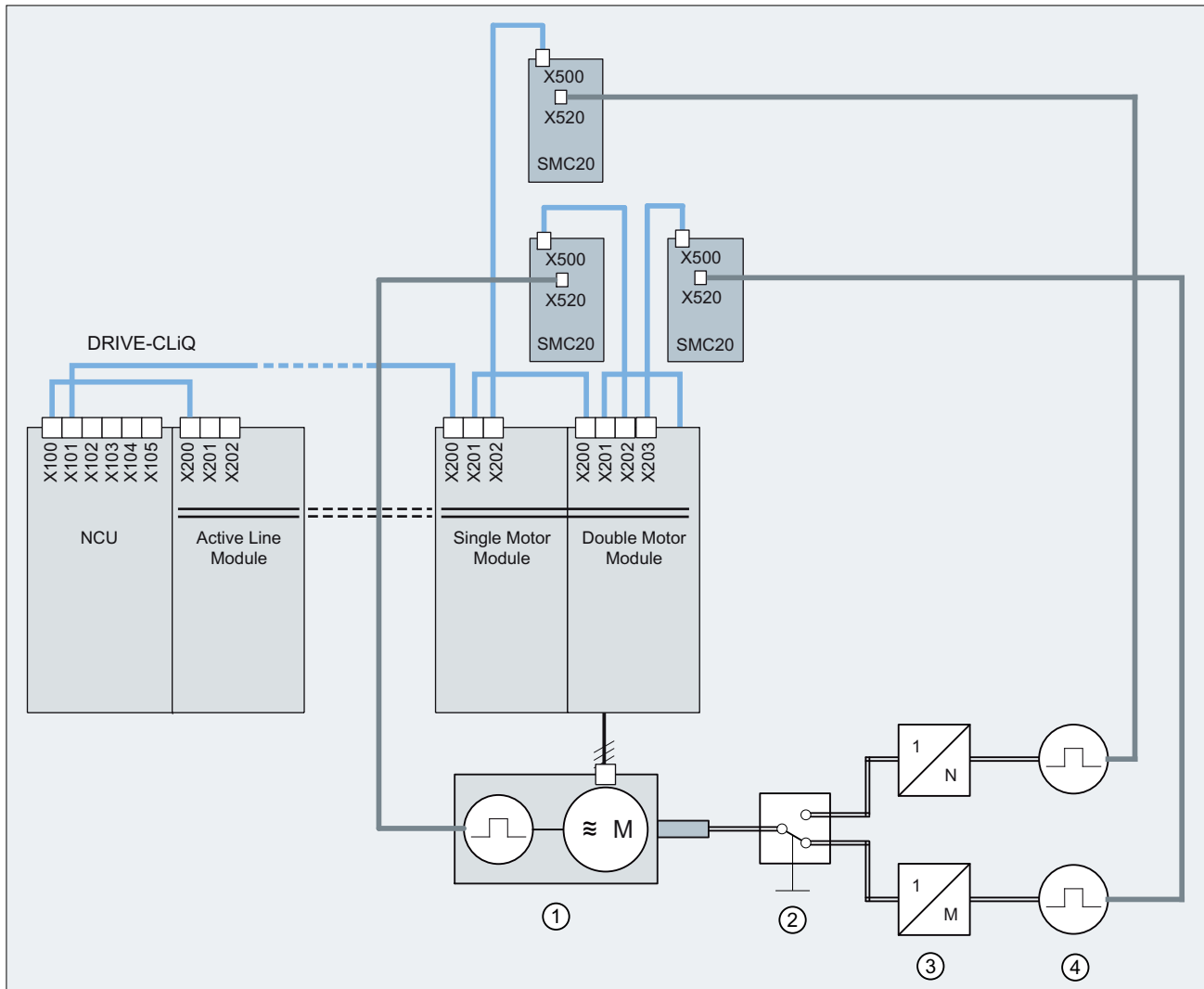
Durch die Benutzung eines einzelnen Antriebs kann zum selben Zeitpunkt nur einer der Regelkreise geschlossen sein. Achsen ohne Antriebskontrolle werden mit offenem Lageregler und nachgeführten Positionen betrieben.

Hinweis

Hängende Achsen

Für die Dauer der Umschaltung werden die Reglerfreigaben aller beteiligten Achsen automatisch durch die Steuerung weggenommen. Achsen ohne vorhandene Antriebskontrolle sind nicht in Regelung. Für hängende Achsen sollte daher eine Bremsensteuerung realisiert werden.

Beispiel



- ① Motor mit Geber
- ② Mechanische Umschaltvorrichtung
- ③ Getriebe 1
- ④ Geber 1 (z. B. für Fräskopf)
- ⑤ Getriebe 2
- ⑥ Geber 2 (z. B. für Spindel)

Bild 12-1 Sollwertumschaltung mit 2 Achsen

Ablösung der Technologiefunktion "Sollwertumschaltung" (TE5)

Die Funktion "Sollwertumschaltung" löst die Technologiefunktion "Sollwertumschaltung" (TE5) ab. Eine Migration der Technologiefunktion auf die Funktionalität der Funktion "Sollwertumschaltung" erfordert Anpassungen in Maschinendaten und im PLC-Anwenderprogramm:

- Das Maschinendatum MD63750 entfällt.
- Die Bedeutungen verschiedener NC/PLC-Nahtstellensignale wurden geändert.
- Die Alarme 70451 und 70452 entfallen.
- Eine Sollwertumschaltung mit simulierten Achsen (MD30130 = 0) wird nicht mehr unterstützt.
- Bekannte Einschränkungen der Technologiekartenfunktion entfallen.

12.2 Inbetriebnahme

Maschinendaten

Sollwertzuordnung

Zur Festlegung der Achsen, die an der Sollwertumschaltung teilnehmen, wird der Sollwertkanal eines Antriebs mehrfach zugeordnet. Hierfür muss das folgende Maschinendatum für jede Achse mit der logischen Nummer dieses Antriebs gleich vorbelegt werden:

MD30110 \$MA_CTRL_OUT_MODULE_NR (Sollwertzuordnung: Baugruppennummer)

Hinweis

Bei fehlender Option wird der Alarm 26018 ausgegeben.

Geberzuordnung

Die Geberzuordnung erfolgt achsspezifisch im Maschinendatum:

MD30230 \$MA_ENC_INPUT_NR (Istwertzuordnung: Eingang auf Antriebsmodul/
Messkreiskarte)

NC/PLC-Nahtstellensignale

Hinweis

Zugriffsrechte

Die Verwendung der achsspezifischen NC/PLC-Nahtstellensignale bleibt trotz der Zuordnung eines Antriebs zu mehreren Maschinenachsen unverändert. Dies erfordert im PLC-Anwenderprogramm eine explizite Koordinierung der Zugriffe auf die NC/PLC-Nahtstellensignale der einzelnen an der Sollwertumschaltung beteiligten Maschinenachsen.

Anforderung der Antriebskontrolle

Die Anforderung zur Übernahme der Antriebskontrolle erfolgt über:

DB31, ... DBX24.5 (Sollwertumschaltung: Anforderung Antriebskontrolle)

Zustand der Antriebskontrolle

Der aktuelle Zustand der Antriebskontrolle wird angezeigt über:

DB31, ... DBX96.5 (Sollwertumschaltung: Antriebskontrolle aktiv)

Statussignale

Nach der erstmaligen Übernahme der Antriebskontrolle für eine der an der Sollwertumschaltung beteiligten Maschinenachsen (DB31, ... DBX24.5 = 1 (Sollwertumschaltung: Umschalten)) werden folgende Statussignale für alle beteiligten Maschinenachsen immer gleich angezeigt:

DB31, ... DBB92 - DBB95

Steuersignale

Die folgenden Steuersignale wirken ausschließlich in der Maschinenachse, die aktuell die Antriebskontrolle hat (DB31, ... DBX96.5 == 1 (Sollwertumschaltung aktiv)):

DB31, ... DBB20 bis DBB21

Reglerfreigabe

Die Reglerfreigabe (DB31, ... DBX2.1) ist nur in der Maschinenachse wirksam, welche aktuell die Antriebskontrolle hat (DB31, ... DBX96.5 == 1 (Status Sollwertumschaltung))

Nachführbetrieb

Die an der Sollwertumschaltung beteiligten Maschinenachsen **ohne** Antriebskontrolle werden steuerungsintern immer in den Nachführbetrieb geschaltet (DB31, ... DBX61.3 == 0).

Hinweis

Eindeutigkeit

Zu jedem Zeitpunkt sollte es immer eine und nur eine Achse geben, die über die Antriebskontrolle verfügt DB31, ... DBX24.5 == 1 und DB31, ... DBX96.5 == 1.

Sonderfälle

Mehrere gleichzeitige Übernahmeanforderungen

Bei mehreren gleichzeitigen Übernahmeanforderungen findet keine Umschaltung statt. Die Kontrolle über den Antrieb verbleibt bei der zuletzt verwendeten Achse. Das Fehlen von Übernahmeanforderungen hat die gleiche Auswirkung.

Fehlende Übernahmeanforderung im Hochlauf

Bei fehlender Übernahmeanforderung im Hochlauf der Steuerung, wird die Antriebskontrolle **steuerungsintern** der ersten gefundenen Maschinenachse mit der gleichen logischen Antriebsnummer zugewiesen.

MD30110 \$MA_CTRLOUT_MODULE_NR[0,AX1] = 1

MD30110 \$MA_CTRLOUT_MODULE_NR[0,AX2] = 2

MD30110 \$MA_CTRLOUT_MODULE_NR[0,AX3] = 3

MD30110 \$MA_CTRLOUT_MODULE_NR[0,AX4] = 4 ; Sollwertumschaltung: 1.
Achse

MD30110 \$MA_CTRLOUT_MODULE_NR[0,AX5] = 4 ; Sollwertumschaltung: 2.
Achse

Um mit der Achse den Antrieb verfahren zu können, muss vorher das NC/PLC-Nahtstellensignal zur Anforderung der Antriebskontrolle (DB31, ... DBX24.5) gesetzt werden.

12.3 Ablaufdiagramm

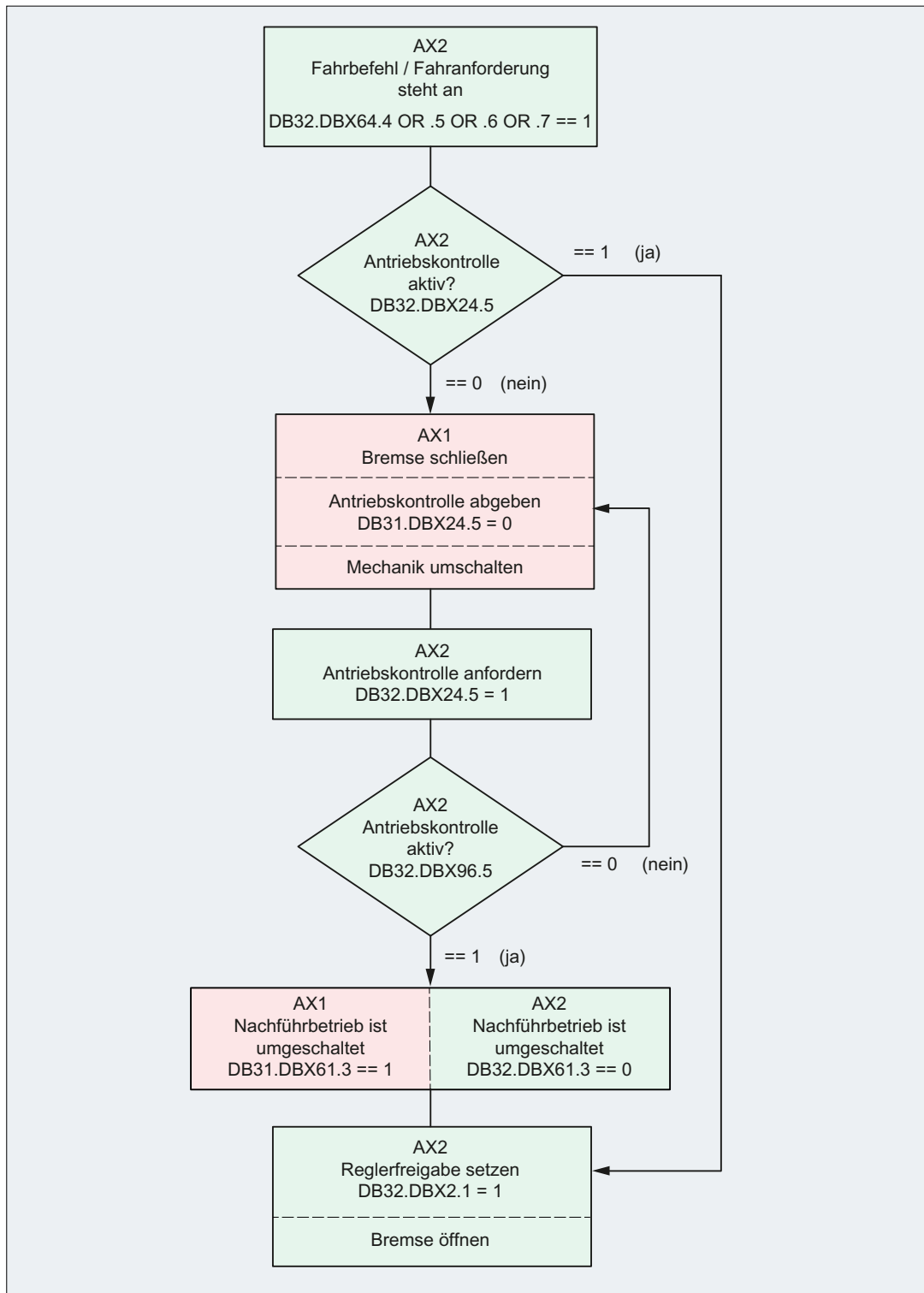


Bild 12-2 Ablauf einer Sollwertumschaltung von Maschinenachsen AX1 nach AX2

12.4 Randbedingungen

Alarmer

Antriebsalarmer werden nur von Achsen mit aktiver Antriebskontrolle angezeigt.

Lageregelkreis

Während der Sollwertumschaltung wird der Antriebsstrang und damit auch der Lageregelkreis getrennt. Um Instabilitäten zu vermeiden, wird die Umschaltung nur im Stillstand und mit gelöschten Reglerfreigaben durchgeführt.

Durch die Benutzung eines einzelnen Antriebs kann zum selben Zeitpunkt nur einer der Regelkreise geschlossen sein. Achsen ohne Antriebskontrolle werden mit offenem Lageregler und nachgeführten Positionen betrieben.

Referenzpunkte

Beim Einsatz lastseitiger Geber bleiben achsspezifische Referenzpunkte von einer Sollwertumschaltung unberührt.

Im Gegensatz dazu kann bei einer lastseitigen Position, die sich ausschließlich aus dem Motorgeber herleitet, der mechanische Bezug zur Last nach einer Sollwertumschaltung verloren gehen. Solche Achsen müssen nach jeder Sollwertumschaltung erneut referenziert werden.

Parametersatzumschaltung

Eine Sollwertumschaltung wird nicht ausgeführt, wenn einer der folgenden Zustände vorliegt:

- Eine Parametersatzumschaltung in einer der beiden Maschinenachsen ist nicht abgeschlossen.
- Die Parametersätze der beiden Maschinenachsen sind nicht gleich.

Hinweis

Wenn eine Sollwertumschaltung aufgrund von Randbedingungen nicht ausgeführt wird, wird **keine** Meldung angezeigt.

Zustand "Parken"

Befindet sich eine Achse im Zustand "Parken", kann der Zustand nur dann wieder aufgehoben werden, wenn diese Achse die Antriebskontrolle hat.

Serviceanzeige Antrieb

Das HMI-Diagnosebild "Serviceanzeige Antrieb" berücksichtigt die wechselnde Zuordnung zwischen Maschinenachsen und Antrieb **nicht**.

Inbetriebnahme über SinuCom NC

Über das Inbetriebnahmetool "SinuCom NC" ist die Inbetriebnahme der Sollwertumschaltung nur über die Expertenliste möglich.

Safety Integrated (nur 840D sl)

Eine ausführliche Beschreibung zu den Randbedingungen bei Sollwertumschaltung in Zusammenhang mit Safety Integrated findet sich in:

Literatur:

Handbuch SINUMERIK Safety Integrated

12.5 Datenlisten

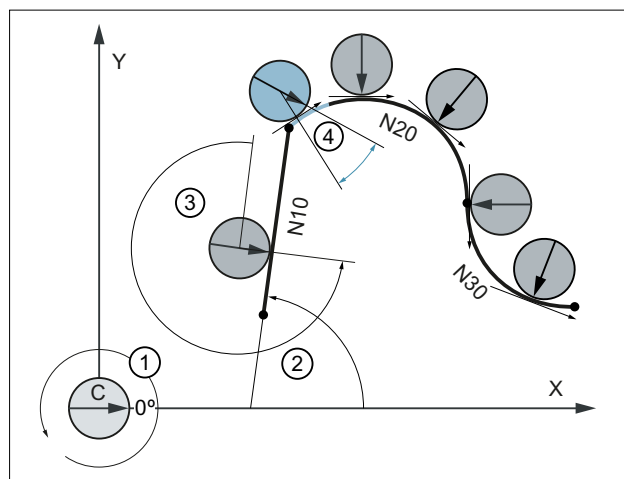
12.5.1 Maschinendaten

12.5.1.1 Achs-/Spindel-spezifische Maschinendaten

Nummer	Bezeichner: \$MA_	Beschreibung
30130	CTRLOUT_TYPE	Ausgabeart des Sollwertes
30200	NUM_ENCS	Anzahl der Geber
30220	ENC_MODULE_NR	Istwertzuordnung: Antriebsnummer / Messkreisnummer
30230	ENC_INPUT_NR	Istwertzuordnung: Eingang auf Antriebsmodul / Messkreiskarte

T3: Tangentialsteuerung - nur 840D sl

Durch die Funktion "Tangentialsteuerung" wird eine Rundachse als Folgeachse auf zwei Geometrieachsen als Leitachsen so gekoppelt, dass die Ausrichtung der Folgeachse eine Funktion der Bahntangente der Leitachsen ist.



- ① Grundstellung und positive Drehrichtung der Folge- / Rundachse C
- ② Winkel der Bahntangente in der Bearbeitungsebene X/Y
- ③ Offsetwinkel = 270° bzw. -90°
- ④ Überschleifweg und momentane Winkelabweichung

Im Bild sind die bahnbestimmenden Leitachsen die Geometrieachsen X und Y und die Rundachse C die Folgeachse der Tangentialkopplung. Die Rundachse wird mit einem programmierbaren Offsetwinkel von 270° zur Bahntangente nachgeführt. Beim Satzübergang von N10 zu N20 enthält die Kontur einen nichtstetigen Übergang bzw. Ecke. Aufgrund einer zu geringen Dynamik, kann die Folgeachse über einen Bahnbereich (blau) der Bahntangente nicht folgen. Die Dynamik der Folgeachse ist aber ausreichend, um im weiteren Verlauf der Kreissätze N20 und N30 der Bahntangente exakt zu folgen.

Eigenschaften

Die Tangentialkopplung zeichnet sich durch folgende Eigenschaften aus:

- Die Leitachsen der Tangentialkopplung müssen zwei Geometrieachsen sein.
- Die Folgeachse der Tangentialkopplung muss eine Rundachse sein.
- Die Tangentialkopplung kann im NC-Programm definiert, ein- und ausgeschaltet und gelöscht werden.
- Die Tangentialkopplung ist für das Basis- und das Werkstückkoordinatensystem definiert.
- Die positive Richtung der Bahntangente ist gleich der Verfahrrichtung der Bahn.
- Bei einem Offsetwinkel von 0° zwischen der Rundachse und der Bahntangente, zeigt die 0° -Richtung.

- Die Achsen der Tangentialkopplung müssen Kanalachsen des gleichen Kanals sein.
- Die Position der Folgeachse kann Eingangswert für eine weitere Transformation sein.
- Die Tangentialkopplung ist nur in den Betriebsarten AUTOMATIK und MDA aktiv.

Anwendungsbeispiele

- Tangentiales Anstellen eines drehbaren Werkzeugs beim Nibbeln.
- Nachführen der Werkstückausrichtung bei einer Bandsäge.
- Anstellen eines Abrichtwerkzeugs an eine Schleifscheibe.
- Anstellen eines Schneidrädchens zur Glas- oder Papierverarbeitung.
- Tangentiale Zuführung eines Drahts beim 5-achsigen Schweißen.

Eckenverhalten

Hat die durch die Leitachsen beschriebene Kontur einen unsteten Satzübergang bzw. Ecke, kann zwischen folgendem Eckenverhalten gewählt werden:

- Die Dynamik der Rundachse hat keine Rückwirkung auf die Leitachsen.
- Die Dynamik der Rundachse wird bei der Bahnplanung der Leitachsen zusammen mit den programmierbaren Parametern "Überschleifweg" und "Winkeltoleranz" berücksichtigt.
- Die Leitachsen werden vor der Ecke angehalten und die Folgeachse in einem automatisch generierten Zwischensatz neu ausgerichtet.
Die Eckenerkennung erfolgt abhängig vom Wert des Maschinendatums MD37400 \$MA_EPS_TLIFT_TANG_STEP (Tangentenwinkel für Eckenerkennung).

13.1 Inbetriebnahme

Maschinendaten

Eckenerkennung TLIFT()

Im Zusammenhang mit der Funktion Eckenerkennung TLIFT() (Seite 627) wird eine Unstetigkeit in der programmierten Bahn der Leitachsen als Ecke erkannt, wenn die Änderung des Tangentenwinkels für die Folgeachse an dieser Stelle größer als der im Maschinendatum angegebene Wert ist:

MD37400 \$MA_EPS_TLIFT_TANG_STEP[<Folgeachse>] = <Tangentenänderungswinkel>

Parametrierbarer Offsetwinkel

Über das Maschinendatum wird für die Folgeachse ein parametrierbarer Offsetwinkel vorgegeben:

Maschinendatum MD37402 \$MA_TANG_OFFSET[<Folgeachse>] = <parametrierbarer Offsetwinkel>

Der Offsetwinkel der Folgeachse ist der Winkel zwischen der Nullstellung der Folgeachse und der Bahntangente der programmierten Bahn der Leitachsen bei aktiver Tangentenkopplung.

Der wirksame Offsetwinkel ist die Summe aus dem im Maschinendatum parametrisierten Offsetwinkel und dem beim Einschalten der Tangentialkopplung mit TANGON() (Seite 629) programmierten Offsetwinkel:

Satzsuchlauf-Einstellungen

Über das Maschinendatum wird die Geschwindigkeit ausgewählt, mit der die Tangentialachse beim Einschalten der Tangentialkopplung während oder nach einem Satzsuchlauf auf ihre Position verfährt:

- Eilgangsgeschwindigkeit (G0)
- Programmierte Geschwindigkeit (G1 F....)
- Settingdatum SD42121 \$SC_AX_ADJUST_FEED ≠ 0

MD11450 \$MN_SEARCH_RUN_MODE, Bit 7

Siehe auch: Randbedingungen (Seite 633)

Systemvariablen

Satztyp \$AC_BLOCKTYPE

Über die Systemvariable \$AC_BLOCKTYPE kann ermittelt werden, ob der aktuelle Satz ein durch TLIFT() erzeugter Zwischensatz ist. Der aktuelle Satz ist ein durch TLIFT() erzeugter Zwischensatz, wenn gilt:

\$AC_BLOCKTYPE == 6

13.2 Programmierung

13.2.1 Kopplung definieren (TANG)

Über die vordefinierte Prozedur TANG(...) wird eine Tangentialkopplung zwischen einer Rundachse als Folgeachse und zwei Geometrieachsen als Leitachsen definiert. Die Folgeachse wird dabei kontinuierlich zur Bahntangente der Leitachsen ausgerichtet.

Hinweis

Koppelfaktor

Ein Koppelfaktor von 1 muss nicht explizit programmiert werden.
Durch den Koppelfaktor -1 wird die Richtung der Tangentialachse gedreht.

Syntax

TANG(<Folgeachse>, <Leitachse_1>, <Leitachse_2>, <Koppelfaktor>, <Koordinatensystem>, <Optimierung>)

Bedeutung

TANG(...):	Definieren einer Tangentialkopplung	
<Folgeachse>:	Achsenname der Folgeachse (Rundachse)	
	Datentyp:	AXIS
	Wertebereich:	Kanalachsnamen
<Leitachse_1> <Leitachse_2>:	Achsenamen der Leitachsen (Geometrieachsen) ¹⁾	
	Datentyp:	AXIS
	Wertebereich:	Geometrieachsnamen des Kanals
<Koppelfaktor>:	Faktor n der Winkeländerung der Folgeachse zur Änderung der Bahntangente der Leitachsen: Winkeländerung _{Folgeachse} = Winkeländerung _{Bahntangente} * n	
	Datentyp:	REAL
	Defaultwert:	1,0
<Koordinatensystem>:	Wirksames Koordinatensystem ²⁾	
	Datentyp:	CHAR
	Wert:	"B":
"W":		Werkstückkoordinatensystem (nicht verfügbar)

<Optimierung>:	Optimierungsart	
	Datentyp:	CHAR
	Wert:	<p>"S": Standard (Defaultwert) Die Dynamik der Rundachse hat keine Rückwirkung auf die Leitachsen. Ist die Dynamik der Rundachse höher als für die Nachführung erforderlich, ist dieses Verfahren ausreichend genau. Ist die Dynamik der Rundachse nicht hoch genug, um der Änderung der Bahntangenten zu folgen, weicht die Ausrichtung der Rundachse entlang eines nicht definierten Überschleifwegs von der Sollausrichtung ab.</p> <p>"P": Die Dynamik der Rundachse wird bei der Bahnplanung der Leitachsen berücksichtigt. Dazu müssen beim Einschalten der Tangentialkopplung mit TANGON() zwei zusätzliche Parameter angegeben werden:</p> <ul style="list-style-type: none"> • Überschleifweg • Winkeltoleranz <p>Siehe Kapitel "Kopplung einschalten (TANGON) (Seite 629)"</p> <p>Hinweis Im Zusammenhang mit kinematischen Transformationen wird empfohlen, die Optimierungsart "P" zu verwenden.</p>
<p>Hinweis Defaultwerte müssen nicht explizit programmiert werden.</p> <p>1) Hinweis Als Leitachsen der Tangentialkopplung müssen die Geometrieachsen verwendet werden, die, bezogen auf die Grundstellung der Maschine, im Maschinenkoordinatensystem (MKS) die programmierte Bahn abfahren. Wird z. B. auf einer Fräsmaschine mit Schwenkkopf der Schwenkzyklus CYCLE800 verwendet, erfolgt, abhängig von der Konfiguration des Zyklus, die Interpolation im WKS z. B. mit den Geometrieachsen X und Y. Die Tangentialkopplung muss aber mit den Geometrieachsen als Leitachsen definiert werden, die im MKS die programmierte Bahn abfahren. Dazu sind als Leitachsen die Geometrieachsen im ungeschwenkten Zustand der Maschine zu verwenden.</p> <p>2) Hinweis Das Basiskoordinatensystem (BKS) darf gegenüber dem MKS nicht gedreht sein. Wird z. B. über den Befehl ROT oder über den Schwenkzyklus CYCLE800 das BKS gedreht, arbeitet die Tangentialsteuerung nicht mehr korrekt.</p>		

13.2.2 Zwischensatzerzeugung einschalten (TLIFT)

Überschreitet die Tangentenänderung der Folgeachse an einer Stelle der programmierten Bahn der Leitachsen den im Maschinendatum MD37400 \$MA_EPS_TLIFT_TANG_STEP parametrisierten Grenzwert, erfolgt die weitere Bahnplanung abhängig vom eingestellten Eckenverhalten. Ohne Verwendung der vordefinierten Prozedur TLIFT(...) wird entsprechend des im Zusammenhang mit TANG(...) (Seite 625) und TANGON(...) (Seite 629) programmierten Überschleifenverhaltens verfahren.

Zwischensatzerzeugung einschalten

Mit Programmierung von TLIFT(...) im Anschluss an TANG(...) wird vom Vorlauf beim Erkennen einer Ecke an dieser Stelle der Bahn ein von der Steuerung automatisch generierter Zwischensatz eingefügt.

Bei der Abarbeitung des Programms werden dann bei Erreichen des Zwischensatzes die Leitachsen angehalten. Im Zwischensatz wird die Folgeachse mit maximaler Achsdynamik in Richtung der Bahntangente des nachfolgenden Satzes gedreht. Danach werden die Leitachsen weiter auf der programmierten Bahn verfahren.

Zwischensatzerzeugung ausschalten

Zum Ausschalten der Zwischensatzerzeugung muss die Tangentialkopplung mittels TANG(...), aber ohne nachfolgendes Einschalten der Zwischensatzerzeugung mittels TLIFT(...), erneut definiert werden.

Syntax

TLIFT (<Folgeachse>)

Bedeutung

TLIFT (...):	Einschalten der Eckenerkennung mit Zwischensatzberechnung	
<Folgeachse>:	Achsenname der Folgeachse (Rundachse)	
	Datentyp:	AXIS
	Wertebereich:	Kanalachsenamen

Drehgeschwindigkeit der Folgeachse

Bahnachse

Wurde die Folgeachse vor der Aktivierung der Tangentialkopplung schon als Bahnachse verfahren, erfolgt die Drehbewegung im Zwischensatz als Bahnachse.

Durch die Vorgabe des Bezugsradius mit FGREF [<Achse>]=0.001, erfolgt die Drehbewegung mit der parametrisierten maximalen Achsgeschwindigkeit:

MD32000 \$MA_MAX_AX_VELO[<Folgeachse>]

Positionierachse

Wurde die Folgeachse vor der Aktivierung der Tangentialkopplung noch nicht als Bahnachse verfahren, erfolgt die Drehung im Zwischensatz als Positionierachse.

Die Drehbewegung erfolgt dabei mit der parametrisierten Positionierachsgeschwindigkeit:

MD32060 \$MA_POS_AX_VELO[<Folgeachse>]

13.2.3 Kopplung einschalten (TANGON)

Über die vordefinierte Prozedur TANGON(...) wird eine zuvor mit TANG(...) (Seite 625) definierte Tangentialkopplung eingeschaltet. Die Folgeachse wird dann beim nachfolgenden Verfahren der Leitachsen kontinuierlich zur Bahntangente ausgerichtet.

Winkel der Folgeachse

Der Winkel, den die Folgeachse in Bezug zur Bahntangente einnimmt, ist abhängig vom in TANG(...) vorgegebenen Übersetzungsverhältnis, dem im Maschinendatum MD37402 \$MA_TANG_OFFSET parametrisierten Offsetwinkel und dem dazu additiv wirkenden bei TANGON(...) vorgegebenen Offsetwinkel.

Optimierung "P"

Wurde bei der Definition der Tangentialkopplung (TANG(...)) als Optimierungsparameter der Wert "P" angegeben, muss beim Einschalten der Kopplung der Parameter "Überschleifweg" und optional der Parameter "Winkeltoleranz" angegeben werden.

Wird als Winkeltoleranz der Wert 0 vorgegeben, wirkt nur der Parameter "Überschleifweg".

Wird als Winkeltoleranz ein Wert größer 0 vorgegeben, ergibt sich der wirksame Überschleifweg aus dem Minimum des parametrisierten Überschleifwegs und des Überschleifwegs aufgrund der parametrisierten Winkeltoleranz.

Ist die Dynamik der Folgeachse nicht ausreichend, um den parametrisierten Bedingungen zu folgen, wird die Bahngeschwindigkeit der Leitachsen entsprechend reduziert.

Syntax

```
TANGON(<Folgeachse>, <Offsetwinkel>, <Überschleifweg>,
<Winkeltoleranz>)
```

Bedeutung

TANGON(...):	Einschalten einer Tangentialkopplung	
<Folgeachse>:	Achsenname der Folgeachse (Rundachse)	
	Datentyp:	AXIS
	Wertebereich:	Kanalachsnamen
<Offsetwinkel>:	Offsetwinkel der Folgeachse zur Bahntangente Bezugspunkt ist der Nullpunkt der Rundachse.	
	Datentyp:	REAL
<Überschleifweg>:	Maximal zulässiger Überschleifweg Würde der Überschleifweg aufgrund der Dynamikbedingungen größer, wird die Bahngeschwindigkeit der Leitachsen vermindert.	
	Datentyp:	REAL
<Winkeltoleranz>:	Maximal zulässige Toleranz bezüglich des vorgegebenen Winkels zwischen der Folgeachs-Nullstellung und der Bahntangente	
	Datentyp:	REAL

13.2.4 Kopplung ausschalten (TANGOF)

Über die vordefinierte Prozedur TANGOF(...) wird eine mit TANG(...) (Seite 625) definierte und mit TANGON(...) (Seite 629) eingeschaltete Tangentialkopplung ausgeschaltet. Die Folgeachse wird dann nicht mehr auf die Bahntangente der Leitachse ausgerichtet. Die Kopplung der Folgeachse an die Leitachsen bleibt aber auch nach dem Ausschalten weiterhin bestehen, was z.B. folgende Funktionen verhindert:

- Ebenenwechsel
- Geometrieachsumschaltung
- Definition einer neuen Tangentialkopplung für die Folgeachse

Das vollständige Aufheben der Bindung der Folgeachse an die Leitachsen erfolgt erst nach dem Löschen der Kopplung durch TANGDEL(...) (Seite 630).

Programmierung

TANGOF (<Folgeachse>)

Bedeutung

TANGOF (. . .):	Ausschalten einer Tangentialkopplung	
<Folgeachse>:	Achsenname der Folgeachse (Rundachse)	
	Datentyp:	AXIS
	Wertebereich:	Kanalachsnamen

13.2.5 Kopplung löschen (TANGDEL)

Eine mit TANG(...) (Seite 625) definierte Tangentialkopplung bleibt auch nach dem Ausschalten der Tangentialkopplung mit TANGOF(...) (Seite 630) hinaus bestehen. Die bestehende Tangentialkopplung verhindert dann z.B. weiterhin folgende Funktionen:

- Ebenenwechsel
- Geometrieachsumschaltung
- Definition einer neuen Tangentialkopplung für die Folgeachse

Mit der vordefinierten Prozedur TANGDEL(...) wird nach dem Ausschalten der Tangentialkopplung mit TANGOF(...) die bestehende Tangentialkopplung gelöscht.

Syntax

TANGDEL (<Folgeachse>)

Bedeutung

TANGDEL (. . .):	Löschen einer mit TANG() definierten Tangentialkopplung	
	Wirksamkeit:	Satzweise

<Folgeachse>:	Achsenname der Folgeachse, deren Tangentialkopplung gelöscht werden soll	
	Datentyp:	AXIS
	Wertebereich:	Kanalachsenamen

Beispiele

Leitachwechsel

Bevor für die Folgeachse eine neue Tangentialkopplung mit einer anderen Leitachse definiert werden kann, muss die bestehende Tangentialkopplung zuerst gelöscht werden.

Programmcode	Kommentar
N10 TANG (A, X, Y, 1)	; Tangentialkopplung für Folgeachse A definieren: A zu X und Y
N20 TANGON(A)	; Tangentialkopplung für Folgeachse A einschalten
N30 X10 Y20	
...	
N80 TANGOF(A)	; Tangentialkopplung für Folgeachse A ausschalten
N90 TANGDEL (A)	; Tangentialkopplung für Folgeachse A löschen
...	
N120 TANG (A, X, Z)	; Neue Tangentialkopplung für Folgeachse A definieren
N130 TANGON(A)	; Neue Tangentialkopplung für Folgeachse A einschalten
...	

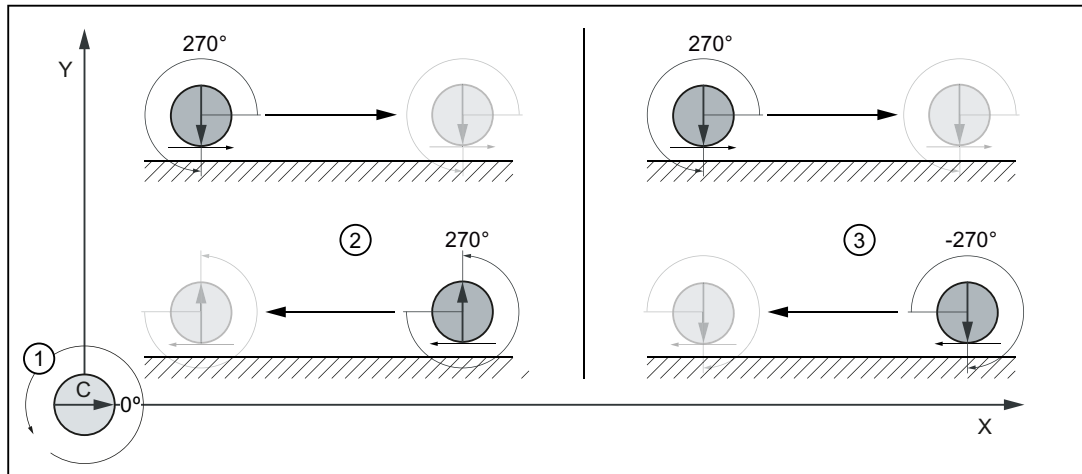
Geometrieachsumschaltung

Bevor eine Geometrieachsumschaltung für eine bestehende Kopplung vorgenommen werden kann, muss die Kopplung zuerst gelöscht werden.

Programmcode	Kommentar
N10 GEOAX(2, Y1)	; 2. Geometrieachse = Maschinenachse Y1
N20 TANG(A, X, Y)	; Tangentialkopplung für Folgeachse A definieren
N30 TANGON(A, 90)	; Tangentialkopplung für Folgeachse A einschalten
N40 G2 F8000 X0 Y0 I0 J50	; Bewegungssatz
N50 TANGOF(A)	; Tangentialkopplung für Folgeachse A ausschalten
N60 TANGDEL (A)	; Tangentialkopplung für Folgeachse A löschen
N70 GEOAX (2, Y2)	; 2. Geometrieachse = Maschinenachse Y2
N80 TANG(A, X, Y)	; Neue Tangentialkopplung für Folgeachse A definieren
N90 TANGON(A, 90)	; Neue Tangentialkopplung für Folgeachse A einschalten
...	

13.3 Grenzwinkel

Bei alternierender Verfahrbewegung springt die Richtung der Bahrtangente im Umkehrpunkt um 180° . Die Ausrichtung der Folgeachse springt dadurch ebenfalls um 180° . Dieses Verhalten ist in der Regel nicht sinnvoll (siehe Bild, ②). Stattdessen soll bei der Rückbewegung die Folgeachse mit der gleichen Ausrichtung wie bei der Hinbewegung verfahren werden (siehe Bild, ③).



- ① Grundstellung und positive Drehrichtung der Folge- / Rundachse C
- ② Hin- und Rückbewegung mit gleichem Offsetwinkel 270°
- ③ Hin- und Rückbewegung mit alternierendem Offsetwinkel 270° und -270°

Vermeidung des Umkehrsprungs

Über die Arbeitsfeldbegrenzung G25 (Untergrenze) und G26 (Obergrenze) muss für die Folgeachse ein minimaler und ein maximaler Wert für die Positionswerte bezogen auf das Basiskoordinatensystem vorgegeben werden. Danach muss die Arbeitsfeldbegrenzung der Folgeachse entsprechend der Bearbeitungssituation mit den Befehlen WALIMON / WALIMOF ein- / ausgeschaltet werden. Zum **Zeitpunkt der Bahnumkehr** muss die Arbeitsfeldbegrenzung **aktiv** sein.

Auswirkung

Bewegt sich beim Verfahren der Folgeachse mit dem aktuelle Offsetwinkel die Achsposition außerhalb der aktiven Arbeitsfeldbegrenzung, wird von der Steuerung geprüft, ob sich die Folgeachse bei Verwendung des negativen Offsetwinkels wieder im zulässigen Arbeitsbereich bewegen würde. Ist dies der Fall, wird die Folgeachse mit dem negativen Offsetwinkel verfahren.

Literatur

Programmierhandbuch Grundlagen; Kapitel "Ergänzende Befehle" > "Arbeitsfeldbegrenzung"
> "Arbeitsfeldbegrenzung im BKS (G25/G26, WALIMON, WALIMOF)"

13.4 Randbedingungen

Satzsuchlauf bei aktiver Tangentialkopplung

Es wird empfohlen, bei aktiver Tangentialsteuerung für einen Satzsuchlauf ausschließlich die unter TANG() (Seite 625) programmierbare Optimierungsart "P" und den Suchlauftyp 2 ("Satzsuchlauf mit Berechnung an Kontur") oder 5 ("Satzsuchlauf mit Berechnung im Modus Programmtest (SERUPRO)") zu verwenden.

Explizite Positionierung der Folgeachse

Wird die Folgeachse einer aktiven Tangentialkopplung explizit verfahren, wirkt der dabei verfahrenene Weg additiv zum programmierten Offsetwinkel.

13.5 Beispiele

Beispiel 1: Tangentialkopplung definieren und einschalten

Programmcode	Kommentar
; Tangentialkopplung definieren:	
; - Folgeachse C, Leitachsen X und Y	
; - Koppelfaktor: 1.0	
; - Koordinatensystem: BKS	
; - Optimierungsart: "P"	
N10 TANG(C,X,Y,1,"B","P")	; Vereinfacht: TANG(C,X,Y,,,"P")
; Tangentialkopplung einschalten:	
; - Offsetwinkel: 90°	
N20 TANGON(C,90)	
...	

Beispiel 2: Ebenenwechsel

Programmcode	Kommentar
; Tangentialkopplung definieren:	
; - Folgeachse A, Leitachsen X und Y	
; - Koppelfaktor: 1.0	
; - Koordinatensystem: BKS	
; - Optimierungsart: "S"	
N10 TANG(A,X,Y)	
; Tangentialkopplung einschalten:	
; - Offsetwinkel: 0°	
N20 TANGON(A)	
N30 X10 Y20	; Leitachsen verfahren
...	
N80 TANGOF(A)	; Tangentialkopplung ausschalten

13.5 Beispiele

Programmcode	Kommentar
N90 TANGDEL(A)	; Tangentialkopplung löschen
...	
; NEUE Tangentialkopplung definieren:	
; - Folgeachse A, Leitachsen X und Z	
; - Koppelfaktor: 1.0	
; - Koordinatensystem: BKS	
; - Optimierungsart: "S"	
TANG(A, X, Z)	
TANGON(A)	
...	

Beispiel 3: Geometrieachsumschaltung

Programmcode	Kommentar
N10 GEOAX(2, Y1)	; 2. Geometrieachse: Y1
; Tangentialkopplung definieren:	
; - Folgeachse A, Leitachsen X und Y	
; - Koppelfaktor: 1.0	
; - Koordinatensystem: BKS	
; - Optimierungsart: "S"	
N20 TANG(A, X, Y)	
; Tangentialkopplung einschalten:	
; - Offsetwinkel: 90°	
N30 TANGON(A, 90)	
N40 G2 F8000 X0 Y0 I0 J50	; Kreis verfahren
N50 TANGOF(A)	; Tangentialkopplung ausschalten
N60 TANGDEL(A)	; Tangentialkopplung löschen
N70 GEOAX(2, Y2)	; 2. Geometrieachse: Y2
; NEUE Tangentialkopplung definieren:	
; - Folgeachse A, Leitachsen X und Y	
; - Koppelfaktor: 1.0	
; - Koordinatensystem: BKS	
; - Optimierungsart: "S"	
N80 TANG(A, X, Y)	
; Tangentialkopplung einschalten:	
; - Offsetwinkel: 90°	
N90 TANGON(A, 90)	
...	

Beispiel 4: Optimierungsart "P"

Programmcode	Kommentar
...	

Programmcode	Kommentar
N100 G0 C0	
N110 G1 X1000 Y500 F50000	
N120 TRAORI	; Transformation einschalten
N130 G642	; Überschleifen einschalten
N171 TRANS X50 Y50	; Nullpunktverschiebung WKS
; Tangentialkopplung definieren:	
; - Folgeachse C, Leitachsen X und Y	
; - Koppelfaktor: 1.0	
; - Koordinatensystem: BKS	
; - Optimierungsart: "P"	
N180 TANG(C,X,Y,1,, "P")	
; Tangentialkopplung einschalten:	
; - Offsetwinkel: 0°	
; - Überschleifweg 5.0 mm	
; - Winkeltoleranz 2.0°	
N190 TANGON(C,0,5.0,2.0)	
N210 G1 X1310 Y500	
N215 G1 X1420 Y500	
N220 G3 X1500 Y580 I=AC(1420) J=AC(580)	
N230 G1 X1500 Y760	
N240 G3 X1360 Y900 I=AC(1360) J=AC(760)	
N250 G1 X1000 Y900	
N280 TANGOF(C)	; Tangentialkopplung ausschalten
N290 TRAFOOF	; Transformation ausschalten
...	

Beispiel 5: Ecke im Raum

Programmcode	; Kommentar
; Tangentialkopplung definieren:	
; - Folgeachse A, Leitachsen X und Y	
; - Koppelfaktor: 1.0	
; - Koordinatensystem: BKS	
; - Optimierungsart: "S" (Standard)	
TANG(A,X,Y,1.0,"B")	
TLIFT(A)	; Zwischensatzerzeugung einschalten
G1 G641 X0 Y0 Z0 A0	; Grundstellungen anfahren
; Tangentialkopplung einschalten:	
; - Offsetwinkel: 0°	
TANGON(A,0)	
N4 X10	; Leitachse X verfahren
N5 Z10	; Geometrieachse Z verfahren
N6 Y10	; Leitachse Y verfahren
M30	

Durch das Verfahren der Geometrieachsen in den Sätzen N4 und N6 entsteht ein nichtstetiger Verlauf d.h. eine Ecke in der Kontur der Leitachsen der Tangentialkopplung. Durch die Programmierung von TLIFT() wird vor N6 ein Zwischensatz eingefügt, in dem die Folgeachse A um 90° in die Tangentenrichtung der Y-Achse gedreht wird. Erst danach wird die Y-Achse in N6 verfahren.

13.6 Datenlisten

13.6.1 Maschinendaten

13.6.1.1 NC-spezifische Maschinendaten

Nummer	Bezeichner: \$MN_	Beschreibung
11450	SEARCH_RUN_MODE	Satzsuchlauf-Parametrierung

13.6.1.2 Achs-/Spindel-spezifische Maschinendaten

Nummer	Bezeichner: \$MA_	Beschreibung
37400	EPS_TLIFT_TANG_STEP	Tangentenwinkel für Eckenerkennung
37402	TANG_OFFSET	Voreinstellwinkel für tangentielle Nachführung

13.6.2 Settingdaten

13.6.2.1 Kanal-spezifische Settingdaten

Nummer	Bezeichner: \$SC_	Beschreibung
42121	AX_ADJUST_FEED	Bahnvorschub in Justierbewegungen der Tangentialachsen

13.6.3 Systemvariablen

Bezeichner	Beschreibung
\$AC_BLOCKTYPE	Aktueller Satz ist von TLIFT erzeugter Zwischensatz

T4: Automatisches Nachoptimieren mit AST (Option)

- nur 840D sl

14

14.1 Funktion

Hinweis

Die Funktion "Automatisches Nachoptimieren mit AST" ist eine lizenzpflichtige Option für SINUMERIK 840D sl.

Mit der Funktion "Automatisches Nachoptimieren mit AST" ist es möglich, Achsen bei veränderter Mechanik aus dem Teileprogramm heraus nachzuoptimieren.

Für die dazu erforderlichen Funktionen der automatischen Servo Optimierung (AST) stellt SIEMENS vordefinierte Zyklen zur Verfügung (siehe Kapitel "Programmierung (Seite 640)"). Diese kann der Maschinenhersteller verwenden, um eigene Optimierungszyklen zu erstellen. Der Maschinenbediener darf nur den Zyklus des Maschinenherstellers benutzen.

Hinweis

Ablageort der Optimierungszyklen

Damit der Maschinenbediener die vom Maschinenhersteller auf Basis der AST-Zyklen erstellten Optimierungszyklen aufrufen kann, müssen diese im Herstellerzyklen-Verzeichnis (`_N_CMA_DIR`) des NC-Speichers abgelegt werden.

Anwendung

Allgemeine Problemstellung: Abhängigkeit von der Beladung

Bei allen direkt angetriebenen Achsen müssen nach Veränderungen der Beladung und damit des Trägheitsmoments die Stromsollwertfilter angepasst werden, um die Stabilität des Drehzahlregelkreises sicherzustellen.

Solche Achsen sind:

- Tische mit Torque-Motoren
- Linearachsen mit Linearmotoren, die verschiedene Massen tragen
- Direkt angetriebene Hauptspindeln mit unterschiedlichen Spannvorrichtungen (Drehfutter)

14.1 Funktion

Alle konventionell angetriebenen Achsen (Servo-Motor mit Kugelrollspindel und evtl. mit Getriebe) erfordern meistens keine Nachoptimierung des Drehzahlreglers.

Hinweis

Wenn der aktuelle Parameterdatensatz nach einer Veränderung der Beladung nicht mehr passt, um die direkt angetriebene Achse sicher und ohne Instabilitäten zu verfahren, dann müssen die Parameterwerte **vor** dem Setzen der Reglerfreigabe (DB31, ... DBX2.1 = 1) angepasst werden. Das automatische Nachoptimieren mit AST ist erst nach der Reglerfreigabe durchführbar.

Möglich sind folgende Maßnahmen:

- Der aktive Parameterdatensatz ist so eingestellt, dass er für alle Beladungen geeignet ist und ohne Anpassung verwendet werden kann ("moderat" eingestellter Parameterdatensatz).
 - Ein "moderat" eingestellter Parameterdatensatz wird durch das automatische Nachoptimieren mit AST angepasst und anschließend aktiviert.
 - Die erforderlichen Parameterwerte werden per PLC (FB3, PI-Dienste, ...) und/oder Hersteller-Zyklus **vor** dem Setzen der Reglerfreigabe zugewiesen.
-

Typische Anwendungsfälle

Typische Anwendungsfälle siehe Kapitel "Beispiele (Seite 662)".

Einrichten der Optimierungszyklen / Fehlersuche

Es wird empfohlen, einen neu erstellten oder geänderten Optimierungszyklus vor Auslieferung der Maschine auszuführen und zu testen.

Dies hat folgende Gründe:

- Bei Programmierfehlern bzw. fehlenden Voraussetzungen (z. B. XML-Datei mit Optimierungsstrategie fehlt) wird das Programm abgebrochen. Anhand des ausgegebenen Alarms kann die Programmzeile mit dem Fehler erkannt und der Fehler anschließend behoben werden. Sobald der Optimierungszyklus fehlerfrei ausführbar ist, sollten auch beim Endkunden keine Programmstopps mehr auftreten.
- Wenn auf der Maschine vorher noch nie die bedienoberflächenbasierte automatische Servo Optimierung (AST) verwendet wurde und die Maschinendaten nicht nach der aktuellen Empfehlung für Vorsteuerung gesetzt sind (z. B. MD33000 \$MA_FIPO_TYPE = 3, MD32620 \$MA_FFW_TYPE = 4), dann wird der erste Durchlauf der AST-Zyklen dazu führen, dass diese Maschinendaten gesetzt werden und ein Power On (NC-Reset) oder ein Kanal-Reset notwendig wird.

Literatur:

Informationen zur bedienoberflächenbasierten automatischen Servo Optimierung siehe:

- Inbetriebnahmehandbuch CNC: NC, PLC, Antrieb

14.2 Inbetriebnahme

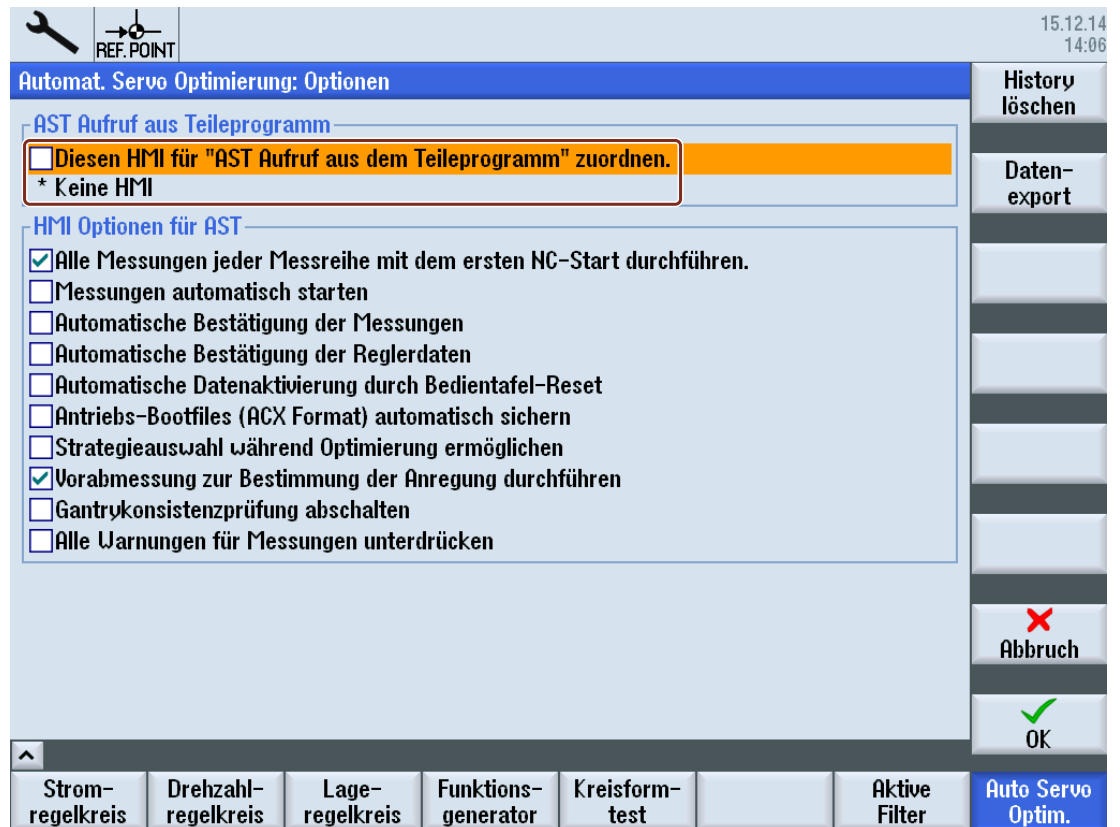
Systemvoraussetzungen

- SINUMERIK 840D sl mit SINUMERIK Operate
- CNC-Software ab Version 4.7 SP1
- Option "Automatisches Nachoptimieren mit AST" ist gesetzt.

Maschinen mit mehreren HMI-Komponenten

Die Funktion "Automatisches Nachoptimieren mit AST" kann immer nur einem HMI zugeordnet sein.

Um die aktuelle Zuordnung zu ändern, muss in dem für die Funktion vorgesehenen HMI das Fenster »Automat. Servo Optimierung: Optionen« aufgerufen und die Option »Diesen HMI für "AST Aufruf aus dem Teileprogramm" zuordnen« ausgewählt werden.



14.3 Programmierung

14.3.1 Übersicht

Vordefinierte Zyklen für das automatische Nachoptimieren mit AST

Folgende Zyklen stellt SIEMENS dem Maschinenhersteller für die Programmierung eigener Optimierungszyklen zur Verfügung:

- CYCLE751 - Optimierungssitzung öffnen/ausführen/schließen (Seite 641)
- CYCLE752 - Achse zu einer Optimierungssitzung hinzufügen (Seite 642)
- CYCLE753 - Optimierungsmodus auswählen (Seite 645)
- CYCLE754 - Datensatz hinzufügen/entfernen (Seite 646)
- CYCLE755 - Daten sichern/wiederherstellen (Seite 647)
- CYCLE756 - Optimierungsergebnisse aktivieren (Seite 648)
- CYCLE757 - Optimierungsdaten speichern (Seite 650)
- CYCLE758 - Parameterwert ändern (Seite 653)
- CYCLE759 - Parameterwert lesen (Seite 655)

Jeder dieser Zyklen ruft intern CYCLE750 auf, der das MMC-Kommando mit dem eigentlichen Funktionsaufruf enthält.

Datenhandling

Ablage in Standardverzeichnissen

Die automatische Servo Optimierung (AST) nutzt drei Standardverzeichnisse im Filesystem zur spezifischen Datenablage:

Verzeichnispfad	Zur Ablage folgender Dateien
<i>/user/sinumerik/nck/data/optimization</i>	<ul style="list-style-type: none"> • XML-Dateien mit den achs- bzw. bahnspezifischen Optimierungsdaten, die bei der bedienoberflächenbasierten automatischen Servo Optimierung automatisch erzeugt werden. • XML-Dateien mit den achs- bzw. bahnspezifischen Optimierungsdaten, die beim automatischen Nachoptimieren mit AST über CYCLE757 erzeugt werden.
<i>/user/sinumerik/nck/data/optimization/restore</i>	<ul style="list-style-type: none"> • Sicherungsdateien, die beim automatischen Nachoptimieren mit AST über CYCLE755 erzeugt werden.
<i>/user/sinumerik/nck/data/optimization/data</i>	<ul style="list-style-type: none"> • CSV-Dateien, die beim automatischen Nachoptimieren mit AST über CYCLE757 erzeugt werden.

Adressierung über relative Pfadangaben

Ausgangspunkt für die Adressierung einer Optimierungsdatei mit einer relativen Pfadangabe ist das datenspezifische Standardverzeichnis im Filesystem. Im Falle der Adressierung einer

XML-Datei mit achsspezifischen Optimierungsdaten würde die relative Pfadangabe z. B. vom Standardverzeichnis */user/sinumerik/nck/data/optimization* ausgehen.

Die relative Pfadangabe beginnt ohne Schrägstrich ("/").

Die Angabe von Unterverzeichnissen ist möglich.

Beispiel:

data/my_ast_ax1.xml

(entspricht der absoluten Pfadangabe */user/sinumerik/nck/data/optimization/data/my_ast_ax1.xml*)

Adressierung über absolute Pfadangaben

Über absolute Pfadangaben sind auch Optimierungsdateien außerhalb der Standardverzeichnisse adressierbar.

Bei der Adressierung einer Optimierungsdatei mit einer absoluten Pfadangabe wird die komplette Adresse der Datei angegeben.

Bei Linux besteht die komplette Adresse aus folgenden Elementen:

- Verzeichnispfad
- Dateiname

Beispiel:

- */user/sinumerik/hmi/log/optimization/data/my_ast_ax1.xml*

Adressierung von Dateien externer Speicher

Bei der Adressierung von Optimierungsdateien eines externen Speichers muss am Anfang des Adressierungspfades der Laufwerksname angegeben werden.

Es sind alle in *card/user/sinumerik/hmi/cfg/logdrive.ini* projektierten symbolischen Gerätenamen zulässig.

Beispiele:

- *LOCAL_DRIVE:/...* (lokales Laufwerk)
- *//DEV_5:/...* (benutzerdefiniertes Laufwerk)

14.3.2 CYCLE751 - Optimierungssitzung öffnen/ausführen/schließen

Syntax

CYCLE751 (<S_I_SESSIONCOMMAND>)

Parameter

Nr.	Parameter		
1	<S_I_SESSIONCOMMAND>		
	Datentyp:	INT	
	Wert:	0	nicht definiert
		1	Neue Optimierungssitzung öffnen Hinweis: Es können nicht mehrere Sitzungen gleichzeitig geöffnet sein. Durch das Öffnen einer neuen Sitzung wird die aktuelle Sitzung automatisch geschlossen.
		2	Aktuelle Optimierungssitzung schließen
3		Optimierungsfunktionen der aktuellen Sitzung ausführen mit sofortiger Übernahme der Optimierungsergebnisse Der Umfang der auszuführenden Optimierungsfunktionen (Vermessung des Drehzahlregelkreises, Vermessung des Lageregelkreises, Optimierung der Bahninterpolation) ist abhängig von den über die Bedienoberfläche einstellbaren achsspezifischen Optimierungsstrategien. Die Optimierungsergebnisse werden sofort übernommen (Aktualisierung der Achs- und Antriebssparameter, Aktualisierung der Optimierungsdateien).	
4	Optimierungsfunktionen der aktuellen Sitzung ausführen ohne sofortiger Übernahme der Optimierungsergebnisse Im Unterschied zu 3 werden die Optimierungsergebnisse nicht sofort übernommen. Die Übernahme kann zu einem späteren Zeitpunkt der Optimierungssitzung mit CYCLE756 erfolgen.		

14.3.3 CYCLE752 - Achse zu einer Optimierungssitzung hinzufügen

Nach dem Öffnen der Optimierungssitzung wird mit CYCLE752 angegeben, welche Achsen neu optimiert werden sollen. Ist eine Nachoptimierung der Bahninterpolation vorgesehen, werden mit CYCLE752 auch die Achsen angegeben, die zum Abgleich der Bahninterpolation notwendig sind.

Syntax

CYCLE752 (<S_I_AXIS>, <S_I_ACTIONREQUEST>, <S_B_ISPATHMEMBER>, <S_SZ_FILENAME>)

Parameter

Nr.	Parameter	
1	<S_I_AXIS>	
	Bedeutung:	Maschinenachsnummer Gibt die Achse an, die der Optimierungssitzung hinzugefügt werden soll. Hinweis: Bei Gantry- oder Master-Slave-Kopplungen ist nur die Leit- bzw. Masterachse des Kopplungsverbundes anzugeben.
	Datentyp:	INT

Nr.	Parameter		
2	<S_I_ACTIONREQUEST>		
	Bedeutung:	Gibt an, ob für die hinzuzufügende Achse gespeicherte Optimierungsdaten (Optimierungsstrategie und Optimierungsergebnisse) verwendet oder die Achse mit AST-Standard Einstellungen neu optimiert werden soll. Hinweis: Wenn die Achse vorher schon einmal optimiert und die Optimierungsdaten in einer Optimierungsdatei gespeichert wurden, wird man i. d. R. zunächst auf diese Daten zurückgreifen. Anschließend kann mit CYCLE753 das weitere Vorgehen festgelegt werden.	
	Datentyp:	INT	
	Wert:	0	nicht definiert
		1	Neuoptimierung mit AST-Standard Einstellungen Die Achse soll unter Verwendung von AST-Standard Einstellungen neu vermessen und nachoptimiert werden. Auf der Bedienoberfläche ausgewählte AST-Einstellungen werden nicht berücksichtigt.
2		Optimierungsdaten aus benutzerdefinierter Optimierungsdatei Für die Achse werden die gespeicherten Optimierungsdaten aus der angegebenen benutzerdefinierten Optimierungsdatei (siehe Parameter <S_SZ_FILENAME>) verwendet. Hinweis: Diese Einstellung ist nur möglich, wenn die Achse vorher schon einmal optimiert und die Ergebnisse in einer benutzerdefinierten Optimierungsdatei gespeichert wurden.	
3	Optimierungsdaten aus Standard-Optimierungsdatei Für die Achse werden die gespeicherten Optimierungsdaten aus der achsspezifischen Standard-Optimierungsdatei verwendet. Auf der Bedienoberfläche ausgewählte AST-Einstellungen werden berücksichtigt. Hinweis: Die Standard-Optimierungsdateien werden bei der bedienoberflächenbasierten automatischen Servo Optimierung zur Sicherung der Optimierungsinformationen (Optimierungsstrategie, Mess- und Optimierungsergebnisse) automatisch erzeugt und im Filesystem unter <i>/user/sinumerik/nck/data/optimization</i> abgelegt. Es gibt eine Standard-Optimierungsdatei für jede Achse und eine Standardoptimierungsdatei für die Bahninterpolation. Die Einstellung <S_I_ACTIONREQUEST> = 2 bzw. 3 wird z. B. in folgenden Fällen verwendet: <ul style="list-style-type: none"> Die Achse soll unter Verwendung von gespeicherten Strategie- und Messeinstellungen nachoptimiert werden. Dazu werden die gespeicherten Optimierungsdaten mit CYCLE752 geladen und anschließend mit CYCLE753 der anzuwendende Optimierungsmodus festgelegt. Im Anschluss an die Nachoptimierung einzelner Achsen muss die Bahninterpolation nachoptimiert werden. Gespeicherte Optimierungsergebnisse (Werte für NC- und Antriebsparameter) sollen wiederhergestellt werden. 		
3	<S_B_ISPATHMEMBER>		
	Bedeutung:	Berücksichtigung der Achse beim letzten Optimierungsschritt (Optimierung der Bahninterpolation)	
	Datentyp:	BOOL	
	Wert:	0 (FALSE)	Achse bei der Optimierung der Bahninterpolation nicht berücksichtigen
1 (TRUE)		Achse bei der Optimierung der Bahninterpolation berücksichtigen Alle Achsen, die mit CYCLE752(, , TRUE ,) hinzugefügt werden, werden bei der Optimierung der Bahninterpolation berücksichtigt. Dies geschieht im Rahmen der Optimierung, die z. B. mit CYCLE751(3) gestartet wird.	

Nr.	Parameter
4	<S_SZ_FILENAME>
	<p>Bedeutung: Adresse der benutzerdefinierten achsspezifischen Optimierungsdatei (nur relevant bei <S_I_ACTION-REQUEST> = 2!)</p> <p>Wenn die Datei über das programm-basierte automatische Nachoptimieren mit AST mittels CYCLE757 erzeugt wurde, dann befindet sich die Datei standardmäßig im Verzeichnis <i>/user/sinumerik/nck/data/optimization</i> und eine Angabe des Verzeichnispfades ist nicht erforderlich.</p> <p>Der häufigere Fall wird jedoch sein, dass die Datei über die bedienoberflächenbasierte automatische Servo Optimierung erzeugt und im Verzeichnis <i>/user/sinumerik/hmi/log/optimization</i> gespeichert wurde. Die Adresse muss dann neben dem Dateinamen auch den Verzeichnispfad enthalten (es sei denn, die Datei wurde nachträglich in das Standardverzeichnis <i>/user/sinumerik/nck/data/optimization</i> kopiert). Siehe hierzu auch "Beispiel 6: Vermessen einer Achse ohne Optimierung (Seite 679)".</p>
	<p>Datentyp: STRING [100]</p>

Hinweis

Gantry- / Master-Slave-Kopplung

Bei Gantry- oder Master-Slave-Kopplungen wird mit CYCLE752 nur die Leit- bzw. Master-Achse des Kopplungsverbundes angegeben. Die AST-Software fügt die gekoppelten Folge- bzw. Slave-Achsen automatisch hinzu.

AST wird dann automatisch alle Achsen eines Kopplungsverbundes vermessen und mit der gleichen Reglereinstellung optimieren. Dabei werden die Lagereglerverstärkungen gleich eingestellt, die Nachstellzeit im Drehzahlregler gleich gewählt und die effektive Drehzahlreglerverstärkung (relativ zur Gesamtträgheit) gleich gewählt. Die Stromsollwertfilter können unterschiedlich optimiert werden.

Gekoppelte Achsen müssen vor der automatischen Optimierung referenziert sein!

Hinweis

Vorabmessung

Die Entscheidung zur Durchführung einer Vorabmessung ist abhängig von den vorhandenen Optimierungsdaten der Achse.

- Fälle, in denen von der Steuerung eine Vorabmessung zur Bestimmung der Messparameter durchgeführt wird:
 - Die Optimierungsstrategie sieht eine Vorabmessung vor, Daten einer früheren Vorabmessung sind **nicht** vorhanden.
 - In den Optimierungsdaten ist **keine** Optimierungsstrategie festgelegt.
- Fälle, in denen keine Vorabmessung durchgeführt wird:
 - Die Optimierungsstrategie sieht eine Vorabmessung vor, Daten einer früheren Vorabmessung sind vorhanden.
 - Die Optimierungsstrategie sieht **keine** Vorabmessung vor (die Optimierungsstrategie hat dabei Vorrang vor der über die Bedienoberfläche anwählbaren Option "Vorabmessung zur Bestimmung der Anregung durchführen").

14.3.4 CYCLE753 - Optimierungsmodus auswählen

Die Programmierung von CYCLE753 ist nötig, wenn für die Achse andere Strategieeinstellungen wirken sollen, als im Standard hinterlegt bzw. über die Bedienoberfläche ausgewählt sind. Dies ist z. B. der Fall, wenn eine Achse mit benutzerdefinierten Strategie- und Messeinstellungen aus einer gespeicherten Optimierungsdatei neu vermessen und nachoptimiert werden soll.

Syntax

```
CYCLE753 (<S_I_AXIS>, <S_I_ACTION>, <S_B_SAVEBOOTFILES>)
```

Parameter

Nr.	Parameter		
1	<S_I_AXIS>		
	Bedeutung:	Maschinenachsnummer	
	Datentyp:	INT	
2	<S_I_ACTION>		
	Bedeutung:	Legt den Optimierungsmodus für die angegebene Achse fest.	
	Datentyp:	INT	
	Wert:	0	nicht definiert
		1	<p>Ableich der Bahninterpolation</p> <p>Die Achse (Bahnachse) soll nicht neu vermessen und nachoptimiert, sondern nur bei der Optimierung der Bahninterpolation berücksichtigt und angepasst werden.</p> <p>Für die Achse werden die gespeicherten Ergebnisse aus der Optimierungsdatei verwendet. Durch die Optimierung der Bahninterpolation können sich die achsspezifischen Maschinendaten und Antriebsparameter noch ändern.</p>
2		<p>Nachoptimierung mit anderer Optimierungsstrategie auf Basis gespeicherter Messungen</p> <p>Die Achse soll auf Grundlage der Ergebnisse einer früheren Messung und der über die Bedienoberfläche oder CYCLE758 geänderten aktuell gültigen Optimierungsstrategie nachoptimiert werden.</p> <p>Anwendung findet dieser Modus z. B. bei der Eliminierung bekannter periodischer Störfrequenzen (siehe "Beispiel 5: Nachoptimieren des Drehzahlregelkreises zur Eliminierung bekannter periodischer Störfrequenzen (Seite 677)").</p>	
3	<p>Messung und Nachoptimierung (→ Modus bei veränderter Mechanik/Beladung!)</p> <p>Die Mechanik/Beladung hat sich geändert. Die Achse soll unter Verwendung der benutzerdefinierten Strategie- und Messeinstellungen aus einer Optimierungsdatei neu vermessen und nachoptimiert werden. Die Strategie- und Messeinstellungen können über die Bedienoberfläche oder CYCLE758 noch angepasst werden.</p>		
3	<S_B_SAVEBOOTFILES>		
	Bedeutung:	Aktualisierte Antriebsparameterwerte als persistente Daten sichern	
	Datentyp:	BOOL	
	Wert:	0 (FALSE)	nein
1 (TRUE)		ja (Standardeinstellung)	

14.3.5 CYCLE754 - Datensatz hinzufügen/entfernen

Wenn die Optimierungsergebnisse statt des aktuellen Datensatzes einen bestimmten Achs- bzw. Antriebsdatensatz überschreiben sollen, muss der zu überschreibende Datensatz mit CYCLE754 zur Datensatzliste hinzugefügt werden. Bei dieser Liste handelt es sich um eine funktionsinterne, nicht sichtbare Liste zur Definition der zu optimierenden Datensätze. Standardmäßig ist darin nur der aktuell angewählte Datensatz enthalten. Soll ein Datensatz wieder aus der Liste entfernt werden, ist ebenfalls CYCLE754 zu verwenden.

Hinweis

Das Hinzufügen bzw. Entfernen eines Achs- bzw. Antriebsdatensatzes mit CYCLE754 bezieht sich nur auf die Datensatzliste, d. h. ein aus der Liste entfernter Achs- bzw. Antriebsdatensatz ist in der Steuerung bzw. im Antrieb nach wie vor existent.

Syntax

```
CYCLE754 (<S_I_AXIS>, <S_I_ACTIONREQUEST>, <S_I_DATASETTYPE>, <S_I_INDEX>)
```

Parameter

Nr.	Parameter	
1	<S_I_AXIS>	
	Bedeutung: Maschinenachsnummer	
	Datentyp: INT	
2	<S_I_ACTIONREQUEST>	
	Bedeutung: Legt fest, ob der angegebene Achs- bzw. Antriebsdatensatz zur Datensatzliste hinzugefügt oder aus der Liste entfernt werden soll.	
	Datentyp: INT	
	Wert:	0 nicht definiert
	1 Datensatz hinzufügen	
2 Datensatz entfernen		
3 Alle Datensätze aus der Liste entfernen		
Hinweis: Wenn die Liste leer ist, können die Optimierungsergebnisse nur in den aktiven Datensatz geschrieben werden.		
3	<S_I_DATASETTYPE>	
	Bedeutung: Datensatztyp	
	Datentyp: INT	
	Wert:	0 nicht definiert
	1 Achsdatensatz (Servo-Parametersatz)	
2 Antriebsdatensatz DDS		

Nr.	Parameter	
4	<S_I_INDEX>	
	Bedeutung:	Datensatzindex Gibt den Datensatz an, der hinzugefügt und in den die Optimierungsergebnisse geschrieben werden sollen oder der aus der Liste entfernt werden soll. Hinweis: Für den Achs datensatz-Index gilt: <n> = Parametersatznummer - 1 Beispiel: Index 0 $\hat{=}$ Parametersatz 1
	Datentyp:	INT
	Wertebereich:	-1, 0, 1, ... <MAX_INDEX> -1 aktiver Datensatz

14.3.6 CYCLE755 - Daten sichern/wiederherstellen

Mit CYCLE755 können Sicherungsdateien im Filesystem angelegt werden, die eine Sicherung der aktuellen Daten und damit eine spezifische Datenwiederherstellung zu jedem Zeitpunkt der Optimierungssitzung ermöglichen. Die Datenwiederherstellung wird ebenfalls mit CYCLE755 aufgerufen. Außerdem wird CYCLE755 dazu benutzt, das Vorhandensein einer Sicherungsdatei abzufragen sowie Sicherungsdateien zu löschen.

Hinweis

Nach erfolgreichem Abschluss der Optimierungssitzung werden die Sicherungsdateien nicht mehr benötigt. Sie sollten daher am Ende der Optimierungssitzung mit CYCLE755 wieder gelöscht werden.

Typische Beispiele für die Anwendung von CYCLE755 sind:

- Das Rückgängigmachen von Datenänderungen, die nur temporär gültig sein sollen.
- Das Wiederherstellen von Daten nach einer unterbrochenen Optimierungssitzung. Ein Programm, das das Vorhandensein einer Sicherungsdatei abfragt und bei einem positiven Ergebnis die Daten aus dieser Datei wiederherstellt, kann dazu benutzt werden, um nach Unterbrechung der Optimierungssitzung zu einem definierten Ausgangszustand zurückzukehren.

Syntax

```
CYCLE755 (<S_I_ACTIONREQUEST>, <S_SZ_NAME>, <S_SZ_GUD_BOOL_NAME>)
```

Parameter

Nr.	Parameter		
1	<S_I_ACTIONREQUEST>		
	Datentyp:	INT	
	Wert:	0	nicht definiert
		1	Daten für eine spätere Wiederherstellung sichern Mit dieser Einstellung werden die Datenwerte zum Zeitpunkt des Zyklenaufrufs in der angegebenen Datei gespeichert. Der Inhalt der Datensicherung wird von folgenden Parametern bestimmt: <ul style="list-style-type: none"> • der aktuell gültigen Optimierungsstrategie • der aktuell der Optimierungssitzung hinzugefügten Achsen • der aktuellen Liste der zu schreibenden Parametersätze
		2	Daten wiederherstellen Mit dieser Einstellung werden die Daten aus der angegebenen Datei wiederhergestellt. Voraussetzungen: <ul style="list-style-type: none"> • Die Optimierungssitzung ist noch aktiv. • Die Achs- und Datensatzauswahl zum Zeitpunkt der Sicherung ist noch vorhanden.
3		Sicherungsdatei löschen	
4	Vorhandensein einer Sicherungsdatei abfragen Das Ergebnis der Abfrage wird in einer Kanal-globalen Anwendervariablen (GUD) vom Typ BOOL abgelegt (siehe Parameter <S_SZ_GUD_BOOL_NAME>).		
2	<S_SZ_NAME>		
	Bedeutung:	Name der Sicherungsdatei Der Dateiname ist frei wählbar mit folgender Einschränkung: die Buchstabenfolge "AST" darf nicht am Anfang des Dateinamens stehen. Eine Pfadangabe (absolut oder relativ) ist nur erforderlich, wenn die Datei nicht im Standardverzeichnis /user/sinumerik/nck/data/optimization/restore abgelegt werden soll.	
	Datentyp:	STRING [100]	
3	<S_SZ_GUD_BOOL_NAME>		
	Bedeutung:	Name der GUD, in der das Ergebnis der Dateiabfrage abgelegt wird (nur relevant, wenn <S_I_ACTIONREQUEST> = 4!) Hinweis: Die angegebene GUD muss in einer Definitionsdatei als Kanal-globale Variable vom Typ BOOL definiert sein: DEF CHAN BOOL	
	Datentyp:	STRING [50]	

14.3.7 CYCLE756 - Optimierungsergebnisse aktivieren

Mit Hilfe von CYCLE756 kann die Aktivierung der Optimierungsergebnisse zu einem bestimmten Zeitpunkt / unter bestimmten Bedingungen erfolgen. Die Bedingungen müssen im Hersteller-Zyklus abgefragt werden. Außerdem kann mit CYCLE756 angegeben werden, welche Reglerparameter aktiv werden sollen: Originaldaten, achsoptimale Daten oder bahnoptimale Daten.

Syntax

```
CYCLE756(<S_I_AXIS>, <S_I_REGULATOR_ROLE>, <S_B_SAVEBOOTFILES>)
```

Parameter

Nr.	Parameter	
1	<S_I_AXIS>	
	Bedeutung: Maschinenachsnummer	
	Datentyp: INT	
2	<S_I_REGULATOR_ROLE>	
	Bedeutung: Gibt an, welche Optimierungsergebnisse für die angegebene Achse in der Steuerung und im Antrieb wirksam werden sollen. Hinweis: Ist die Achse Teil eines Gantry-Verbundes oder einer Master-Slave-Kopplung, dann sind in den Optimierungsergebnissen auch die Abstimmungen enthalten, die zur Dynamikanpassung aller Achsen des Kopplungsverbundes nötig waren.	
	Datentyp: INT	
	Wert:	0 nicht definiert
	1 Für die angegebene Achse sollen die ursprünglichen Daten wirksam werden.	
2 Für die angegebene Achse sollen die aktuellen Optimierungsergebnisse ohne Optimierung der Bahninterpolation wirksam werden (achsoptimale Daten).		
3 Für die angegebene Achse sollen die aktuellen Optimierungsergebnisse mit Optimierung der Bahninterpolation wirksam werden (bahnoptimale Daten).		
3	<S_B_SAVEBOOTFILES>	
	Bedeutung: Aktualisierte Antriebsparameterwerte als persistente Daten sichern	
	Datentyp: BOOL	
	Wert:	0 (FALSE) nein 1 (TRUE) ja (Standardeinstellung)

Anwendung

Die Anwendung von CYCLE756 ist z. B. in folgenden Fällen nützlich:

- Mit CYCLE751(4) werden die Optimierungsfunktionen der aktuellen Sitzung ausgeführt, ohne dass die Optimierungsergebnisse sofort übernommen werden. Das Aktivieren der Optimierungsergebnisse kann zu einem späteren Zeitpunkt der Optimierungssitzung mit CYCLE756 erfolgen. Vor dem Aktivieren können die Daten mit CYCLE758/759 gelesen und geprüft werden.
- Mit CYCLE752 werden gespeicherte Optimierungsdaten aus einer Optimierungsdatei geladen. Mit CYCLE756 können die geladenen Daten aktiviert werden.

14.3.8 CYCLE757 - Optimierungsdaten speichern

Im Gegensatz zur bedienoberflächenbasierten automatischen Servo Optimierung, bei der die Speicherung der Optimierungsdaten automatisch erfolgt (Standard-Optimierungsdateien), muss die Speicherung der Optimierungsdaten beim programm-basierten automatischen Nachoptimieren mit AST durch Programmierung von CYCLE757 explizit aufgerufen werden.

Folgende benutzerdefinierte Optimierungsdateien können mit CYCLE757 erzeugt werden:

- Optimierungsdateien im XML-Format:
 - für die achsspezifischen Optimierungsdaten
 - für die Daten zur Optimierung der Bahninterpolation
- Optimierungsdateien im CSV-Format mit Frequenz, Amplitude und Phase der folgenden Frequenzgänge:
 - gemessene Drehzahlregelstrecke
 - gemessener Mechanikfrequenzgang
 - berechneter geschlossener Drehzahlregler
 - berechneter geschlossener Lageregler
 - gemessener geschlossener Drehzahlregler
 - gemessener geschlossener Lageregler

Ohne Pfadangabe werden die Dateien im Filesystem in folgenden Standardverzeichnissen abgelegt:

- XML-Dateien: */user/sinumerik/nck/data/optimization*
- CSV-Dateien: */user/sinumerik/nck/data/optimization/data*

ACHTUNG

Datenverlust

Ohne Programmierung von CYCLE757 stehen Messdaten sowie Parameter zum Betrachten und zur Offline-Optimierung nach dem Schließen der Optimierungssitzung nicht mehr zur Verfügung. Die Optimierung wird aber (je nach Strategie) in die Achs- und Antriebsparameter geschrieben.

Abhilfe: Zu sichernde Optimierungsdaten vor dem Schließen der Optimierungssitzung mit CYCLE757 speichern.

Syntax

```
CYCLE757 (<S_I_AXIS>, <S_SZ_FILENAME>, <S_I_CONTENT_TYPE>,  
<S_I_FORMATFILTER>, <S_I_SUPPINFO>)
```

Parameter

Nr.	Parameter	
1	<S_I_AXIS>	
	Bedeutung:	Maschinenachsnummer
	Datentyp:	INT
2	<S_SZ_FILENAME>	
	Bedeutung:	<p>Name der zu erstellenden Optimierungsdatei (mit Dateierweiterung ".xml" oder ".csv")</p> <p>Der Dateiname ist frei wählbar mit folgender Einschränkung: die Buchstabenfolge "AST" darf nicht am Anfang des Dateinamens stehen.</p> <p>Eine Pfadangabe (absolut oder relativ) ist nur erforderlich, wenn die Datei nicht im Standardverzeichnis abgelegt werden soll.</p> <p>Beispiele:</p> <ul style="list-style-type: none"> • AXIS.xml • PATH.xml • SPEEDCTRL_PLANT.csv • POSCTRL_MECHRESP.csv <p>Spezialfall:</p> <ul style="list-style-type: none"> • Leerstring (""); nur gültig, wenn <S_I_CONTENT_TYPE> = 1 oder 2 <p>Die Angabe eines Leerstrings bewirkt, dass die Daten der programm-basierten automatischen Nachoptimierung in die Standard-Optimierungsdatei für die Achse (<S_I_CONTENT_TYPE> = 1) bzw. für die Bahninterpolation (<S_I_CONTENT_TYPE> = 2) geschrieben werden.</p> <p>Achtung:</p> <p>Die Standard-Optimierungsdateien enthalten die Daten der benutzeroberflächenbasierten automatischen Servo Optimierung. Durch Programmierung von <S_SZ_FILENAME> = "" werden diese von den Daten der programm-basierten automatischen Nachoptimierung überschrieben.</p>
Datentyp:	STRING [100]	

Nr.	Parameter															
3	<S_I_CONTENT_TYPE>															
	Bedeutung:	Legt fest, welche Optimierungsdaten in der zu erstellenden Optimierungsdatei gespeichert werden sollen.														
	Datentyp:	INT														
	Wert:	<table border="1"> <tr> <td data-bbox="344 412 411 451">0</td> <td data-bbox="411 412 1437 451">nicht definiert</td> </tr> <tr> <td data-bbox="344 451 411 846">1</td> <td data-bbox="411 451 1437 846"> Erzeugt eine XML-Datei, in der die achsspezifischen Optimierungsdaten für die angegebene Achse gespeichert werden. Folgende Daten sind enthalten: <ul style="list-style-type: none"> • Messbedingungen • Messergebnisse • Optimierungsstrategie • Optimierungsergebnisse Hinweis: Ist die Achse Teil eines Gantry-Verbundes oder einer Master-Slave-Kopplung, dann sind in den Optimierungsinformationen für die Achse auch die Optimierungsinformationen aller anderen Achsen des Kopplungsverbundes enthalten. </td> </tr> <tr> <td data-bbox="344 846 411 1029">2</td> <td data-bbox="411 846 1437 1029"> Erzeugt eine XML-Datei, in der die Daten zur Optimierung der Bahninterpolation gespeichert werden. Folgende Daten sind enthalten: <ul style="list-style-type: none"> • Liste aller beteiligten Achsen • Strategie für die Optimierung der Bahninterpolation </td> </tr> <tr> <td data-bbox="344 1029 411 1255">3</td> <td data-bbox="411 1029 1437 1255"> Erzeugt eine CSV-Datei, in der die Daten zum Frequenzgang der gemessenen Drehzahlregelstrecke gespeichert werden. Die Daten werden in Form einer 3-spaltigen Tabelle abgelegt: <ul style="list-style-type: none"> • Spalte 1: Frequenz [Hz] • Spalte 2: lineare Amplitude • Spalte 3: Phase [rad] </td> </tr> <tr> <td data-bbox="344 1255 411 1481">4</td> <td data-bbox="411 1255 1437 1481"> Erzeugt eine CSV-Datei, in der die Daten zum gemessenen Mechanikfrequenzgang gespeichert werden. Die Daten werden in Form einer 3-spaltigen Tabelle abgelegt: <ul style="list-style-type: none"> • Spalte 1: Frequenz [Hz] • Spalte 2: lineare Amplitude • Spalte 3: Phase [rad] </td> </tr> <tr> <td data-bbox="344 1481 411 1706">5</td> <td data-bbox="411 1481 1437 1706"> Erzeugt eine CSV-Datei, in der die Daten zum Frequenzgang des berechneten geschlossenen Drehzahlreglers gespeichert werden. Die Daten werden in Form einer 3-spaltigen Tabelle abgelegt: Spalte 1: Frequenz [Hz] Spalte 2: lineare Amplitude Spalte 3: Phase [rad] </td> </tr> <tr> <td data-bbox="344 1706 411 1915">6</td> <td data-bbox="411 1706 1437 1915"> Erzeugt eine CSV-Datei, in der die Daten zum Frequenzgang des berechneten geschlossenen Lagereglers gespeichert werden. Die Daten werden in Form einer 3-spaltigen Tabelle abgelegt: Spalte 1: Frequenz [Hz] Spalte 2: lineare Amplitude Spalte 3: Phase [rad] </td> </tr> </table>	0	nicht definiert	1	Erzeugt eine XML-Datei, in der die achsspezifischen Optimierungsdaten für die angegebene Achse gespeichert werden. Folgende Daten sind enthalten: <ul style="list-style-type: none"> • Messbedingungen • Messergebnisse • Optimierungsstrategie • Optimierungsergebnisse Hinweis: Ist die Achse Teil eines Gantry-Verbundes oder einer Master-Slave-Kopplung, dann sind in den Optimierungsinformationen für die Achse auch die Optimierungsinformationen aller anderen Achsen des Kopplungsverbundes enthalten.	2	Erzeugt eine XML-Datei, in der die Daten zur Optimierung der Bahninterpolation gespeichert werden. Folgende Daten sind enthalten: <ul style="list-style-type: none"> • Liste aller beteiligten Achsen • Strategie für die Optimierung der Bahninterpolation 	3	Erzeugt eine CSV-Datei, in der die Daten zum Frequenzgang der gemessenen Drehzahlregelstrecke gespeichert werden. Die Daten werden in Form einer 3-spaltigen Tabelle abgelegt: <ul style="list-style-type: none"> • Spalte 1: Frequenz [Hz] • Spalte 2: lineare Amplitude • Spalte 3: Phase [rad] 	4	Erzeugt eine CSV-Datei, in der die Daten zum gemessenen Mechanikfrequenzgang gespeichert werden. Die Daten werden in Form einer 3-spaltigen Tabelle abgelegt: <ul style="list-style-type: none"> • Spalte 1: Frequenz [Hz] • Spalte 2: lineare Amplitude • Spalte 3: Phase [rad] 	5	Erzeugt eine CSV-Datei, in der die Daten zum Frequenzgang des berechneten geschlossenen Drehzahlreglers gespeichert werden. Die Daten werden in Form einer 3-spaltigen Tabelle abgelegt: Spalte 1: Frequenz [Hz] Spalte 2: lineare Amplitude Spalte 3: Phase [rad]	6	Erzeugt eine CSV-Datei, in der die Daten zum Frequenzgang des berechneten geschlossenen Lagereglers gespeichert werden. Die Daten werden in Form einer 3-spaltigen Tabelle abgelegt: Spalte 1: Frequenz [Hz] Spalte 2: lineare Amplitude Spalte 3: Phase [rad]
0	nicht definiert															
1	Erzeugt eine XML-Datei, in der die achsspezifischen Optimierungsdaten für die angegebene Achse gespeichert werden. Folgende Daten sind enthalten: <ul style="list-style-type: none"> • Messbedingungen • Messergebnisse • Optimierungsstrategie • Optimierungsergebnisse Hinweis: Ist die Achse Teil eines Gantry-Verbundes oder einer Master-Slave-Kopplung, dann sind in den Optimierungsinformationen für die Achse auch die Optimierungsinformationen aller anderen Achsen des Kopplungsverbundes enthalten.															
2	Erzeugt eine XML-Datei, in der die Daten zur Optimierung der Bahninterpolation gespeichert werden. Folgende Daten sind enthalten: <ul style="list-style-type: none"> • Liste aller beteiligten Achsen • Strategie für die Optimierung der Bahninterpolation 															
3	Erzeugt eine CSV-Datei, in der die Daten zum Frequenzgang der gemessenen Drehzahlregelstrecke gespeichert werden. Die Daten werden in Form einer 3-spaltigen Tabelle abgelegt: <ul style="list-style-type: none"> • Spalte 1: Frequenz [Hz] • Spalte 2: lineare Amplitude • Spalte 3: Phase [rad] 															
4	Erzeugt eine CSV-Datei, in der die Daten zum gemessenen Mechanikfrequenzgang gespeichert werden. Die Daten werden in Form einer 3-spaltigen Tabelle abgelegt: <ul style="list-style-type: none"> • Spalte 1: Frequenz [Hz] • Spalte 2: lineare Amplitude • Spalte 3: Phase [rad] 															
5	Erzeugt eine CSV-Datei, in der die Daten zum Frequenzgang des berechneten geschlossenen Drehzahlreglers gespeichert werden. Die Daten werden in Form einer 3-spaltigen Tabelle abgelegt: Spalte 1: Frequenz [Hz] Spalte 2: lineare Amplitude Spalte 3: Phase [rad]															
6	Erzeugt eine CSV-Datei, in der die Daten zum Frequenzgang des berechneten geschlossenen Lagereglers gespeichert werden. Die Daten werden in Form einer 3-spaltigen Tabelle abgelegt: Spalte 1: Frequenz [Hz] Spalte 2: lineare Amplitude Spalte 3: Phase [rad]															

Nr.	Parameter	
	7	Erzeugt eine CSV-Datei, in der die Daten zum Frequenzgang des gemessenen geschlossenen Drehzahlreglers gespeichert werden. Die Daten werden in Form einer 3-spaltigen Tabelle abgelegt: Spalte 1: Frequenz [Hz] Spalte 2: lineare Amplitude Spalte 3: Phase [rad]
	8	Erzeugt eine CSV-Datei, in der die Daten zum Frequenzgang des gemessenen geschlossenen Lagereglers gespeichert werden. Die Daten werden in Form einer 3-spaltigen Tabelle abgelegt: Spalte 1: Frequenz [Hz] Spalte 2: lineare Amplitude Spalte 3: Phase [rad]
4	<S_I_FORMATFILTER>	
	Bedeutung:	reserviert
	Datentyp:	INT
5	<S_I_SUPPINFO>	
	Bedeutung:	reserviert
	Datentyp:	INT

14.3.9 CYCLE758 - Parameterwert ändern

Mit CYCLE758 können vor der Optimierung einzelne Strategie und Messeinstellungen bzw. nach der Optimierung die Optimierungsergebnisse verändert werden.

Syntax

CYCLE758 (<S_I_AXIS>, <S_I_PARAMID>, <S_I_MEASTYPE>, <S_I_MEASINDEX>, <S_SZ_NEWVALUE>)

Parameter

Nr.	Parameter	
1	<S_I_AXIS>	
	Bedeutung:	Maschinenachsnnummer (nicht relevant bei Parametern, die die Optimierung der Bahninterpolation betreffen!)
	Datentyp:	INT
2	<S_I_PARAMID>	
	Bedeutung:	ID-Nummer des Parameters, dessen Wert geändert werden soll Parameter-IDs siehe "Liste der Parameter zur automatischen Servo Optimierung (Seite 656)".
	Datentyp:	INT

Nr.	Parameter		
3	<S_I_MEASTYPE>		
	Bedeutung:	Messtyp, auf den sich der Parameter bezieht (nur relevant bei Messparametern!) Dient zur eindeutigen Identifizierung von Messparametern mit gleicher ID-Nummer.	
	Datentyp:	INT	
	Wert:	1	Mechanikfrequenzgang
		2	Geschlossener Lageregelkreis
3		Geschlossener Drehzahlregelkreis	
4		Drehzahlreglerstrecke	
5	Drehzahlsollwertsprung		
4	<S_I_MEASINDEX>		
	Bedeutung:	Index der Messung, auf die sich der Parameter bezieht (nur relevant bei Messparametern!) Dient zur eindeutigen Identifizierung von Messparametern mit gleicher ID-Nummer, die sich auf den gleichen Messtyp beziehen. Die Angabe ist nur nötig, wenn mehrere Messungen eines Messtyps existieren.	
	Datentyp:	INT	
5	<S_SZ_NEWVALUE>		
	Bedeutung:	Neuer Parameterwert	
	Datentyp:	STRING [20]	
	Wert:	Der Wert ist als Zeichenkette mit Anführungszeichen anzugeben. Die Zeichenkette darf aus maximal 20 Zeichen bestehen. Zulässig sind nur Werte, die dem Datentyp des zu ändernden Parameters entsprechen und innerhalb des gültigen Wertebereichs liegen. Beispiele: 1. Der zu ändernde Parameter ist vom Datentyp BOOL (z. B. <S_I_PARAMID> = 157). Wertebereich: 0 (= FALSE), 1 (= TRUE) Daraus ergeben sich für <S_SZ_NEWVALUE> folgende zulässige Zeichenketten: "true", "false", "TRUE", "FALSE", "0" oder "1" Unzulässige Zeichenketten wären z. B.: 0, 1, true, false, "2", "my_bool" 2. Der zu ändernde Parameter ist vom Datentyp REAL (z. B. <S_I_PARAMID> = 34). In diesem Fall kann mit <S_SZ_NEWVALUE> theoretisch jede Realzahl innerhalb des Wertebereichs $\pm (\sim 2,2 \cdot 10^{-308} \dots \sim 1,8 \cdot 10^{+308})$ angegeben werden. Zulässige Zeichenketten sind z. B.: "14.1", ".0023" Unzulässige Zeichenketten wären z. B.: "true", 14.1, "my_real", "R1"	

Hinweis

Gantry- / Master-Slave-Kopplung

Soll der Messparameter einer bestimmten Folge- oder Slave-Achse geändert werden, muss im CYCLE758 die entsprechende Nummer dieser Achse angegeben werden (**nicht** die Nummer der Leit- oder Master-Achse wie beim Hinzufügen gekoppelter Achsen mit CYCLE752 (Seite 642)).

14.3.10 CYCLE759 - Parameterwert lesen

Mit CYCLE759 können Parameter zur Strategie und Messung und optimierte Reglerparameter gelesen werden. Dies ist hilfreich, um vor der Optimierung die mit CYCLE752 geladene Datei zu prüfen oder nach der Optimierung die Ergebnisse zu lesen (z. B. ermittelte Gesamtträgheit, ermittelte Verstärkung).

Syntax

```
CYCLE759(<S_I_AXIS>, <S_I_PARAMID>, <S_I_MEASTYPE>, <S_I_MEASINDEX>,
<S_SZ_GUDRESULT>)
```

Parameter

Nr.	Parameter		
1	<S_I_AXIS>		
	Bedeutung:	Maschinenachsnnummer (nicht relevant bei Parametern, die die Optimierung der Bahninterpolation betreffen!)	
	Datentyp:	INT	
2	<S_I_PARAMID>		
	Bedeutung:	ID-Nummer des Parameters, dessen Wert gelesen werden soll Parameter-IDs siehe "Liste der Parameter zur automatischen Servo Optimierung (Seite 656)".	
	Datentyp:	INT	
3	<S_I_MEASTYPE>		
	Bedeutung:	Messtyp, auf den sich der Parameter bezieht (nur relevant bei Messparametern!) Dient zur eindeutigen Identifizierung von Messparametern mit gleicher ID-Nummer.	
	Datentyp:	INT	
	Wert:	1	Mechanikfrequenzgang
		2	Geschlossener Lageregelkreis
		3	Geschlossener Drehzahlregelkreis
4		Drehzahlreglerstrecke	
5		Drehzahlsollwertsprung	
4	<S_I_MEASINDEX>		
	Bedeutung:	Index der Messung, auf die sich der Parameter bezieht (nur relevant bei Messparametern!) Dient zur eindeutigen Identifizierung von Messparametern mit gleicher ID-Nummer, die sich auf den gleichen Messtyp beziehen. Die Angabe ist nur nötig, wenn mehrere Messungen eines Messtyps existieren.	
	Datentyp:	INT	
5	<S_SZ_GUDRESULT>		
	Bedeutung:	Name der globalen Anwendervariablen (GUD), in die der Wert des Parameters kopiert werden soll. Hinweis: Die angegebene GUD muss in einer Definitionsdatei definiert sein. Der Datentyp der GUD muss mit dem Datentyp des Parameters übereinstimmen.	
	Datentyp:	STRING [20]	

14.3.11 Liste der Parameter zur automatischen Servo Optimierung

Die ID eines Parameters zur automatischen Servo Optimierung, dessen Wert mit CYCLE758 geändert bzw. mit CYCLE759 gelesen werden soll, kann aus der folgenden Tabelle entnommen werden:

ID	Bedeutung	Einheit	Datentyp	Wertebereich	Schreiben mit CYCLE758	Lesen mit CYCLE759
Messbedingungen						
001	Anfangsposition der Achse während einer Messung	mm oder deg	REAL		-	+
002	Richtungssequenz während einer Messung <ul style="list-style-type: none"> • 1 = plus • 2 = minus • 3 = plus, dann minus • 4 = minus, dann plus 		INT	1, 2, 3, 4	+	+
003	Anzahl Messwiederholungen für die Frequenzbereichsmittelung		INT	0 ...	+	+
004	Anzahl Zeitbereichsfenster in jedem Zeitbereichsmittel		INT	1 ...	+	+
005	Einschwingzeit: Zeit, die nach der PRBS-Anregung abzuwarten ist, bevor das erste Messfenster erfasst wird	s	REAL	0 ...	+	+
006	Bandbreite der Messung Dieser Wert bestimmt nur die Bandbreite der Messung (also Anregung und Abtastfrequenz). Dadurch kann die Frequenzauflösung beeinflusst werden.	Hz	REAL	Float: 0 ... fs/2	+	+
007	Zeit für die Durchführung einer einzelnen Messung (ohne Wiederholung, Hochlauf- und Einschwingzeit)	s	REAL		-	+
008	Erforderliche Zeit, um den Geschwindigkeits-Offset auf den Sollwert zu bringen	s	REAL		-	+
009	Geschätzte zurückgelegte Wegstrecke während der maximalen Verfahrbewegung in der Wiederholungssequenz	m oder rad	REAL		-	+
010	Geschätzte Gesamtzeit der Achsbewegung während der Messung	s	REAL		-	+
011	Gibt den Messtyp an: <ul style="list-style-type: none"> • 1 = Mechanikfrequenzgang • 2 = Geschlossener Lageregelkreis • 3 = Geschlossener Drehzahlregelkreis • 4 = Drehzahlreglerstrecke • 5 = Drehzahlsollwertsprung 		INT		-	+
012	Gibt an, ob die Achsbewegung innerhalb der Verfahrbereichsgrenzen liegen wird		BOOL	0 (= FALSE) 1 (= TRUE)	-	+

ID	Bedeutung	Einheit	Datentyp	Wertebereich	Schreiben mit CYCLE758	Lesen mit CYCLE759
013	Gibt an, ob eine Messung mit maximaler Bandbreite durchgeführt wird		BOOL	0 (= FALSE) 1 (= TRUE)	-	+
021	Amplitude der Anregung	Nm, N, m/s, rad/s, m, rad (abhängig vom Messtyp ¹⁾)	REAL	dynamisch	+	+
022	Geschwindigkeits-Offset	m/s oder rad/s (abhängig vom Messtyp ¹⁾)	REAL	dynamisch	+	+
Drehzahlregelung (Regler)						
031	Proportionalverstärkung für den Drehzahlregelkreis	Nms/rad oder Ns/m	REAL		+	+
032	Nachstellzeit für den Drehzahlregelkreis	s	REAL		+	+
033	Trägheitsmoment des Motors	kgm ² oder kg	REAL		-	+
034	Gesamt-Trägheitsmoment der Achse, ausgedrückt in Einheiten des Motors	kgm ² oder kg	REAL		+	+
035	Gibt an, ob das Referenzmodell für die Drehzahlregelung aktiviert ist		BOOL	0 (= FALSE) 1 (= TRUE)	+	+
036	Frequenz des Referenzmodells PT2	Hz	REAL		+	+
037	Dämpfung des Referenzmodells PT2		REAL		+	+
038	Verhältnis der Proportionalverstärkung Kp zum Trägheitsmoment, das die 1/s-Bandbreite des idealisierten PT1-äquivalenten (proportional geregelten) Systems darstellt	1/s	REAL		-	+
Lageregelung (Regler)						
101	Lageregelkreisverstärkung	1000/min	REAL		+	+
102	Gibt an, ob DSC (Dynamische Steifigkeitsregelung) aktiv ist		BOOL	0 (= FALSE) 1 (= TRUE)	+	+
103	Gibt die Art der Vorsteuerung an: <ul style="list-style-type: none"> 0 = Keine 1 = Drehzahlvorsteuerung 2 = Drehzahl- und Momentenvorsteuerung 		INT	0, 1, 2	+	+
104	Gesamt-Trägheitsmoment der Achse als Parameter für die Momentenvorsteuerung	kg m ² oder kg	REAL		+	+
105	Verzögerung der Drehzahlvorsteuerung im Antrieb (z. B. p1429)	s	REAL		+	+
106	Ersatzzeitkonstante des Drehzahlregelkreises als Parameter für die Drehzahlvorsteuerung	s	REAL		+	+

14.3 Programmierung

ID	Bedeutung	Einheit	Datentyp	Wertebereich	Schreiben mit CYCLE758	Lesen mit CYCLE759
107	Angepasster Parameterwert 106 zur Berücksichtigung folgender Einflüsse: <ul style="list-style-type: none"> • Inkonsistente Verzögerungen • Änderungen durch den NC 	s	REAL		+	+
108	Ersatzzeitkonstante des Stromregelkreises als Parameter für die Momentenvorsteuerung	s	REAL		+	+
109	Angepasster Parameterwert 108 zur Berücksichtigung folgender Einflüsse: <ul style="list-style-type: none"> • Inkonsistente Verzögerungen • Änderungen durch den NC 	s	REAL		+	+
110	Zur Dynamikanpassung verwendete PT1-Zeitkonstante der Achse (MD32910 \$MA_DYN_MATCH_TIME)	s	REAL		+	+
111	Zur Dynamikanpassung verwendete Zeitkonstante für das achsspezifische Sollwert-Phasenfilter (MD32895 \$MA_DESVAL_DELAY_TIME)	s	REAL		+	+
112	Berechnete Gesamtzeitkonstante für das Positionierverhalten inkl. Vorsteuerung und allen Vorfiltern Diese Zeitkonstante dient nur als Information für den Benutzer. Man kann Sie als Zielzeitkonstante für die Lageregler-Strategie einer anderen Achse benutzen. Durch das Erzwingen der gleichen Gesamtzeitkonstante passt die Interpolation schon nach der Achs-Optimierung.	s	REAL		-	+
Lageregelung (Strategie)						
151	Anpassungsfaktor (Reduzierung) für den optimierten maximalen Kv-Wert		REAL	0,1 ... 1,0	+	+
152	nicht definiert					
153	Gibt die zusätzliche Methode an, die zur Maximierung des Kv-Werts verwendet wird: <ul style="list-style-type: none"> • 0 = keine (Standard) • 1 = PT1-Drehzahlfilter 2 • 2 = Dämpfung optimaler Drehzahlregler • 3 = PT1-Drehzahlfilter 1 		INT	0 ... 3	+	+
154	DSC		BOOL	0 (= FALSE) 1 (= TRUE)	+	+
155	Vorsteuerungsart wählen: <ul style="list-style-type: none"> • 0 = Keine • 1 = Drehzahlvorsteuerung • 2 = Drehzahl- und Momentenvorsteuerung 		INT	0, 1, 2	+	+
156	Maximal zulässiger Kv-Wert	1000/min	REAL	0,1 ... 99,0	+	+
157	Legt fest, ob für die Achse die Vorsteuerung immer aktiv sein soll		BOOL	0 (= FALSE) 1 (= TRUE)	+	+

ID	Bedeutung	Einheit	Datentyp	Wertebereich	Schreiben mit CYCLE758	Lesen mit CYCLE759
158	Sollwert für die Äquivalenzzeit der Gesamtreaktion des Lagereglers (einschließlich Vorsteuerung und Sollwertfilter)	s	REAL		+	+
159	Zeigt an, ob versucht werden soll, die Solläquivalenzzeit zu erreichen		BOOL	0 (= FALSE) 1 (= TRUE)	+	+
160	Gibt die Auswahl zur Ruckfiltertechnologie an: <ul style="list-style-type: none"> 0 = Bestehende Ruckfiltereinstellungen in den Maschinendaten beibehalten 1 = Ruckfilter ausschalten 2 = Ruckfilter Typ 2 (= Gleitende Mittelwertbildung) verwenden 		INT	0 ... 2	+	+
161	Zeitkonstante (MD32410 \$MA_AX_JERK_TIME) für Ruckfilter Typ 2 (Parameter 160 = 2)	s	REAL	0 ... 0,5	+	+
Drehzahlregelung (Strategie)						
201	Sollwert für die Amplitudenreserve, der bei der Optimierung verwendet wird	dB	REAL		+	+
202	Sollwert für die Phasenreserve, der bei der Optimierung verwendet wird	deg	REAL		+	+
203	Zeigt an, ob das Referenzmodell aktiviert werden soll		BOOL	0 (= FALSE) 1 (= TRUE)	+	+
204	Legt den zulässigen Minimalwert für die Nachstellzeit des Drehzahlregelkreises fest	s	REAL	0,001 ... 1,000	+	+
205	Legt fest, ob beim Entwurf von Bandsperrfiltern eine konstante Absenkung verwendet werden soll (für den Fall, dass deutlich mehr Pole als Filter vorhanden sind)		BOOL	0 (= FALSE) 1 (= TRUE)	+	+
206	Optimierungsaggressivität des Drehzahlreglers Ein höherer Wert führt zu einem besseren Optimierungsergebnis bei gleichzeitiger Abnahme der Robustheit		REAL	0 ... 1	+	+

14.3 Programmierung

ID	Bedeutung	Einheit	Datentyp	Wertebereich	Schreiben mit CYCLE758	Lesen mit CYCLE759
207	<p>Kp-Obergrenze basierend auf der Bandbreite eines PT1-äquivalenten proportional geregelten Systems</p> <p>Die Angabe einer maximalen Bandbreite des Drehzahlreglers ist eine weitere Regel zur Begrenzung des Kp. Dabei wünscht man sich eine Begrenzung der Dynamik des Drehzahlreglers in Form einer "Regelbandbreite" (Führungsverhalten des Drehzahlreglers im Frequenzbereich). Amplituden- und Phasenreserve müssen nach wie vor eingehalten werden. Sofern also die Amplituden- und Phasenreserve einen höheren Kp zulassen, wird Kp unter folgenden Annahmen berechnet:</p> <ul style="list-style-type: none"> • Die gesamte Trägheit ist eine starre Masse. • Der geschlossene Drehzahlregelkreis kann als Tiefpass erster Ordnung (PT1) angenähert werden. 	Hz	REAL		+	+
208	Zeigt an, ob ein Verstärkungsfilter zur Eliminierung frequenzspezifischer periodischer Störungen entworfen und angewendet werden soll		BOOL	0 (= FALSE) 1 (= TRUE)	+	+
209	Frequenz des optionalen Verstärkungsfilters zur Eliminierung frequenzspezifischer periodischer Störungen	Hz	REAL		+	+
Optimierung der Bahninterpolation (Strategie)						
251	Zeigt an, ob versucht werden soll, eine Übereinstimmung mit dem effektiven Kp-Wert zu erzielen		BOOL	0 (= FALSE) 1 (= TRUE)	+	+
252	Zeigt an, ob versucht werden soll, eine Übereinstimmung mit der Nachstellzeit Tn zu erzielen		BOOL	0 (= FALSE) 1 (= TRUE)	+	+
253	Anteil, um den der Kp-Wert maximal reduziert werden darf, um eine Übereinstimmung mit dem effektiven Kp-Wert zu erzielen		REAL	0,1 ... 1,0	+	+
254	Legt fest, ob Nichtübereinstimmungen des Kv-Werts zwischen Achsen zulässig sind		BOOL	0 (= FALSE) 1 (= TRUE)	+	+
255	Zeigt an, ob die Achsen ohne Vorsteuerung oder mit Drehzahlvorsteuerung abgegliche Referenzmodelle haben müssen		BOOL	0 (= FALSE) 1 (= TRUE)	+	+
256	<p>Gibt die Strategie für den Bahnabgleich an:</p> <ul style="list-style-type: none"> • 0 = Ersatzzeit des Regelkreises anpassen über Reglerparameter und/oder MD32800/MD32810 • 1 = MD32895 (Zeitkonstante für das axiale Sollwert-Phasenfilter) zur Anpassung verwenden 		INT	0, 1	+	+

ID	Bedeutung	Einheit	Datentyp	Wertebereich	Schreiben mit CYCLE758	Lesen mit CYCLE759
257	Legt fest, ob für alle Achsen in der Gruppe die Vorsteuerung immer aktiv sein soll		BOOL	0 (= FALSE) 1 (= TRUE)	+	+
258	Legt fest, ob die Äquivalenzzeiten für alle Achsen gleich sein sollen		BOOL	0 (= FALSE) 1 (= TRUE)	+	+

¹⁾ Die Einheit des Parameterwerts ist vom Messtyp abhängig:

- Folgende Messungen werden im Antrieb ausgeführt. Aus diesem Grund beziehen sich die Einheiten auf die Motorseite. Die resultierende Achsgeschwindigkeit ergibt sich durch Berücksichtigung der Getriebefaktoren oder Spindelsteigung. Nur bei Linearmotoren und Torque-Motoren (ohne Getriebe) ist die hier angegebene Geschwindigkeit gleich der Achsgeschwindigkeit. Hierbei ist zu beachten, dass in den AST-Bildschirmmasken die Achsgeschwindigkeit in z. B. mm/min vorgegeben wird. Die Zahlenwerte aus den Masken können hier nicht übernommen werden.

Messtyp	Einheit			
	Parameter 021: Amplitude der Anregung		Parameter 022: Geschwindigkeits-Offset	
	Drehmotor	Linearmotor	Drehmotor	Linearmotor
Drehzahlreglerstrecke	Nm	N	rad/s	m/s
Geschlossener Drehzahlregelkreis	rad/s	m/s	rad/s	m/s

- Folgende Messungen werden über die Steuerung ausgeführt. Hier beziehen sich die Einheiten auf die Achse, also auf die Lastseite. Es ist ggf. eine Umrechnung auf SI-Einheiten notwendig.

Messtyp	Einheit			
	Parameter 021: Amplitude der Anregung		Parameter 022: Geschwindigkeits-Offset	
	Rundachse	Linearachse	Rundachse	Linearachse
Mechanikfrequenzgang	rad/s	m/s	rad/s	m/s
Geschlossener Lageregelkreis	rad	m	rad/s	m/s

14.4 Diagnose

Tritt bei der Zyklenausführung ein Fehler auf, wird ein entsprechender HMI-Alarm (Nummernbereich 150001 bis 150004) sowie NC-Alarm 61840 ausgegeben. Anhand der angegebenen Satznummer im Alarm 61840 kann der Anwender feststellen, welcher NC-Satz mit einem CYCLE75x-Aufruf den Fehler verursacht hat.

Literatur:

Alarmbeschreibungen siehe Diagnosehandbuch Alarmer.

14.5 Randbedingungen

"Automatisches Nachoptimieren mit AST" bei aktiver Kopplung

Die folgende Tabelle zeigt, welche Kopplungsfunktionen das "automatische Nachoptimieren mit AST" unterstützen:

Kopplungsfunktion	"Automatisches Nachoptimieren mit AST" anwendbar?
Gantry-Achsen	+
Master-Slave-Kopplung	+
Mitschleppen	-
Leitwertkopplung	-
Elektronisches Getriebe	-
Synchronspindel	-

14.6 Beispiele

14.6.1 Beispiel 1: Vermessen einer Achse mit AST-Standard Einstellungen

Die Rundachse (hier: 4. Achse), die mit einem Torque-Motor angetrieben wird, ist stark von der Trägheit der aufgespannten Masse abhängig. Ändert sich die Lastträgheit, muss die Achse neu vermessen werden. Es sollen keine Reglerwerte von AST überschrieben werden, die Optimierung wird aber dennoch anhand neuer Messungen ermittelt. Die Ergebnisse, wie z. B. die Gesamtträgheit, können mit CYCLE759 gelesen werden. Anhand der neu berechneten Werte kann auch erkannt werden, ob sich die Regelstrecke geändert hat. Dann kann fallweise auf vordefinierte Reglerdatensätze umgeschaltet oder die Beschleunigung geändert werden. Die AST-Standard Einstellung beinhaltet die Drehzahl- und Lageregler-Optimierung mit dem Optimierungsziel "normal". Die ermittelten Reglerdaten werden in diesem Beispiel aber nicht aktiviert.

Programmierung

Definition der globalen Anwendervariablen (GUD)

Zum Auslesen der Gesamtträgheit muss in der (ggf. neu zu erstellenden) Datei MGUD.DEF eine neue Variable vom Typ CHAN REAL definiert werden:

```
; Definition der GUD zum Auslesen der Gesamtträgheit.
DEF CHAN REAL _AST_R_ESTINERTIA
```

Optimierungszyklus

Hinweis

Im folgenden Programmbeispiel sind Zyklenaufrufe, die für die Vermessung unbedingt erforderlich sind, durch **Fettschrift** hervorgehoben.

Nicht hervorgehobene Zyklenaufrufe sind optional.

```

DEF INT myaxiswithnewload=4
; In sicheren Datensatz wechseln.
; Achse auf sichere Position/Nullstellung fahren.
; Optimierungsstrategie über die Bedienoberfläche gesetzt?
; Ziel-Antriebsdatensatz (DDS) vorhanden, der überschrieben werden darf?

; Optimierungssitzung öffnen.
CYCLE751 (1)

; Achse 4 optimieren: Optimierung anhand gewählter Strategie auf Basis
neuer Messung.
; Optimierung mit Standardwerten (die Auswahl über die AST-Bedienoberfläche
wird nicht benutzt).
; Keine Bahninterpolation, es werden keine Werte durch AST geschrieben.
CYCLE752 (myaxiswithnewload,1,false)

; Originalzustand in Datei sichern.
CYCLE755(1,"restorepoint1")

; Optimierung starten ohne Aktivierung der Ergebnisse.
CYCLE751 (4)

; Optimierungsdaten in XML-Datei speichern.
CYCLE757(myaxiswithnewload,"axis_retuned.xml",1,,)

; Drehzahlreglerstrecke exportieren.
CYCLE757(myaxiswithnewload,"speedctrlplant.csv",3,,)

; Trägheit auslesen.
CYCLE759 (myaxiswithnewload,34,,,"_AST_R_ESTINERTIA")

; Ggf. Optimierungsergebnisse prüfen, danach aktivieren.
; Ergebnisse nicht aktivieren.
; Es werden vordefinierte Parametersätze verwendet.

; Optimierungssitzung schließen.
CYCLE751 (2)

```

14.6 Beispiele

```
; In neu optimierten DDS wechseln ODER vordefinierten DDS abhängig von der  
Lastträgheit wählen.  
; Beschleunigung abhängig von der neuen Trägheit begrenzen.
```

M17

14.6.2 Beispiel 2: Drehzahlregler einer Achse neu optimieren

Die Reglereinstellung der Rundachse (hier: 4. Achse), die mit einem Torque-Motor angetrieben wird, ist von der Aufspannvorrichtung und der Lastträgheit abhängig. Ändert sich die Aufspannung und Lastträgheit, muss die Achse neu vermessen werden. Es muss ein neuer Reglerparametersatz ermittelt werden (ein DDS wird überschrieben), weil die Achse aufgrund neuer Resonanzen sonst instabil ist. Es müssen wahrscheinlich neue Stromsollwertfilter gesetzt werden. Es wird nur der Drehzahlregler neu optimiert. Da die Lageregelung gleich bleiben und auch die Vorsteuerung nicht geändert werden soll, soll wieder eine ähnliche Bandbreite des Drehzahlreglers erreicht werden. Hierfür kann man in der Strategie der Drehzahlregloptimierung eine Zielbandbreite wählen. Diese ist bei Optimierungsziel "normale Ausregelung" auf 100 Hz eingestellt. Wenn aufgrund der neuen Lastträgheit und Aufspannung viele Filter notwendig sind und das Verhältnis aus Lastträgheit/Motortragheit hoch ist, kann dieses Ziel u. U. nicht erreicht werden. In diesem Fall wird eine Meldung ausgegeben. Die Achsmaschinendaten sind also noch von der ersten Optimierung (Erstinbetriebnahme) richtig gesetzt. D. h., dass der Vorsteuermodus (MD32620 \$MA_FFW_MODE), DSC (MD32640 \$MA_STIFFNESS_CONTROL_ENABLE), Lagereglerverstärkung (MD32200 \$MA_POSCTRL_GAIN) und Ersatzzeit Vorsteuerung (MD32800 \$MA_EQUIV_CURRCTRL_TIME oder MD32810 \$MA_EQUIV_SPEEDCTRL_TIME) richtig gesetzt sind, sodass die Kreisinterpolation stimmt.

Die Ergebnisse der Nachoptimierung, wie z. B. die Gesamtträgheit, können mit CYCLE759 gelesen werden. Der neu optimierte Drehzahlregler wird in einen anderen Datensatz geschrieben. Die Bahninterpolation wird hier nicht berücksichtigt, da man davon ausgeht, dass die Vorsteuer-Daten sich nicht wesentlich ändern werden oder eine andere Achse eine größere Zeitkonstante in MD32800 \$MA_EQUIV_CURRCTRL_TIME hat. Es kommt die Optimierungsstrategie für Achse und Drehzahlregler zur Anwendung, die über die Bedienoberfläche ausgewählt und in der XML-Datei für diese Achse (*/user/sinumerik/nck/data/optimization/AST_AX4_A1...xml*) gespeichert wurde.

Vorbereitung

- Über die Bedienoberfläche Optimierungsstrategie 105 auswählen:

Automat. Servo Optimierung: Auswahl der vordefinierten Strategie		AX4:A1	Strategie anpassen
Strategie: Achse			Messparameter
105. Drehzahlregler messen und neu optimieren			Achse
Vorabmessung zur Bestimmung der Anregung durchführen:	<input type="checkbox"/>		Drehzahl
nicht unterstützte Reglereigenschaften in NC und Antrieb deaktivieren:	<input checked="" type="checkbox"/>		Position
Messe geglätteten Drehzahlsollwertsprung zur Erstellung eines Mechanikmodells:	<input type="checkbox"/>	▶	
Drehzahlregelstrecke messen:	<input checked="" type="checkbox"/>	▶	
Überprüfung der Messdatenqualität:	<input checked="" type="checkbox"/>		
Drehzahlregelstrecke bei reduzierter Bandbreite messen:	<input checked="" type="checkbox"/>	▶	
Überprüfung der Messdatenqualität:	<input checked="" type="checkbox"/>		
Drehzahlregelkreis für Drehzahlregler-Streckenmodell messen:	<input type="checkbox"/>	▶	
Überprüfung der Messdatenqualität:	<input type="checkbox"/>		
Optimieren Drehzahlregler:	<input checked="" type="checkbox"/>		
Dehzahlregelkreis zwecks Überprüfung messen:	<input type="checkbox"/>	▶	
Vermessen Mechanik:	<input type="checkbox"/>	▶	
Überprüfung der Messdatenqualität:	<input type="checkbox"/>		
Optimieren Lageregler:	<input type="checkbox"/>		
Kontrollmessung Lageregelkreis:	<input type="checkbox"/>	▶	
			Abbruch
			OK
Stromregelkreis	Drehzahlregelkreis	Lageregelkreis	Funktionsgenerator
Kreisformtest	Aktive Filter	Auto Servo Optim.	

- Ggf. Optimierungsziel (schnell, moderat, robust) ändern (über den Softkey "Drehzahl").

Programmierung

Definition der globalen Anwendervariablen (GUD)

Zum Auslesen der Gesamtträgheit muss in der (ggf. neu zu erstellenden) Datei MGUD.DEF eine neue Variable vom Typ CHAN REAL definiert werden:

```
; Definition der GUD zum Auslesen der Gesamtträgheit.
DEF CHAN REAL _AST_R_ESTINERTIA
```

Optimierungszyklus

Hinweis

Im folgenden Programmbeispiel sind Zyklenaufrufe, die für die Nachoptimierung unbedingt erforderlich sind, durch **Fettschrift** hervorgehoben.

Nicht hervorgehobene Zyklenaufrufe sind optional.

```
DEF INT myaxiswithnewload=4
; In sicheren Datensatz wechseln.
```

14.6 Beispiele

```
; Achse auf sichere Position/Nullstellung fahren.
; Optimierungsstrategie über die Bedienoberfläche gesetzt?
; Ziel-Antriebsdatensatz (DDS) vorhanden, der überschrieben werden darf?

; Optimierungssitzung öffnen.
CYCLE751 (1)

; Achse 4 optimieren: Optimierung anhand gewählter Strategie auf Basis
neuer Messung.

; Achse 4 hinzufügen.
; Optimierungsstrategie aus Standard-Optimierungsdatei (XML-Datei)
übernehmen.
; Keine Berücksichtigung der Achse bei einer Optimierung der
Bahninterpolation.
CYCLE752 (myaxiswithnewload,3,false)

; Aktuellen DDS aus Datensatzliste entfernen.
CYCLE754 (myaxiswithnewload,2,2,-1)

; DDS3 zur Datensatzliste hinzufügen und nach Optimierung überschreiben.
CYCLE754 (myaxiswithnewload,1,2,3)

; Originalzustand in Datei sichern.
CYCLE755(1,"restorepoint1")

; Optimierung starten mit Aktivierung der Ergebnisse.
; CYCLE751(3)

; Optimierung starten ohne Aktivierung der Ergebnisse, ggf. Ergebnisse erst
prüfen.
CYCLE751 (4)

; Optimierungsdaten in XML-Datei speichern.
CYCLE757(myaxiswithnewload,"axis_retuned.xml",1,,)

; Drehzahlreglerstrecke exportieren.
CYCLE757(myaxiswithnewload,"speedctrlplant.csv",3,,)

; Trägheit kann zur Begrenzung der Beschleunigung verwendet werden:
; Trägheit auslesen.
CYCLE759(myaxiswithnewload,34,,,"_AST_R_ESTINERTIA")

; Ggf. Optimierungsergebnisse prüfen, danach aktivieren.

; Ergebnis aktivieren:
```

```
; Achsoptimale Ergebnisse aktivieren.  
; Bootfiles sichern.  
CYCLE756 (myaxiswithnewload,2,1)  
  
; Optimierungssitzung schließen.  
CYCLE751 (2)  
  
; In neu optimierten DDS wechseln ODER Beschleunigung abhängig von der  
neuen Trägheit begrenzen.
```

M17

14.6.3 Beispiel 3: Drehzahlregler und Lageregler einer Achse neu optimieren

Die Reglereinstellung der Rundachse (hier: 4. Achse), die mit einem Torque-Motor angetrieben wird, ist von der Aufspannvorrichtung und der Lastträgheit abhängig. Ändert sich die Aufspannung und Lastträgheit, muss die Achse neu vermessen werden. Es müssen wahrscheinlich neue Stromsollwertfilter gesetzt werden. Der Drehzahlregler muss neu optimiert werden. Da nicht bekannt ist, ob sich das niederfrequente Verhalten verändert hat, muss auch der Lageregler neu optimiert und dafür der Mechanikfrequenzgang gemessen werden.

Die Ergebnisse, wie z. B. Gesamtträgheit, können mit CYCLE759 gelesen werden. Der neu optimierte Drehzahlregler wird in einen anderen Datensatz geschrieben. Der neu optimierte Lageregler kann in einen gewünschten Achsparametersatz geschrieben werden. Die korrekte Bahninterpolation wird sichergestellt, indem ein Zielwert für das (vorgesteuerte) Positionierverhalten entsprechend den bereits optimierten Achsen vorgegeben wird. Es kommt die Optimierungsstrategie für Achse und Drehzahlregler zur Anwendung, die über die Bedienoberfläche ausgewählt und in der XML-Datei für diese Achse (*/user/sinumerik/nck/data/optimization/AST_AX4_A1....xm*) gespeichert wurde.

Vorbereitung

- Über die Bedienoberfläche Optimierungsstrategie 102 auswählen:

Automat. Servo Optimierung: Auswahl der vordefinierten Strategie			AX4:A1	Strategie anpassen							
Strategie: Achse				Messparameter							
102. Drehzahl- und Lageregler messen und neu optimieren				Achse							
Vorabmessung zur Bestimmung der Anregung durchführen:	<input type="checkbox"/>			Drehzahl							
nicht unterstützte Reglereigenschaften in NC und Antrieb deaktivieren:	<input checked="" type="checkbox"/>			Position							
Messe geglätteten Drehzahlsollwertsprung zur Erstellung eines Mechanikmodells:	<input type="checkbox"/>	▶									
Drehzahlregelstrecke messen:	<input checked="" type="checkbox"/>	▶									
Überprüfung der Messdatenqualität:	<input checked="" type="checkbox"/>										
Drehzahlregelstrecke bei reduzierter Bandbreite messen:	<input checked="" type="checkbox"/>	▶									
Überprüfung der Messdatenqualität:	<input checked="" type="checkbox"/>										
Drehzahlregelkreis für Drehzahlregler-Streckenmodell messen:	<input type="checkbox"/>	▶									
Überprüfung der Messdatenqualität:	<input type="checkbox"/>										
Optimieren Drehzahlregler:	<input checked="" type="checkbox"/>										
Dehzahlregelkreis zwecks Überprüfung messen:	<input type="checkbox"/>	▶									
Vermessen Mechanik:	<input type="checkbox"/>	▶									
Überprüfung der Messdatenqualität:	<input type="checkbox"/>										
Optimieren Lageregler:	<input checked="" type="checkbox"/>			Abbruch							
Kontrollmessung Lageregelkreis:	<input type="checkbox"/>	▶		OK							
<table border="1"> <tr> <td>Stromregelkreis</td> <td>Drehzahlregelkreis</td> <td>Lageregelkreis</td> <td>Funktionsgenerator</td> <td>Kreisformtest</td> <td>Aktive Filter</td> <td>Auto Servo Optim.</td> </tr> </table>				Stromregelkreis	Drehzahlregelkreis	Lageregelkreis	Funktionsgenerator	Kreisformtest	Aktive Filter	Auto Servo Optim.	
Stromregelkreis	Drehzahlregelkreis	Lageregelkreis	Funktionsgenerator	Kreisformtest	Aktive Filter	Auto Servo Optim.					

- Über den Softkey "Position" das Fenster zum Anpassen der Lagereglerstrategie einblenden und "Benutzerdefinierte Strategie 209" auswählen:

Automat. Servo Optimierung: Auswahl der vordefinierten Strategie			AX4:A1	Strategie anpassen
Strategie: Lageregelkreis				
Benutzerdefinierte Strategie (209)				
DSC aktiv:	<input checked="" type="checkbox"/>			
Kv Reduzierung:		0.6		
Kv-Obergrenze:		4	1000/min	
Vorsteuerungsart:		Moment		Achse
FFW immer aktiv:	<input checked="" type="checkbox"/>			
Methode für maximalen Kv:		Standard		
Zielwert Äquivalenzzeit erzwingen:	<input checked="" type="checkbox"/>			Drehzahl
Zielwert Äquivalenzzeit:	0.003		s	
Drehzahlsollwertfilter (1-2):				Position
Index	Filter		Res	Filter Details
1	nicht verwendet		<input type="checkbox"/>	
2	nicht verwendet		<input type="checkbox"/>	
				Abbruch
				OK
Stromregelkreis	Drehzahlregelkreis	Lageregelkreis	Funktionsgenerator	Kreisformtest
			Aktive Filter	Auto Servo Optim.

- Option "Zielwert Äquivalenzzeit erzwingen" anwählen.

Hinweis

Die Option "Zielwert Äquivalenzzeit erzwingen" wird gesetzt, wenn zusätzlich zum Drehzahlregler auch der Lageregler neu optimiert und trotzdem eine Ersatzzeit der Vorsteuerung vorgegeben, die Bahninterpolation aber nicht neu optimiert werden soll.

Zum Sicherstellen der korrekten Bahninterpolation kann eine Ersatzzeit vorgegeben werden. Diese sollte der Ersatzzeit der bereits auf die Bahn optimierten Achsen entsprechen.

- Die Ersatzzeit (MD32800 \$MA_EQUIV_CURRCTRL_TIME oder MD32810 \$MA_EQUIV_SPEEDCTRL_TIME, je nach MD32620 \$MA_FFW_MODE) aus den anderen Achsen übernehmen (hier z. B. 3 ms).
- Im Servo-Parametersatz müssen folgende Parameter gesetzt sein:
 - Ersatzzeit Vorsteuerung (MD32800 \$MA_EQUIV_CURRCTRL_TIME) = 3 ms
 - Lagereglerverstärkung (MD32200 \$MA_POSCTRL_GAIN)

Hinweis

Für den Automatikbetrieb muss dann in den anderen Achsen der Servo-Parametersatz mit den passenden Bahn-optimalen Werten angewählt sein (hier: mit Ersatzzeit = 3 ms).

Programmierung

Definition der globalen Anwendervariablen (GUD)

Zum Auslesen der Gesamtträgheit muss in der (ggf. neu zu erstellenden) Datei MGUD.DEF eine neue Variable vom Typ CHAN REAL definiert werden:

```
; Definition der GUD zum Auslesen der Gesamtträgheit.  
DEF CHAN REAL _AST_R_ESTINERTIA
```

Optimierungszyklus

Hinweis

Im folgenden Programmbeispiel sind Zyklenaufrufe, die für die Nachoptimierung unbedingt erforderlich sind, durch **Fettschrift** hervorgehoben.

Nicht hervorgehobene Zyklenaufrufe sind optional.

```
DEF INT myaxiswithnewload=4  
; In sicheren Datensatz wechseln.  
; Achse auf sichere Position/Nullstellung fahren.  
; Strategieeinstellungen wurden über die Bedienoberfläche getroffen:  
Zielwert für Ersatzzeit!  
; Ziel-Antriebsdatensatz (DDS) vorhanden, der überschrieben werden darf?  
  
; Optimierungssitzung öffnen.  
CYCLE751 (1)  
  
; Achse 4 optimieren: Optimierung anhand gewählter Strategie auf Basis  
neuer Messung.  
; Achse 4 hinzufügen.  
; Optimierungsstrategie aus Standard-Optimierungsdatei (XML-Datei)  
übernehmen.  
; Keine Berücksichtigung der Achse bei einer Optimierung der  
Bahninterpolation.  
CYCLE752 (myaxiswithnewload,3,false)  
  
; Aktuellen DDS aus Datensatzliste entfernen.  
CYCLE754 (myaxiswithnewload,2,2,-1)  
  
; DDS3 zur Datensatzliste hinzufügen und nach Optimierung überschreiben.  
CYCLE754 (myaxiswithnewload,1,2,3)  
  
; Achse: Parametersatz 4 zur Datensatzliste hinzufügen und nach Optimierung  
überschreiben.  
CYCLE754 (myaxiswithnewload,1,1,3)
```

```
; Originalzustand in Datei sichern.
CYCLE755(1,"restorepoint1")

; Optimierung starten mit Aktivierung der Ergebnisse.
; CYCLE751(3)

; Optimierung starten ohne Aktivierung der Ergebnisse, ggf. Ergebnisse erst
prüfen.
CYCLE751 (4)

; Optimierungsdaten in XML-Datei speichern.
CYCLE757(myaxiswithnewload,"axis_retuned.xml",1,,)

; Drehzahlreglerstrecke exportieren.
CYCLE757(myaxiswithnewload,"speedctrlplant.csv",3,,)

; Trägheit kann zur Begrenzung der Beschleunigung verwendet werden:

; Trägheit auslesen.
CYCLE759(myaxiswithnewload,34,,,"_AST_R_ESTINERTIA")

; Ggf. Optimierungsergebnisse prüfen, danach aktivieren.

; Ergebnis aktivieren:
; Achsoptimale Ergebnisse aktivieren.
; Bootfiles sichern.
CYCLE756(myaxiswithnewload,2,1)

; Optimierungssitzung schließen.
CYCLE751 (2)

; In neu optimierten DDS wechseln.
; Kv wurde optimiert, in aktuellem Parametersatz?
; Ersatzzeit der Vorsteuerung wurde durch die Strategie erzwungen.
; Für Bahninterpolation müssen alle Achsen den richtigen Servo-
Parametersatz anwählen.
; Beschleunigung abhängig von der neuen Trägheit begrenzen.
```

M17

14.6.4 Beispiel 4: Nachoptimieren der Bahninterpolation

Bei einer 5-Achs-Maschine muss die Rundachse mit Torque-Motor (hier: 4. Achse) sehr genau beim Positionieren sein und an der Bahninterpolation teilnehmen. Die Änderungen der Aufspannsituation und Trägheit sind völlig unbekannt oder so groß, sodass auch große Änderungen der Ersatzzeit (auch Stromregler) erwartet werden. Durch die unbekanntes Ersatzzeit muss auch die Bahninterpolation nach der Achsoptimierung nochmals geprüft und angeglichen werden.

Es soll nur die Rundachse (werkstückabhängig) nachoptimiert werden. Die anderen Achsen müssen ggf. nur in der Vorsteuerung angepasst werden, falls sich die Dynamik der Rundachse stark geändert hat. Die Rundachse hat mehrere DDS. Es wird ein sicherer Datensatz angewählt, der für das Spannen der Werkstücke eine stabile Einstellung hat. Dabei muss berücksichtigt werden, dass die zu spannende Massenträgheit nicht bekannt ist. Man muss also mit der maximalen Trägheit rechnen.

Bei der Rundachse müssen wahrscheinlich neue Stromsollwertfilter gesetzt werden. Der Drehzahlregler muss neu optimiert werden. Da nicht bekannt ist, ob sich das niederfrequente Verhalten verändert hat, muss auch der Lageregler neu optimiert und dafür der Mechanikfrequenzgang gemessen werden.

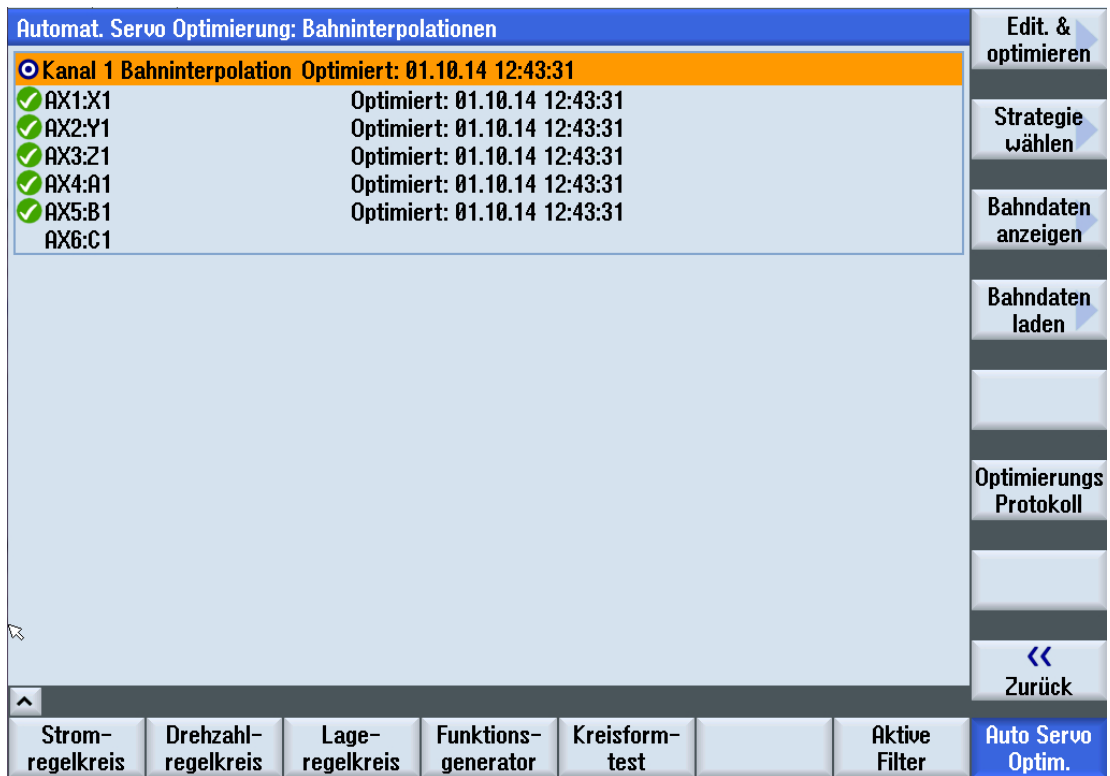
Die Ergebnisse, wie z. B. die Gesamtträgheit, können mit CYCLE759 gelesen werden. Der neu optimierte Drehzahlregler wird in einen anderen Datensatz geschrieben. Der neu optimierte Lageregler kann in einen gewünschten Achsparametersatz geschrieben werden. Die korrekte Bahninterpolation wird sichergestellt, indem alle an der Bahn beteiligten Achsen durch AST optimiert wurden und nun nach der Nachoptimierung der Achse die Optimierung der Bahninterpolation von AST nochmals durchlaufen wird. Es kommt die Optimierungsstrategie für Achse und Drehzahlregler zur Anwendung, die über die Bedienoberfläche ausgewählt wurde und in der XML-Datei für diese Achse (*/user/sinumerik/nck/data/optimization/AST_AX4_A1....xml*) gespeichert wird. Es kommt die Bahnoptimierungsstrategie zur Anwendung, die über die Bedienoberfläche eingestellt wurde und in der XML-Datei für die Optimierung der Bahninterpolation gespeichert wird.

Vorbereitung

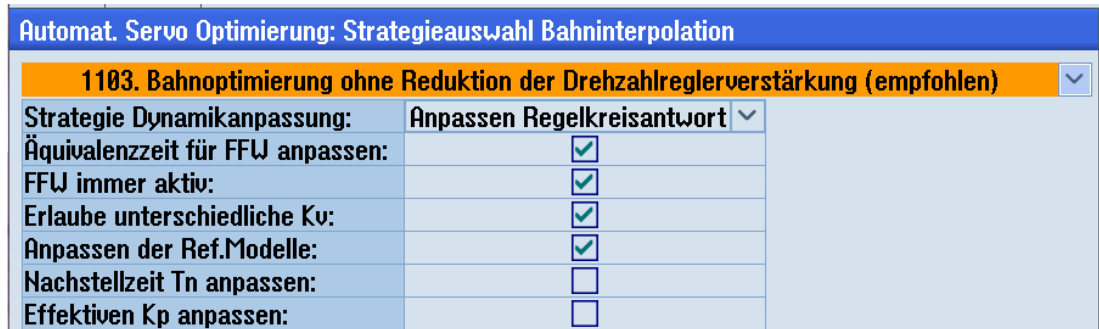
- Es wird keine Äquivalenzzeit erzwungen. Also können die vordefinierten Strategien für die Lageregleroptimierung verwendet werden:

Automat. Servo Optimierung: Auswahl der vordefinierten Strategie			AX4:A1	Strategie anpassen		
Strategie: Lageregelkreis						
203. Kv ohne Drehzahlsollwertfilter und FFW maximieren						
Optimierungsziel:			Normale Ausregelung			
DSC aktiv:	<input checked="" type="checkbox"/>					
Kv Reduzierung:		0.6				
Kv-Obergrenze:		4 1000/min		Achse		
Vorsteuerungsart:		Moment				
FFW immer aktiv:	<input checked="" type="checkbox"/>					
Methode für maximalen Kv:		Standard		Drehzahl		
Zielwert Äquivalenzzeit erzwingen:	<input type="checkbox"/>					
Drehzahlsollwertfilter (1-2):				Position		
Index	Filter		Res			
1	nicht verwendet		<input type="checkbox"/>	Filter Details		
2	nicht verwendet		<input type="checkbox"/>	Abbruch		
				OK		
Stromregelkreis	Drehzahlregelkreis	Lageregelkreis	Funktionsgenerator	Kreisformtest	Aktive Filter	Auto Servo Optim.

- Bei der Bahninterpolation sollte die Wahl der Strategie (alle Ersatzzeiten gleich oder MD32895 \$MA_DESVAL_DELAY_TIME benutzen) bereits für die erste Optimierung getroffen worden sein.



Mit folgender Auswahl werden die Symmetrierzeiten (in MD32800 \$MA_EQUIV_CURRCTRL_TIME oder MD32810 \$MA_EQUIV_SPEEDCTRL_TIME) aller beteiligter Achsen auf die größte Zeitkonstante gesetzt:



- Folgende Fragen sind zu klären:
 - Welcher Achs-Parametersatz darf neu parametrieren werden (für die Bahninterpolation)?
 - Darf der erste Achs-Parametersatz überschrieben werden?
 - Muss die Spindel Teil der Bahninterpolation sein?
 - Wenn das Gewindeschneiden auch mit dieser Achskonfiguration laufen soll, dann müssen ggf. der Parametersatz 2 für die Achsen und in der Spindel die Getriebestufen (Parametersatz 2 ... 6) optimiert werden.
- **Originale Bahn-Daten sichern!**

Programmierung

Definition der globalen Anwendervariablen (GUD)

Zum Auslesen der Gesamttr agheit muss in der (ggf. neu zu erstellenden) Datei MGUD.DEF eine neue Variable vom Typ CHAN REAL definiert werden:

```
; Definition der GUD zum Auslesen der Gesamttr agheit.
DEF CHAN REAL _AST_R_ESTINERTIA
```

Optimierungszyklus

Hinweis

Im folgenden Programmbeispiel sind Zyklenaufrufe, die f ur die Nachoptimierung unbedingt erforderlich sind, durch **Fettschrift** hervorgehoben.

Nicht hervorgehobene Zyklenaufrufe sind optional.

```
DEF INT myaxiswithnewload=4
; In sicheren Datensatz wechseln.
; Achse auf sichere Position/Nullstellung fahren.
; Strategieeinstellungen wurden  ber die Bedienoberfl ache getroffen:
Zielwert f ur Ersatzzeit!
; Ziel-Antriebsdatensatz (DDS) vorhanden, der  berschrieben werden darf?
; Jede Achse wurde bereits einmal mit AST ( ber die Masken) optimiert.
; Somit existiert die Standarddatei f ur CYCLE752 f ur jede Achse.

; Optimierungssitzung  ffnen.
CYCLE751 (1)

; Achse 4 optimieren, mit Bahninterpolation.
; Strategie w ahlen:
; Achse 4 hinzuf ugen.
; Optimierungsstrategie aus Standard-Optimierungsdatei (XML-Datei)
 bernehmen.
; Ber ucksichtigung der Achse bei einer Optimierung der Bahninterpolation.
CYCLE752 (myaxiswithnewload, 3, true)

; Achse 4 neu vermessen und nachoptimieren, Bootfiles sichern.
CYCLE753 (myaxiswithnewload, 3, 1)

; Achse 1 hinzuf ugen wegen Bahninterpolation.
CYCLE752 (1, 3, true)
; Achse 1 nur Bahninterpolation nachoptimieren.
CYCLE753 (1, 1, true)

; Achse 2 hinzuf ugen wegen Bahninterpolation.
```

14.6 Beispiele

```
CYCLE752 (2,3,true)
; Achse 2 nur Bahninterpolation nachoptimieren.
CYCLE753 (2,1,true)

; Achse 3 hinzufügen wegen Bahninterpolation.
CYCLE752 (3,3,true)
; Achse 3 nur Bahninterpolation nachoptimieren.
CYCLE753 (3,1,true)

; Achse 6 hinzufügen wegen Bahninterpolation.
CYCLE752 (6,3,true)
; Achse 6 nur Bahninterpolation nachoptimieren.
CYCLE753 (6,1,true)

; Aktuellen DDS aus Datensatzliste entfernen.
CYCLE754 (myaxiswithnewload,2,2,-1)
; DDS3 zur Datensatzliste hinzufügen und nach Optimierung überschreiben.
CYCLE754 (myaxiswithnewload,1,2,3)

; Achse: Parametersatz 4 zur Datensatzliste hinzufügen und nach Optimierung
überschreiben.
CYCLE754 (myaxiswithnewload,1,1,3)

; Originalzustand in Datei sichern.
CYCLE755(1,"restorepoint1")

; Optimierung starten mit Aktivierung der Ergebnisse.
CYCLE751 (3)

; Optimierungsdaten in XML-Datei speichern.
CYCLE757(myaxiswithnewload,"axis_retuned.xml",1,,)

; Messdaten der Drehzahlregelstrecke in Datei speichern.
CYCLE757(myaxiswithnewload,"speedctrlplant.csv",3,,)

; Ergebnis der Bahnoptimierung in Datei speichern.
CYCLE757(1,"path_retuned.xml",2,0,0)

; Trägheit auslesen.
CYCLE759(myaxiswithnewload,34,,,"_AST_R_ESTINERTIA")

; Optimierungssitzung schließen.
CYCLE751 (2)

; In neu optimierten DDS wechseln.
```


- ; Kv wurde optimiert, in entsprechenden Parametersatz wechseln.
- ; Bahninterpolation wurde neu optimiert.
- ; Beschleunigung in Achse 4 abhängig von der neuen Trägheit begrenzen.

M17

14.6.5 Beispiel 5: Nachoptimieren des Drehzahlregelkreises zur Eliminierung bekannter periodischer Störfrequenzen

In diesem Beispiel wird gezeigt, wie der Drehzahlregler aus dem Teileprogramm heraus ohne erneute Messung nachoptimiert wird. Ziel der Nachoptimierung ist, die Anfälligkeit gegenüber bekannten prozessbedingten periodischen Störfrequenzen zu minimieren. Die Bahninterpolation wird in diesem Beispiel nicht nachoptimiert, weil die Optimierungsstrategie für den Lageregler die Dynamikanpassung durch Setzen einer Äquivalenzzeit vorsieht.

Vorbereitung

Im einfachsten Fall ist nur eine Drehzahlregleroptimierung notwendig:

- Über die Bedienoberfläche Optimierungsstrategie 105 auswählen:

Automat. Servo Optimierung: Auswahl der vordefinierten Strategie		AX4:A1	Strategie anpassen
Strategie: Achse			Messparameter
105. Drehzahlregler messen und neu optimieren			Achse
Vorabmessung zur Bestimmung der Anregung durchführen:	<input type="checkbox"/>		Drehzahl
nicht unterstützte Reglereigenschaften in NC und Antrieb deaktivieren:	<input checked="" type="checkbox"/>		Position
Messe geglätteten Drehzahlsollwertsprung zur Erstellung eines Mechanikmodells:	<input type="checkbox"/>	▶	
Drehzahlregelstrecke messen:	<input checked="" type="checkbox"/>	▶	
Überprüfung der Messdatenqualität:	<input checked="" type="checkbox"/>		
Drehzahlregelstrecke bei reduzierter Bandbreite messen:	<input checked="" type="checkbox"/>	▶	
Überprüfung der Messdatenqualität:	<input checked="" type="checkbox"/>		
Drehzahlregelkreis für Drehzahlregler-Streckenmodell messen:	<input type="checkbox"/>	▶	
Überprüfung der Messdatenqualität:	<input type="checkbox"/>		
Optimieren Drehzahlregler:	<input checked="" type="checkbox"/>		
Dehzahlregelkreis zwecks Überprüfung messen:	<input type="checkbox"/>	▶	
Vermessen Mechanik:	<input type="checkbox"/>	▶	
Überprüfung der Messdatenqualität:	<input type="checkbox"/>		
Optimieren Lageregler:	<input type="checkbox"/>		
Kontrollmessung Lagereglerkreis:	<input type="checkbox"/>	▶	
			Abbruch
			OK
Stromregelkreis	Drehzahlregelkreis	Lageregelkreis	Funktionsgenerator
Kreisformtest	Aktive Filter	Auto Servo Optim.	

Programmierung

```
DEF INT myAxis=4
```

14.6 Beispiele

```
; Optimierungssitzung öffnen.
CYCLE751(1)

; Achse hinzufügen.
; Die Optimierungsdaten werden aus der Standard-Optimierungsdatei geladen.
; Keine Berücksichtigung der Achse bei einer Optimierung der
Bahninterpolation.
CYCLE752(myAxis,3,false,)

; Die Achsdynamik wird nicht neu vermessen.
; Die Achse wird auf Grundlage der geladenen Messergebnisse und der aktuell
gültigen Optimierungsstrategie nachoptimiert.
CYCLE753(myAxis,2,)

; Verwendung des Verstärkungsfilters einschalten.
CYCLE758(myAxis,208,,,"true" )

; Verstärkungsfilter: optimale Frequenz auf 40 Hz einstellen.
CYCLE758(myAxis,209,,,"40.0" )

; Optimierung ausführen.
CYCLE751(3)

; Achsspezifische Optimierungsdaten für die Offline-Analyse speichern.
CYCLE757(myAxis,"AX1_TEMP_40HZ.XML",1,,)

; Bei bekannten periodischen Störfrequenzen (40 Hz):

; Verwendung des Verstärkungsfilters ausschalten.
CYCLE758(myAxis,208,,,"false" )

; Optimierung ausführen.
CYCLE751(3)

; Optimierungssitzung schließen.
CYCLE751(2)

; Bei normalen Breitbandstörungen:
```

M17

14.6.6 Beispiel 6: Vermessen einer Achse ohne Optimierung

Durch Anwahl einer Strategie, die keine Optimierung durchführt, kann ein automatisches Vermessen der Achse durchgeführt werden. Hierdurch können mechanische Veränderungen diagnostiziert werden. Diese können betreffen:

- Lastmasse
- Steifigkeit der Komponenten

Durch die automatische Vermessung kann auch geprüft werden, ob der geschlossene Regelkreis das gleiche Verhalten wie z. B. bei Auslieferung aufweist. Diese AST-Strategie liefert dann die aktuelle Messung des geschlossenen Kreises im CSV-Format.

Dabei können automatisch folgende Daten gemessen und exportiert werden:

- Drehzahlregelstrecke
- Drehzahlregelkreis geschlossen (berechnet und/oder Überprüfungsmessung)
- Mechanikfrequenzgang (wenn 2 Geber vorhanden)
- Lageregelkreis geschlossen (berechnet und/oder Überprüfungsmessung des Lageregelkreises)

Vorbereitung

1. Über die Bedienoberfläche Strategie 109 auswählen:

Automat. Servo Optimierung: Auswahl der vordefinierten Strategie		AX1:X1	Strategie anpassen
Strategie: Achse			Messparameter
109. Messung und Überprüfung der vorhandenen Reglereinstellungen			Achse
Vorabmessung zur Bestimmung der Anregung durchführen:	<input type="checkbox"/>		Drehzahl
nicht unterstützte Reglereigenschaften in NC und Antrieb deaktivieren:	<input type="checkbox"/>		Position
Messe geglätteten Drehzahlsollwertsprung zur Erstellung eines Mechanikmodells:	<input type="checkbox"/>	▶	
Drehzahlregelstrecke messen:	<input checked="" type="checkbox"/>	▶	
Überprüfung der Messdatenqualität:	<input checked="" type="checkbox"/>		
Drehzahlregelstrecke bei reduzierter Bandbreite messen:	<input checked="" type="checkbox"/>	▶	
Überprüfung der Messdatenqualität:	<input checked="" type="checkbox"/>		
Drehzahlregelkreis für Drehzahlregler-Streckenmodell messen:	<input type="checkbox"/>	▶	
Überprüfung der Messdatenqualität:	<input type="checkbox"/>		
Optimieren Drehzahlregler:	<input type="checkbox"/>		
Dehzahlregelkreis zwecks Überprüfung messen:	<input type="checkbox"/>	▶	
Vermessen Mechanik:	<input checked="" type="checkbox"/>	▶	
Überprüfung der Messdatenqualität:	<input checked="" type="checkbox"/>		
Optimieren Lageregler:	<input type="checkbox"/>		
Kontrollmessung Lageregelkreis:	<input type="checkbox"/>	▶	
			Abbruch
			OK
Stromregelkreis	Drehzahlregelkreis	Lageregelkreis	Funktionsgenerator
Kreisformtest	Aktive Filter	Automat. Servo Optim.	

2. Reglereinstellungen messen und überprüfen.

14.6 Beispiele

3. Messergebnisse übernehmen.
4. Daten unter einem neuen Namen (z. B. RESULT_STRATEGY109_AX1.xml) als benutzerdefinierte Datei speichern (damit die Standard-Optimierungsdatei nicht überschrieben wird).
Die benutzerdefinierte Datei wird standardmäßig im Verzeichnis */user/sinumerik/hmi/log/optimization* abgelegt.

Programmierung

Beispiel für einen Zyklus zum Vermessen einer Achse ohne Optimierung:

```
; Bremse lösen
DEF INT AxNr=1

; Optimierungssitzung öffnen.
CYCLE751(1)

; Für Achse 1 werden die gespeicherten Daten aus der benutzerdefinierten
Datei RESULT_STRATEGY109_AX1.xml verwendet.
CYCLE752(AxNr,2,0,"/user/sinumerik/hmi/log/optimization/
RESULT_STRATEGY109_AX1.xml")

; Automatische Vermessung starten.
CYCLE751(4)

; Drehzahlregelstrecke in CSV-Format exportieren.
CYCLE757(AxNr,"speedctrlplant.csv",3)

; Mechanikfrequenzgang in CSV-Format exportieren.
CYCLE757(AxNr,"mechresponse.csv",4)

; Drehzahlregelkreis (berechnet) in CSV-Format exportieren.
CYCLE757(AxNr,"speedloopcalc.csv",5)

; Lageregelkreis (berechnet) in CSV-Format exportieren.
CYCLE757(AxNr,"posloopcalc.csv",6)

; Drehzahlregelkreis (gemessen) in CSV-Format exportieren.
CYCLE757(AxNr,"speedloopmeas.csv",7)

; Lageregelkreis (gemessen) in CSV-Format exportieren.
CYCLE757(AxNr,"posloopmeas.csv",8)

; Optimierungssitzung schließen.
CYCLE751(2)
```

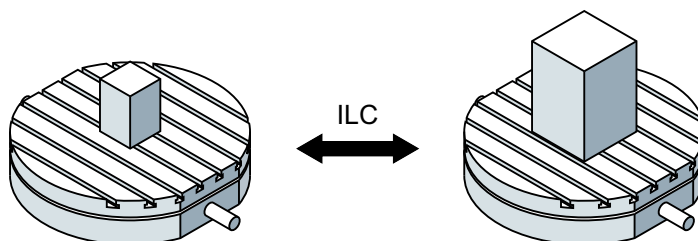
M17

15.1 Funktion

Hinweis

Die Funktion "Intelligente Lastanpassung" ist eine lizenzpflichtige Option für SINUMERIK 840D sl.

Artikelnummer: 6FC5800-0AS11-0YB0



Mit der Funktion "Intelligente Lastanpassung" (Intelligent Load Control, ILC) ist es möglich, Dynamik- und Regelungsparameter mit Hilfe vordefinierter Adaptionenkennlinien an die aktuelle Beladung/Last einer Rundachse (Rundtisch, Werkzeugspindel, ...) anzupassen.

Daraus ergeben sich folgende Vorteile:

- Kürzere Bearbeitungszeiten
- Gesteigerte Dynamik
- Bessere Regelgüte
- Höhere Genauigkeit

Eingangsgröße und Eingangswert

Als Eingangsgröße für Adaptionen wird die von der Beladung abhängige Trägheit der Achse verwendet. Der zu bestimmende aktuelle Wert der Eingangsgröße dient als Eingangswert für die Adaptionberechnungen.

Ausgangsgröße und Adaptionsfaktor

Auf Grundlage des Eingangswerts und der vordefinierten Adaptionenkennlinie berechnet die Steuerung einen Skalierungsgröße (Adaptionsfaktor) für die Ausgangsgröße der Adaption (Dynamik- oder Regelungsparameter).

Inbetriebnahme

Bei der Inbetriebnahme der Funktion definiert der Maschinenhersteller die an der Werkzeugmaschine verfügbaren Adaptionen. Dabei wird er durch Dialoge auf der Bedienoberfläche unterstützt: Bedienbereich "Inbetriebnahme" → "NC" → Menüfortschalt-Taste → Softkey: "Adaptionen"

Literatur:

Ausführliche Informationen siehe "Inbetriebnahmehandbuch CNC: NC, PLC, Antrieb".

Die über die Bedienoberfläche vorgenommenen Einstellungen werden in die zugehörigen Maschinendaten (Seite 684) geschrieben.

Aktivierung/Deaktivierung

Nach Steuerungshochlauf sind die Adaptionen zunächst inaktiv. Die Aktivierung/Deaktivierung erfolgt über den Aufruf von CYCLE782. Mit diesem Zyklus wird auch der Eingangswert ermittelt bzw. vorgegeben.

Literatur:

Informationen zur Bedienung und Programmierung von CYCLE782 siehe:

- Bedienhandbuch Universal/Drehen/Fräsen/Schleifen, Kapitel "Technologische Funktion programmieren (Zyklen)" > "Beladungsabhängige Reglereinstellungen (CYCLE782)"
- Programmierhandbuch Arbeitsvorbereitung, Kapitel "Zyklen extern programmieren" > "Technologische Zyklen" > "CYCLE782 - An Beladung anpassen (Option)"

15.2 Parametrierung

Aktivierungsmodus einer Adaption

Der Aktivierungsmodus einer Adaption wird eingestellt mit dem Maschinendatum:

MD16501 \$MN_CADAPT_MODE[<n>]

mit <n> = Nummer der Adaption (0 ... 99)

Wert	Bedeutung
0	Adaption ist inaktiv (Standardeinstellung)
1	Adaption ist permanent aktiv Nach Aktivierung mit CYCLE782 bleiben permanent aktive Adaptionen über Programmende- und Kanal-Reset hinaus aktiv. Ein zuvor programmierter Eingangswert bleibt ebenfalls wirksam.
2	Adaption ist temporär aktivierbar Temporär aktivierbare Adaptionen werden über den Aufruf von CYCLE782 aktiviert/deaktiviert. Mit Programmende- bzw. Kanal-Reset werden alle noch nicht per CYCLE782 deaktivierten Adaptionen implizit deaktiviert.

Eingangsgröße einer Adaption

Die Eingangsgröße einer Adaption wird festgelegt über das Maschinendatum:

MD16502 \$MN_CADAPT_INPUT[<n>]

mit <n> = Nummer der Adaption (0 ... 99)

Wert	Bedeutung
1	Eingangsgröße für die Adaption ist die Trägheit der Achse

Ausgangsgröße einer Adaption

Die Auswahl der Ausgangsgröße erfolgt über das Maschinendatum:

MD16503 \$MN_CADAPT_OUTPUT[<n>]

mit <n> = Nummer der Adaption (0 ... 99)

Wert	Bedeutung
0	Keine Ausgangsgröße, keine Adaption wirksam (Standardeinstellung)
1	Achsbeschleunigung mit Dynamikmodus DYNNORM
2	Achsbeschleunigung mit Dynamikmodus DYNPOS
3	Achsbeschleunigung mit Dynamikmodus DYNROUGH
4	Achsbeschleunigung mit Dynamikmodus DYNSEMIFIN
5	Achsbeschleunigung mit Dynamikmodus DYNFINISH
11	Lagereglerverstärkung (KV-Faktor)
12	Trägheit der Momentenvorsteuerung
21	Antriebsadaptionfaktor 1
22	Antriebsadaptionfaktor 2
23	Antriebsadaptionfaktor 3
24	Antriebsadaptionfaktor 4

Antriebsadaptionfaktor 1 ... 4

Es können maximal 4 Antriebsgrößen aus Parameter p2782[0...3] "Modus Adaption" adaptiert werden:

- Kp Drehzahlregler P-Verstärkung Adaptiondrehzahl
- Tn Drehzahlregler Nachstellzeit Adaptiondrehzahl
- Stromsollwertfilter 1 Nenner und Zähler
- Stromsollwertfilter 2 Nenner und Zähler
- Drehzahlsollwertfilter 1 Nenner und Zähler
- Drehzahlregler Referenzmodell Eigenfrequenz und Dämpfung
- APC / APC eco Tv (p3769)
- APC / APC eco Filter (p3766)

Hinweis

Die Adaption von Antriebsgrößen setzt die Nutzung von Telegramm 146, 148 oder 149 voraus.

Literatur:

- Informationen zur Adaption von Antriebsgrößen siehe "SINAMICS S120 Funktionshandbuch Antriebsfunktionen".
- Informationen bezüglich Telegramm 146/148/149 siehe "SINAMICS S120 Funktionshandbuch Kommunikation".

Maschinenachsnummer der Eingangsgröße

Die Zuordnung der Eingangsgröße einer Adaption zu einer Maschinenachse erfolgt über das Maschinendatum:

MD16504 \$MN_CADAPT_INPUT_AX[<n>]

mit <n> = Nummer der Adaption (0 ... 99)

Wert	Bedeutung
0	Keine Maschinenachse zugeordnet, Adaption ist inaktiv (Standardeinstellung)
1 ... 31	Nummer der zugeordneten Maschinenachse

Maschinenachsnummer der Ausgangsgröße

Die Zuordnung der Ausgangsgröße einer Adaption zu einer Maschinenachse erfolgt über das Maschinendatum:

MD16505 \$MN_CADAPT_OUTPUT_AX[<n>]

mit <n> = Nummer der Adaption (0 ... 99)

Wert	Bedeutung
0	Keine Maschinenachse zugeordnet, Adaption ist inaktiv (Standardeinstellung)
1 ... 31	Nummer der zugeordneten Maschinenachse

Adaptionskennlinie

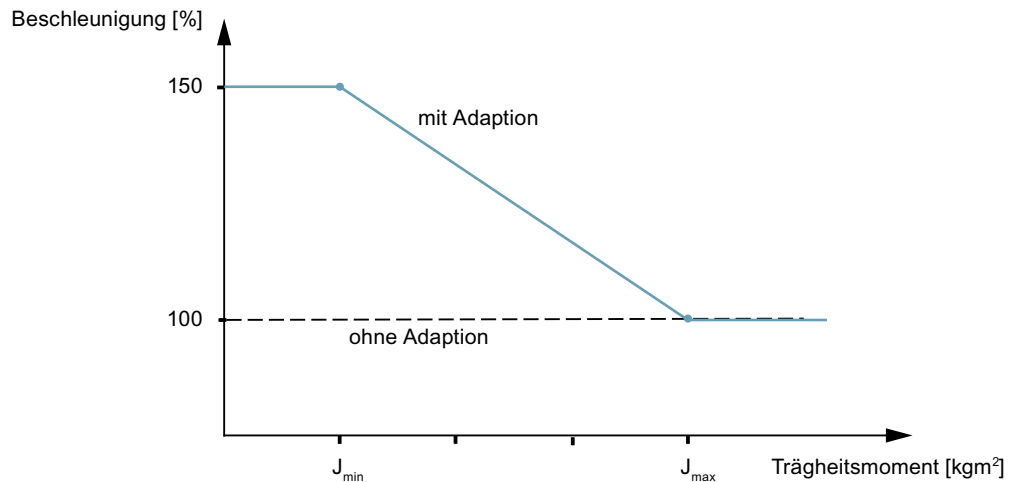
Die Kennlinie einer Lastadaption wird durch Stützpunkte definiert, die sich aus der Vorgabe fester Trägheitswerte und den zugehörigen Skalierungsfaktoren (0% ... 3000%) für die Ausgangsgröße ergeben. Werte zwischen den Stützpunkten werden linear interpoliert.

Für eine einfache Adaption sind zwei Stützpunkte ausreichend:

- MD16506 \$MN_CADAPT_INPUT_VALUE_1[<n>] (Eingangswert 1 der Adaption)
- MD16507 \$MN_CADAPT_INPUT_VALUE_2[<n>] (Eingangswert 2 der Adaption)
- MD16508 \$MN_CADAPT_OUTPUT_VALUE_1[<n>] (Ausgangswert 1 der Adaption)
- MD16509 \$MN_CADAPT_OUTPUT_VALUE_2[<n>] (Ausgangswert 2 der Adaption)

15.3 Beispiel

Für einen Rundtisch mit stark unterschiedlicher Beladung soll die wirksame Beschleunigung an die aktuelle Trägheit adaptiert werden. Mit der Funktion "Intelligente Lastanpassung" kann die Beschleunigung bei minimaler Beladung auf 150 % erhöht werden.



J_{\min} ⇒ MD16506[0] \$MN_CADAPT_INPUT_VALUE_1

J_{\max} ⇒ MD16507[0] \$MN_CADAPT_INPUT_VALUE_2

150 % ⇒ MD16508[0] \$MN_CADAPT_OUTPUT_VALUE_1

100 % ⇒ MD16509[0] \$MN_CADAPT_OUTPUT_VALUE_2

TE01: Installation und Aktivierung ladbarer Compile-Zyklen

16

Inhalt

In den nachfolgenden Kapiteln wird beschrieben, wie Technologie- und Sonderfunktionen in Form einzeln ladbarer Compile-Zyklendateien (*.ELF) in der Steuerung installiert und aktiviert werden.

Von Siemens in Form von Compile-Zyklen verfügbare Technologie-Funktionen

- **Abstandsregelung 1D/3D im Lagereglertakt**
Compile-Zyklus: CCCLC.ELF
Literatur: Funktionshandbuch Sonderfunktionen; Abstandsregelung (TE1)
- **Transformationspaket Handling**
Compile-Zyklus: CCRCTRA.ELF
Literatur: Funktionshandbuch Sonderfunktionen; Transformationspaket Handling (TE4)
- **Sollwertumschaltung**
Compile-Zyklus: CCSETP.ELF
Literatur: Funktionshandbuch Sonderfunktionen; Sollwertumschaltung (S9)
- **Axiale Kopplung im Maschinenkoordinatensystem (MCS-Kopplung)**
Compile-Zyklus: CCMCSC.ELF
Literatur: Funktionshandbuch Sonderfunktionen; MKS-Kopplung (TE6)
- **Wiederaufsetzen auf die Kontur (Retrace-Support)**
Compile-Zyklus: CCRESU
Literatur: Funktionshandbuch Sonderfunktionen; Wiederaufsetzen - Retrace Support (TE7)
- **Schnelle Laserschaltsignal**
Compile-Zyklus: CCHSLC.ELF
Literatur: Funktionshandbuch Sonderfunktionen; Taktunabhängige bahnsynchrone Schaltsignalausgabe (TE8)
- **Achspaar Kollisionsschutz**
Compile-Zyklus: CCPROT.ELF
Literatur: Funktionshandbuch Sonderfunktionen; Achspaar Kollisionsschutz (TE9)

Compile-Zyklendateien (*.ELF) und CNC-Software Versionen

Compile-Zyklendateien (*.ELF), die für CNC-Software Versionen bis SW 4.4 erzeugt wurden, sind auf Systemen mit CNC-Software Versionen ab SW 4.5 nicht ablauffähig.

Von Siemens in Form von Compile-Zyklen bereitgestellte Technologie- und Sonderfunktionen

Neue Compile-Zyklendateien (*.ELF) für Siemens Technologie- und Sonderfunktionen für CNC-Software Versionen ab SW 4.5 sind über Ihren regionalen Siemens-Vertriebspartner zu beziehen.

Von Drittanbietern in Form von Compile-Zyklen bereitgestellte Technologie- und Sonderfunktionen

Compile-Zyklendateien (*.ELF), die für CNC-Software Versionen bis SW 4.4 erzeugt wurden, müssen angepasst und mit der Generierumgebung für CNC-Software Versionen ab SW 4.5 neu übersetzt werden. Einzelheiten zum Anpassen der anwenderspezifischen Technologie- und Sonderfunktionen sind der Open Architecture (OA) Dokumentation zu entnehmen.

Compile-Zyklen

Compile-Zyklen sind funktionale Erweiterungen der NC-Systemsoftware, die vom Maschinenhersteller und/oder von Siemens erstellt und nachträglich in die Steuerung eingebracht werden können.

Im Rahmen der offenen NC-Systemarchitektur haben Compile-Zyklen über definierte Software-Schnittstellen umfangreichen Zugriff auf Daten und Funktionen der NC-Systemebene. Somit lässt sich über Compile-Zyklen die Funktionalität des NC umfangreich erweitern.

Das Einbringen eines Compile-Zyklus in die NC-Systemsoftware erfolgt durch Laden des Compile-Zyklus in das Filesystem des NC. Das Laden des Compile-Zyklus kann zu einem beliebigen Zeitpunkt erfolgen.

Siemens-Compile-Zyklen

Siemens-Compile-Zyklen sind von Siemens zur Verfügung gestellte Technologie-Funktionen.

Mit der Bestellung einer dieser Technologie-Funktionen erhalten Sie nur die entsprechende Software-Lizenznummer. Um den Compile-Zyklus selbst in Form einer ladbaren Datei (Erweiterung ".ELF" für "executable and linking format") zu erhalten, wenden Sie sich bitte an Ihren regionalen Siemens-Vertriebspartner.

Hinweis

Von Siemens erstellte Compile-Zyklen sind Optionen, die jeweils explizit aktiviert und lizenziert werden müssen.

Literatur:

Bestellunterlage Katalog NC 60/61

16.1 Laden von Compile-Zyklen

16.1.1 Laden eines Compile-Zyklus mit SINUMERIK Operate

Voraussetzung

- Der Compile-Zyklus, der auf die Steuerung übertragen werden soll, muss auf einem Speichermedium vorliegen, das direkt an die Steuerung angeschlossen werden kann, z. B. USB-FlashDrive.
- Das Zugriffsrecht der Schutzstufe 1 "Maschinenhersteller" (Default-Passwort "SUNRISE") muss vorliegen.
- Sind an der Bedientafel keine Cursor-Tasten vorhanden, muss zur Durchführung der Bedienhandlungen eine Maus oder Tastatur angeschlossen werden.

Durchführung

Führen Sie zum Laden eines Compile-Zyklus von z. B. einem USB-FlashDrive folgende Handlungsschritte aus:

1. Stecken Sie das USB-FlashDrive in die USB-Schnittstelle an der Frontseite der Bedientafel.
2. Wechseln Sie an der Bedienoberfläche in den Bereich "Systemdaten":
Horizontale Softkeyleiste: "Bedienbereichsumschaltung" > "Inbetriebnahme" > "Systemdaten"
3. Öffnen Sie das USB-FlashDrive: Verzeichnis "USB" (Tastatur: "Eingabe"-Taste).
4. Wählen Sie den Compile-Zyklus an und kopieren Sie ihn:
Vertikale Softkeyleiste: "Kopieren"
5. Wechseln Sie innerhalb des Bereichs "Systemdaten" in das Verzeichnis "NC-Daten" > "Compile-Zyklen" und fügen sie dort den Compile-Zyklus ein:
Vertikale Softkeyleiste: "Einfügen"
6. Lösen Sie zum Wirksamwerden des Compile-Zyklus einen NC-Reset aus:
Horizontale Softkeyleiste: "Bedienbereichsumschaltung" > "Inbetriebnahme" > "Maschinendaten"
Vertikale Softkeyleiste: "Reset (po)"

16.1.2 Laden eines Compile-Zyklus mit HMI Advanced

Voraussetzung

Zur Übertragung eines Compile-Zyklus auf die Steuerung muss folgende Voraussetzung erfüllt sein:

An der PCU ist ein Speichermedium (z. B. USB-FlashDrive) angeschlossen, auf dem sich der Compile-Zyklus befindet.

Durchführung

Führen Sie folgende Handlungsschritte zum Laden eines Compile-Zyklus von einem USB-FlashDrive in die NC aus:

1. Stecken Sie das USB-FlashDrive in die PCU 50 / 70.
2. Öffnen Sie das USB-FlashDrive als lokales Laufwerk:
Bedienbereichsumschaltung > Dienste > Daten verwalten > USB lokal
Wird der Softkey "USB lokal" gegraut dargestellt, wurde das USB-FlashDrive noch nicht erkannt. Warten Sie, bis HMI Advanced das USB-FlashDrive erkannt hat.
3. Wählen Sie den Compile-Zyklus an und kopieren Sie ihn:
vertikaler Softkey "Kopieren"
4. Wechseln Sie in das OEM-Verzeichnis der NC-Card:
ETC-Taste (">") > "NC-Card" > Verzeichnis: "cc" oder "Ladbare-Compile-Zyklen"
5. Fügen Sie den Compile-Zyklus in das OEM-Verzeichnis der NC-Card ein:
vertikaler Softkey "Einfügen"
6. Lösen Sie NC-Reset aus.

16.1.3 Laden eines Compile-Zyklus von einem externen Rechner mit WinSCP3

Voraussetzung

Zur Übertragung eines Compile-Zyklus auf die Steuerung müssen folgende Voraussetzungen erfüllt sein:

- Der externe Rechner (PG / PC), auf dem sich der Compile-Zyklus befindet, ist über eine Netzwerkverbindung (TCP / IP) mit der PCU verbunden.
- Das Programm "WinSCP3" ist auf dem externen Rechner installiert.
- Der Hostname oder die IP-Adresse der PCU sind bekannt.
- Benutzername und Passwort für das Login auf der PCU sind bekannt.

Durchführung

Führen Sie folgende Handlungsschritte zum Laden eines Compile-Zyklus von einem externen Rechner in den NC aus:

1. Starten Sie auf dem externen Rechner (PG / PC) das Programm "WinSCP3".
2. Verbinden Sie sich über Auswahl eines entsprechenden Profils oder über Angabe von Hostname oder IP-Adresse, Benutzername und Passwort mit der PCU.
3. Kopieren Sie den Compile-Zyklus vom externen Rechner in folgendes Verzeichnis der PCU:
/oem/sinumerik/data/cc
Der Dateiname des Compile-Zyklus darf im Verzeichnis der PCU keine Großbuchstaben enthalten.
4. Lösen Sie einen NC-Reset aus.

16.2 Kompatibilität der Interfaceversionen

Die Kommunikation zwischen Compile-Zyklus und NC-Systemsoftware erfolgt über ein SINUMERIK-spezifisches Interface. Die Interface-Version eines geladenen Compile-Zyklus muss kompatibel zur Interface-Version der NC-Systemsoftware sein.

Interface-Versionen

Die jeweiligen Interface-Versionen werden angezeigt unter:

- Interfaceversion der NC-Systemsoftware
HMI Advanced:
Diagnose > Serviceanzeige > Version > NCU Version
Anzeige (Ausschnitt):

CC Interface Version:
@NCKOPI@Interfaces=<1. Stelle>.<2. Stelle>
Loaded Compile Cycles:
. . . .

- Interfaceversion eines noch nicht geladenen Compile-Zyklus
HMI Advanced (Ausschnitt):
Dienste > <Medium> > Softkey: "Eigenschaften"
Anzeige:

Inhalt: Ladbarer-Compile-Zyklus
Interface:@Interfaces=<1. Stelle>.<2. Stelle>

- Interfaceversion eines geladenen Compile-Zyklus
HMI Advanced:
Diagnose > Serviceanzeige > Version > NCU Version
Anzeige (Ausschnitt):

CC Interface Version:
@NCKOPI
Loaded Compile Cycles:
<Bezeichner> <Version> <Generierdatum>
CC start address
N<Bezeichner><Version>IF<1. Stelle><2. Stelle>_ELF . . .

Beispiel:
_N_CLC407IF003001_ELF entspricht Interfaceversion: 3.1

Abhängigkeiten

Zwischen den Interface-Versionen eines Compile-Zyklus und der NC-Systemsoftware bestehen folgende Abhängigkeiten:

- 1. Stelle der Interface-Versionsnummer
Die 1. Stelle der Interface-Versionsnummer eines Compile-Zyklus und der NC-Systemsoftware müssen **gleich** sein.
- 2. Stelle der Interface-Versionsnummer
Die 2. Stelle der Interface-Versionsnummer eines Compile-Zyklus muss **kleiner gleich** der 2. Stelle der NC-Systemsoftware sein.

Hinweis

Wenn nach dem Hochlauf der NC ein Alarm 7200 angezeigt wird, wurde **kein** Compile-Zyklus geladen!

16.3 Software-Version eines Compile-Zyklus

Die SW-Version eines Compile-Zyklus wird angezeigt unter:

HMI Advanced:

Diagnose > Serviceanzeige > Version > NCU Version

Anzeige (Ausschnitt):

```
-----  
CC Interface Version:  
@NCKOPI . . . .  
Loaded Compile Cycles:  
<Bezeichner> <Version> <Generierdatum>  
CC start address  
_N_<Bezeichner><Version>IF<1. Stelle>.<2. Stelle>_ELF . . .  
Code=<Adresse> Data=<Adresse> . . .  
-----
```

Beispiel:

_N_CLC**407**IF003001_ELF entspricht SW-Version 4.7

Hinweis

Die Anzeige der Startadressen von Code- und Datenbereich eines Compile-Zyklus erfolgt zu Diagnosezwecken und ist im Normalbetrieb ohne Bedeutung.

16.4 Aktivieren der Technologiefunktionen im NC

Voraussetzung

Vor dem im weiteren Verlauf beschriebenen Aktivieren einer Technologie-Funktion ist die entsprechende Option zu setzen.

Ist das Optionsdatum nicht gesetzt, wird nach jedem NC-Hochlauf folgender Alarm angezeigt und die Technologie-Funktion wird nicht aktiviert:

Alarm 7202 "XXX_ELF_option_bit_missing: <Bitnummer>"

Funktionsspezifisches Maschinendatum

Jede per Compile-Zyklus geladene Technologie-Funktion legt ein funktionsspezifisches globales NC-Maschinendatum nach folgendem Schema im Nummernbereich 60900 bis 60999 an:

`$MN_CC_ACTIVE_IN_CHAN_<Bezeichner>[n]`, mit $n = 0, 1$

Beispiel:

`$MN_CC_ACTIVE_IN_CHAN_MCSC[0]`

`$MN_CC_ACTIVE_IN_CHAN_MCSC[1]`


Aktivieren für den 1. NC-Kanal

Das Aktivieren der Technologie-Funktionen im ersten NC-Kanal erfolgt über:

`$MN_CC_ACTIVE_IN_CHAN_<Bezeichner>[0]`, Bit0 = 1

Die Bedeutung der Maschinendatenbits Bit1 - Bit31 ist der jeweiligen Funktionsbeschreibung (TE1- TEn) zu entnehmen.

Nach dem nächsten NC-Hochlauf werden die aktivierten Technologie-Funktionen in die Systemsoftware eingebunden.

 VORSICHT
SINUMERIK 840D sl Mit dem erstmaligen Setzen eines Bits in einem der funktionsspezifischen NC-Maschinendaten: <code>\$MN_CC_ACTIVE_IN_CHAN_XXXX[0]</code> wird folgender Alarm angezeigt: Alarm 4400 "MD-Änderung bewirkt Reorganisation des gepufferten Speichers (Datenverlust)" Der Alarm kündigt ein Löschen aller Anwenderdaten (Teileprogramme, Werkzeugdaten, etc.) beim nächsten Hochlauf an. Gegebenenfalls muss nach dem Setzen des Datums und vor dem Auslösen von NC-RESET ein Archiv erstellt werden.

16.5 Funktionsspezifische Inbetriebnahme

Die weitere funktionsspezifische Inbetriebnahme ist der entsprechenden Funktionsbeschreibung (TE1-TE_n) zu entnehmen.

16.6 Anlegen von Alarmtexten

Die Alarmtexte für die Technologie-Funktionen sind im System hinterlegt. Sollte ein neuer Alarm notwendig sein, können die Alarmtexte entsprechend ergänzt werden. Das genaue Vorgehen ist abhängig von der vorhandenen Bedienoberfläche.

16.6.1 Alarmtexte anlegen mit SINUMERIK Operate

Die Alarmtexte der Technologie-Funktionen sollen um den folgenden Alarm ergänzt werden:
075999 0 0 "Kanal %1 Satz %2 Aufrufparameter ist ungültig"

Vorgehensweise

1. Kopieren Sie die Datei "oem_alarms_deu.ts" aus dem Verzeichnis "/siemens/sinumerik/hmi/template/lng" in das Verzeichnis "/oem/sinumerik/hmi/lng".
2. Benennen Sie die Datei um ("xxx_deu.ts").
3. Öffnen Sie die Datei im Editor und fügen Sie die neue Alarmnummer und den neuen deutschen Alarmtext ein:

```
<message>
  <source>075999/NCK</source>
  <translation>
    Kanal %1 Satz %2 Aufrufparameter ist ungültig
  </translation>
</message>
```

Hinweis

Jeder Alarm beginnt mit dem Tag `<message>` und endet mit dem Tag `</message>`. Das Tag `<source>` enthält die Alarmnummer und das Source-URL. Das Tag `<translation>` enthält den Alarmtext.

4. Zum Erstellen einer fremdsprachigen Alarmtextdatei kopieren Sie die gerade geänderte Datei und verändern im Dateinamen das Sprachkürzel (z. B. "xxx_eng.ts für Englisch").
5. Öffnen Sie die fremdsprachige Alarmtextdatei im Editor und tragen Sie den übersetzten Alarmtext im Tag `<translation>` ein.
6. Kopieren Sie die Muster-Konfigurationsdatei "oem_slaesvcadapconf.xml" aus dem Verzeichnis "/siemens/sinumerik/hmi/template/cfg" in das Verzeichnis "/oem/sinumerik/hmi/cfg".
7. Benennen Sie die Datei in "slaesvcadapconf.xml" um.

8. Öffnen Sie die Datei "slaesvcadapconf.xml" im Editor und tragen Sie den neuen Basenamen (Dateiname der neu erstellten Alarmtextdateien ohne Sprachkürzel und Postfix) ein, z. B.:

```
<BaseNames>
  <BaseName_02 type="QString" value="xxx"/>
</BaseNames>
```

9. Starten Sie SINUMERIK Operate neu.

Weiterführende Informationen zum Anlegen von Alarmtexten mit SINUMERIK Operate finden Sie in:

Literatur:

SINUMERIK 840D sl Basesoftware und Bedien-Software Inbetriebnahmehandbuch; Inbetriebnahmehandbuch SINUMERIK Operate (IM9), Kapitel: Alarmer/Maschinendaten konfigurieren

16.6.2 Alarmtexte anlegen mit HMI Advanced

Die Alarmtexte der Technologie-Funktionen sollen um den folgenden Alarm ergänzt werden:

075999 0 0 "Kanal %1 Satz %2 Aufrufparameter ist ungültig"

Vorgehensweise

1. Kopieren Sie die Datei "mbdde.ini" aus dem Verzeichnis "F:\mmc2" in das Verzeichnis "F:\oem".
2. Fügen Sie in der Datei "mbdde.ini" die beiden folgenden Zeilen ein:
[TextFiles]
UserCZYK=F:\oem\alc_
3. Legen Sie im Verzeichnis "F:\oem" die sprachabhängigen Textdateien "alc_XX.com" an, z. B.:
"alc_GR.com" für Deutsch
"alc_UK.com" für Englisch
4. Fügen Sie in den sprachabhängigen Textdateien den neuen Alarmtext ein, z. B.:
075999 0 0 "Kanal %1 Satz %2 Aufrufparameter ist ungültig"
in die deutsche Textdatei.
5. Starten Sie HMI Advanced neu.

Weiterführende Informationen zum Anlegen von Alarmtexten mit HMI Advanced finden Sie in:

Literatur:

SINUMERIK 840Di sl/840D sl/840D Basesoftware und HMI-Advanced Inbetriebnahmehandbuch; Inbetriebnahmehandbuch HMI-Advanced (IM4), Kapitel: Anwender-Alarmer projektieren

Hinweis

HMI-Neuinstallation

Ergänzte Alarmtexte in den Textdateien im Verzeichnis F:\oem bleiben auch nach einer Neuinstallation des HMI erhalten.

16.7 Hochrüsten eines Compile-Zyklus

Zum Hochrüsten eines in der Steuerung installierten Compile-Zyklus ist es **keinesfalls** ausreichend nur die entsprechende ELF-Datei auszutauschen. Wird nur die ELF-Datei ausgetauscht, kann es zu einem undefinierten Verhalten der NC-Software aufgrund inkonsistenter Daten der Speicher- und Datenverwaltung kommen.

Durchführung

Führen Sie zum Hochrüsten eines Compile-Zyklus folgende Handlungsschritte aus:

1. Erstellen Sie ein NC-Archiv **ohne** Compile-Zyklus:
Bedienbereich: "Inbetriebnahme" > "ETC"-Taste > "IBN-Archive" > "Inbetriebnahmearchiv erstellen" > "OK" >
 - Auswahl: "NC-Daten" angewählt
 - Auswahl: "mit Compile-Zyklen" nicht angewählt!
2. Löschen Sie den alten Compile-Zyklus:
Bedienbereich: "Inbetriebnahme" > "Systemdaten" im Verzeichnis: NC-Daten > "Compile-Zyklen" > Datei: <Bezeichner>.ELF
3. Laden Sie den neuen Compile-Zyklus (siehe Kapitel "Laden von Compile-Zyklen (Seite 691)")
4. Lösen Sie einen Power On-Reset mit NC-Urlöschen (NC-Inbetriebnahmeschalter: Stellung "1") aus. Siehe dazu:
Literatur:
Inbetriebnahmehandbuch IBN CNC: NC, PLC, Antrieb; Kapitel: Allgemeine Tipps > Getrenntes NC- und PLC-Urlöschen > NC-Urlöschen
5. Laden Sie das unter Punkt 1. erstellte NC-Archiv wieder in die NCU.
Bedienbereich: "Inbetriebnahme" > "ETC"-Taste > "IBN-Archive" > "Inbetriebnahmearchiv einlesen" > "OK"

Hinweis

Versionskontrolle

Zur Kontrolle der Version des neu geladenen Compile-Zyklus (siehe Kapitel "Software-Version eines Compile-Zyklus (Seite 694)").

Mehrere geladen Compile-Zyklen

Sind mehrere Compile-Zyklen in der Steuerung geladen, bleiben beim Hochrüsten eines Compile-Zyklus nach dem oben beschriebenen Vorgehen, die anderen Compile-Zyklen unverändert erhalten.

Neu angelegte Daten

Werden durch den neuen Compile-Zyklus neue Daten angelegt, die in der Vorgängerversion noch nicht vorhanden waren, sind diese nach dem abschließenden Hochlauf der Steuerung mit Default-Werten vorbesetzt.

16.8 Löschen eines Compile-Zyklus

Soll ein in der Steuerung geladener Compile-Zyklus vollständig gelöscht werden, ist es nicht ausreichend nur die entsprechende ELF-Datei zu löschen. Folgende Daten bleiben bei diesem Vorgehen im remanenten Speicher der Steuerung erhalten:

- Aktivierungsdaten: \$MN_CC_ACTIVE_IN_CHAN_<Bezeichner>[n]
- Funktionsspezifische Maschinendaten

Durchführung

Führen Sie zum Löschen eines Compile-Zyklus folgende Handlungsschritte aus:

1. Erstellen Sie ein NC-Archiv **ohne** Compile-Zyklen:
Bedienbereich: "Inbetriebnahme" > "ETC"-Taste > "IBN-Archive" > "Inbetriebnahmearchiv erstellen" > "OK" >
 - Auswahl: "NC-Daten" angewählt
 - Auswahl: "mit Compile-Zyklen" nicht angewählt!
2. Löschen Sie den Compile-Zyklus:
Bedienbereich: "Inbetriebnahme" > "Systemdaten" im Verzeichnis: NC-Daten > "Compile-Zyklen" > Datei: "<Bezeichner>.ELF" anwählen > "Löschen"
3. Lösen Sie einen Power On-Reset mit NC-Urlöschen (NC-Inbetriebnahmeschalter: Stellung "1") aus. Siehe dazu:
Literatur:
Inbetriebnahmehandbuch IBN CNC: NC, PLC, Antrieb; Kapitel: Allgemeine Tipps > Getrenntes NC- und PLC-Urlöschen > NC-Urlöschen
4. Laden Sie das unter Punkt 1. erstellte NC-Archiv wieder in die NCU.
Bedienbereich: "Inbetriebnahme" > "ETC"-Taste > "IBN-Archive" > "Inbetriebnahmearchiv einlesen" > "OK"

Hinweis

Mehrere geladen Compile-Zyklen

Sind mehrere Compile-Zyklen in der Steuerung geladen, bleiben beim Löschen eines Compile-Zyklus nach dem oben beschriebenen Vorgehen, die anderen Compile-Zyklen unverändert erhalten.

Effiziente Speichernutzung

Für eine möglichst effiziente Nutzung der Speicher-Ressourcen im NC, sollten nur die an der Maschine tatsächlich benötigten Compile-Zyklen (ELF-Dateien) geladen sein.

16.9 Datenlisten

16.9.1 Maschinendaten

16.9.1.1 NC-spezifische Maschinendaten

Nummer	Bezeichner: \$MN_	Beschreibung
60900 + i	CC_ACTIV_IN_CHAN_XXXX[n]	n = 0:
mit:	mit:	Aktivierung der Technologiefunktion in den NC-Kanälen
i = 0, 1, 2,	XXXX = Funktionskürzel	n = 1:
3, ...	n = 0 oder 1	Zusätzliche Funktionen innerhalb der Technologiefunktion

TE02: Simulation von Compile-Zyklen (nur HMI Advanced)

17

17.1 Kurzbeschreibung

17.1.1 Funktion

Werden auf der SINUMERIK-Bedienoberfläche "HMI Advanced" Teileprogramme die Compile-Zyklen verwenden simuliert, müssen abhängig vom verwendeten Compile-Zyklus, spezifische Umgebungsbedingungen hergestellt werden. Diese sind in den nachfolgenden Kapiteln beschrieben.

17.2 OEM-Transformationen

Bei Verwendung von OEM-Transformationen muss die Ablaufumgebung der Simulation eingestellt werden.

Vorgehensweise

1. Legen Sie auf dem Rechner, auf dem HMI Advanced installiert ist, neben dem standardmäßig vorhandenen Verzeichnis "<HMI Advanced Installationspfad>/MMC2" ein neues Verzeichnis "<HMI Advanced Installationspfad>/OEM" an.

2. Legen Sie im Verzeichnis "/OEM" die Datei "DPSIM.TEA" mit folgendem Inhalt an:

```
$MN_NC_USER_CODE_CONF_NAME_TAB[196]="TRAORI"  
$MN_NC_USER_CODE_CONF_NAME_TAB[197]="_TRAORI"  
$MN_NC_USER_CODE_CONF_NAME_TAB[198]="TRACON"  
$MN_NC_USER_CODE_CONF_NAME_TAB[199]="_TRACON"  
CHANDATA(1)  
$MC_AXCONF_GEOAX_ASSIGN_TAB[0]=1  
$MC_AXCONF_GEOAX_ASSIGN_TAB[1]=2  
$MC_AXCONF_GEOAX_ASSIGN_TAB[2]=3  
$MC_TRAFO_RESET_VALUE=0  
; Transformationstypen 4096 - 4101 unbedingt löschen  
$MC_TRAFO_TYPE_1=0  
$MC_TRAFO_TYPE_2=0  
$MC_TRAFO_TYPE_3=0  
; Transformationsverkettungen mit OEM-Transformationen löschen  
$MC_TRACON_CHAIN_1[0]=0  
$MC_TRACON_CHAIN_1[1]=0  
; ACHTUNG! Keine Leerzeichen nach M30  
M30
```

17.2 OEM-Transformationen

3. Legen Sie im Verzeichnis "/OEM" die Datei "DPSIM.INI" mit folgendem Inhalt an:
[PRELOAD]
CYCLES=1
CYCLEINTERFACE=0
4. Beenden Sie die HMI-Applikation.
5. Starten Sie die HMI-Applikation.
6. Legen Sie im Verzeichnis der Hersteller-Zyklen die Datei "TRAORI.SPF" mit folgendem Inhalt an:
PROC TRAORI (INT II)
RET
7. Legen Sie im Verzeichnis der Hersteller-Zyklen die Datei "TRACON.SPF" mit folgendem Inhalt an:
PROC TRACON (INT II)
RET

Hinweis

Die unter 6. und 7. erstellten Hersteller-Zyklen "TRAORI.SPF" und "TRACON.SPF" dürfen nicht in die NC geladen werden.

8. Starten Sie die Simulation.
9. Führen Sie nach dem Hochlauf der Simulation einen Datenabgleich bezüglich der Zyklen durch:
HMI Advanced: **Datenabgleich > Zyklen abgleichen**

Hinweis

Für den Datenabgleich wird mindestens das Passwort der Schutzstufe 3 "Endanwender: Service" benötigt.

TE1: Abstandsregelung - nur 840D sl

18.1 Kurzbeschreibung

18.1.1 Kurzbeschreibung

Funktion

Die Technologiefunktion "Abstandsregelung" dient zur Aufrechterhaltung eines technologisch erforderlichen ein- (1D) bzw. dreidimensionalen (3D) Abstandes innerhalb eines definierten Bearbeitungsprozesses. Der dabei aufrecht zu haltende Abstand ist z. B. die Entfernung eines Werkzeugs von der zu bearbeitenden Werkstückoberfläche.

Funktionskürzel

Das Kürzel der Technologiefunktion "Abstandsregelung" für funktionspezifische Bezeichner von Programmbefehlen, Maschinendaten etc. ist:

CLC (= Clearance Control)

Funktionsbeschränkung

Die Technologiefunktion "Abstandsregelung" ist auch bei Steuerungen mit mehr als einem NC-Kanal nur im ersten NC-Kanal anwendbar.

Daraus folgt:

- Als abstandsgeregelte Achsen dürfen nur Kanalachsen des ersten NC-Kanals verwendet werden.
- Die Verwendung von Teileprogrammbefehlen der Abstandsregelung ist nur in Teileprogrammen erlaubt, die im ersten NC-Kanal abgearbeitet werden.

Hinweis

Die Technologiefunktion "Abstandsregelung" ist nur im **ersten** NC-Kanal anwendbar!

Verfügbarkeit

Die Technologiefunktion "Abstandsregelung" ist nur für SINUMERIK 840D sl verfügbar.

Compile-Zyklus

Die Technologiefunktion "Abstandsregelung" ist ein Compile-Zyklus.

18.1 Kurzbeschreibung

Zur Beschreibung der systemspezifischen Verfügbarkeit und Handhabung von Compile-Zyklen (siehe Kapitel "TE01: Installation und Aktivierung ladbarer Compile-Zyklen (Seite 689)").

18.1.2 Funktionsbeschreibung

Die weitere Beschreibung der Funktionalität der Technologiefunktion "Abstandsregelung" erfolgt beispielhaft anhand der Technologie Laserschneiden.

Laserschneiden

Beim Laserschneiden wird ein aufgeweiteter, paralleler Laserstrahl über Lichtwellenleiter oder Spiegel auf eine im Laser-Bearbeitungskopf montierte Sammellinse geführt. Die Sammellinse fokussiert den Laserstrahl in ihrem Brennpunkt. Typische Brennweiten sind dabei ca. 5 bis 20 cm.

Die relative Lage des Brennpunkts zum Werkstück ist beim Laserschneiden ein extrem kritischer Prozess-Parameter, der innerhalb einer Toleranz von $\leq 100 \mu\text{m}$ konstant gehalten werden muss.

Der prozesstechnisch wichtige Abstand zwischen Brennpunkt und Werkstück wird üblicherweise mit Hilfe eines schnellen kapazitiven Abstandssensors gemessen. Die analoge Ausgangsspannung des Abstandssensors ist näherungsweise proportional zum Abstand des Sensors von der Werkstückoberfläche.

Über eine analoge Peripheriebaugruppe wird die Ausgangsspannung des Abstandssensors als digitaler Eingangswert in die Steuerung übertragen und erzeugt dort bei Abweichungen vom Sollabstand einen zusätzlichen Geschwindigkeitssollwert für die Bewegungsachsen des Bearbeitungskopfes.

Systemüberblick

Einen Überblick über die zur Abstandsregelung benötigten Systemkomponenten im Zusammenhang mit SINUMERIK 840D sl gibt folgendes Bild:

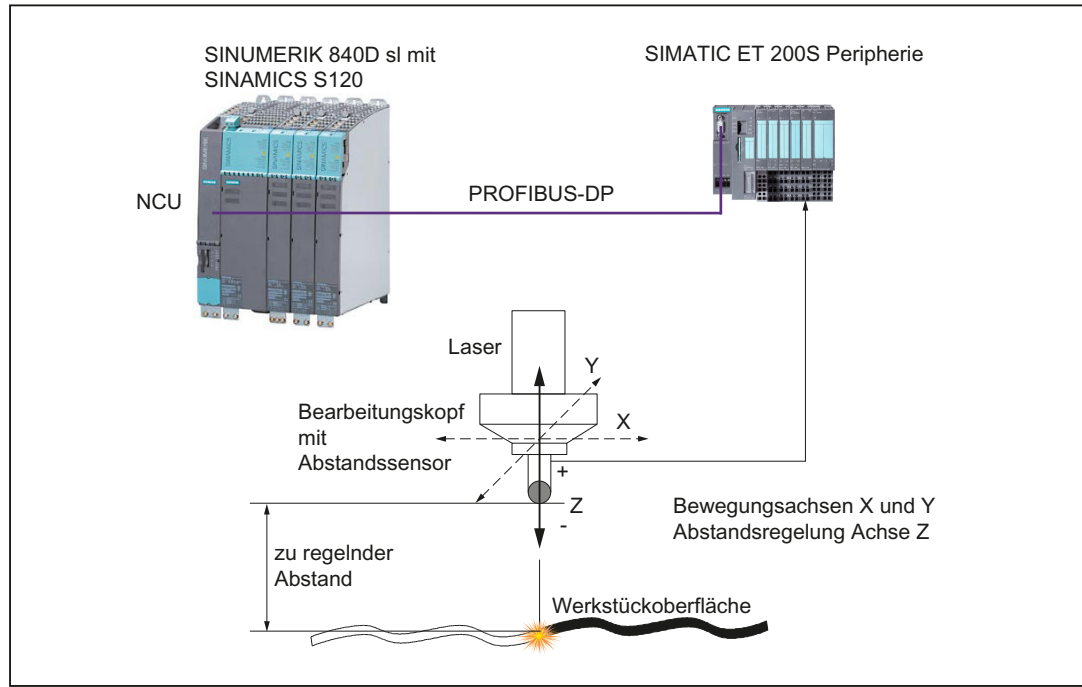


Bild 18-1 Systemkomponenten zur Abstandsregelung mit SINUMERIK 840D sl

1D- / 3D-Bearbeitungen

Die Abstandsregelung kann sowohl bei 1D- als auch bei 3D-Bearbeitungen mit bis zu fünf interpolierenden Achsen eingesetzt werden.

- 1D-Bearbeitung

Bei der 1D-Bearbeitung wird nur eine Achse durch die Abstandsregelung beeinflusst. Z. B. Achse Z, wie in dem unter Systemüberblick aufgezeigten Maschinen-Konfigurationsbeispiel (siehe voriges Bild). Die Abstandsregelung wirkt dabei nur in Richtung der Achse Z.
- 3D-Bearbeitung

Die Positionierung des Werkzeugs erfolgt durch drei Linearachsen. Die Orientierung des Werkzeugvektors durch eine oder zwei Rundachsen (5-Achs-Bearbeitung). Durch die Abstandsregelung werden bis zu drei Linearachsen beeinflusst. Die Ausgleichsbewegung ist wahlweise in Richtungen folgender Vektoren vorgebar:

 - Vektor der Werkzeugorientierung (Standardfall)
 - Beliebiger programmierbarer Vektor (Kompensationsvektor)

18.2 Abstandsregelung

18.2.1 Regeldynamik

Regelkreisverstärkung K_V

Die Dynamik des geschlossenen Regelkreises (Sensor-Steuerung-Achse) wird bestimmt von der max. einstellbaren Regelkreisverstärkung K_V .

Die Regelkreisverstärkung K_V ist definiert als:

$$K_V = \frac{\text{Geschwindigkeit [m/min]}}{\text{Schleppabstand [mm]}} ; \quad \text{in } \left[\frac{\text{[m/min]}}{\text{[mm]}} \right]$$

Kennlinien der Abstandsregelung

Die Abstandsregelung basiert auf den beiden im nachfolgenden Bild dargestellten Kennlinien:

- Kennlinie des Abstandssensors (Sensor-Eigenschaft)
- Kennlinie der Abstandsregelung (über Maschinendaten parametrierbar)

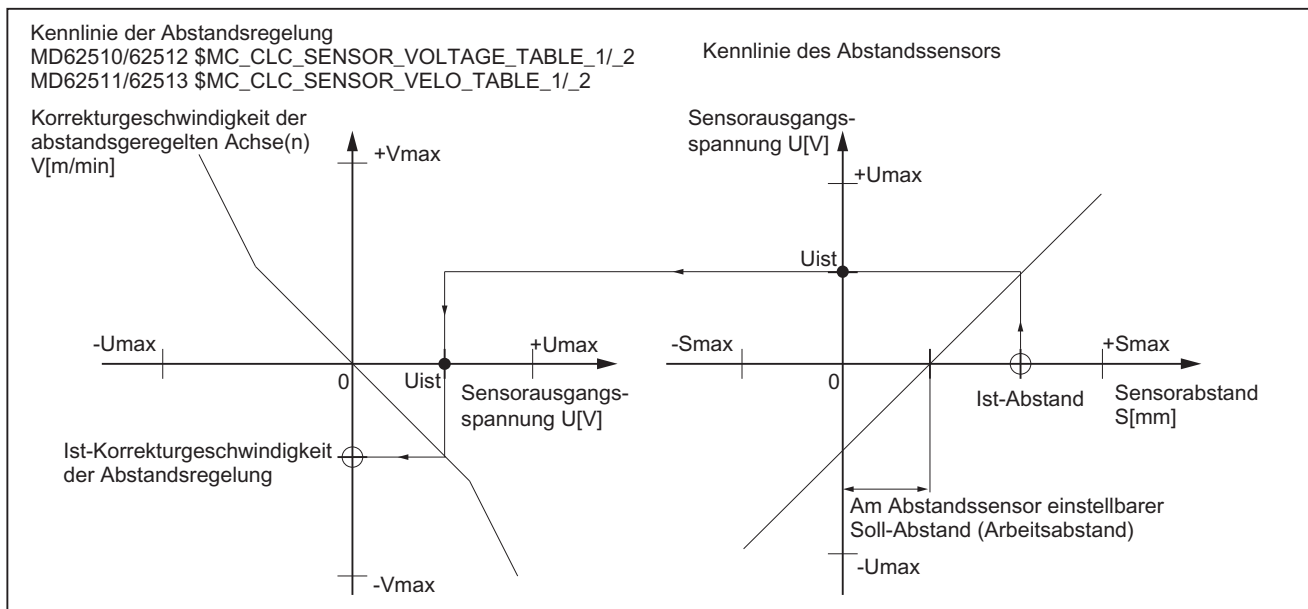


Bild 18-2 Zusammenhang der Kennlinien: Abstandssensor und Abstandsregelung

- Der Abstandssensor erfasst den Ist-Abstand zur Werkstückoberfläche und liefert als Ausgangsgröße eine dem Abstand annähernd direkt proportionale Spannung in Volt.
- Die Abstandsregelung ermittelt anhand der parametrierten Spannungs-/Geschwindigkeits-Kennlinie aus der vom Abstandssensor gelieferten Spannung die dem Abstand entsprechende Korrekturgeschwindigkeit für die abstandsgeregelten Achsen.

Aus Sicht der Steuerung hat die Regelkreisverstärkung die Einheit [(mm/min)/Volt]. Die Normierung auf [(mm/min)/mm] kann ebenso wie die Normierung des Sollabstandes in [mm] nur unter Einbeziehung der Sensor-Elektronik erfolgen.

Max. Regelkreisverstärkung

Die maximal erreichbare Regelkreisverstärkung wird von den folgenden Verzögerungs- und Reaktionszeiten des Gesamtsystems bestimmt:

1. Reaktionszeit des Sensors
2. Verzögerungszeit des A/D-Wandlers
3. Verzögerungs-/Totzeiten der Signalverarbeitung
4. Reaktionszeit des Lagereglers
5. Reaktionszeiten von Drehzahl- und Stromregler
6. Zeitkonstanten von Motor und Mechanik

In der Praxis relevant sind jedoch nur die Punkte 3 und 4.

In Summe ergeben die aufgeführten Einflussgrößen eine effektive Zeitkonstante. Eine, bezogen auf diese Zeitkonstante, zu groß eingestellte Regelkreisverstärkung führt zu Eigenschwingungen der zu regelnden Achse/Achsen im Bereich einiger Hertz.

Die Inbetriebnahme der Abstandregelung hat das Ziel, die entscheidenden Zeitkonstanten in der Weise zu minimieren, dass die prozesstechnisch erforderliche Regelkreisverstärkung ohne Auftreten von Eigenschwingungen eingestellt werden kann.

Totzeit

Um ein hochdynamisches Regelverhalten zu erreichen, findet die Abstandsregelung auf der höchstpriorären Lageregler-Ebene des NC statt.

Für SINUMERIK 840D sl mit über PROFIBUS-DP angeschlossenen Peripheriebaugruppen und Antrieben ergibt sich eine Totzeit T_{tot} von:

$$T_{\text{tot}} = 2 * \text{Lagereglertakt} + 2 * \text{Drehzahlreglertakt} + \text{Eingangsvorhaltezeit } T_i$$

18.2.2 Geschwindigkeitsvorsteuerung

Eliminieren der Verzögerungszeit

Die für den Lageregler eingestellte Regelkreisverstärkung K_v entspricht einer Verzögerungszeit Δt . Die Verzögerungszeit Δt ist dabei die Zeit, die vergeht bis die zu regelnde Achse bei einer Geschwindigkeitsvorgabe v mit ihrer Istposition der Sollposition gefolgt ist.

Mit einer Verzögerungszeit:

$$\Delta t = \frac{1}{K_v}$$

und einer Regelkreisverstärkung K_v in Sekunden:

$$\left[\frac{\text{m/min}}{\text{mm}} \right] = \left[\frac{1000 \text{ mm}/60 \text{ s}}{\text{mm}} \right] = 16,667 \left[\frac{1}{\text{s}} \right]$$

ergibt sich bei einer angenommenen Regelkreisverstärkung $K_v = 4$ die entsprechende Verzögerungszeit Δt zu:

$$\Delta t = \frac{1}{4 * 16,667 \left[\frac{1}{\text{s}} \right]} = 14,999 \text{ ms}$$

Optimierung des Regelverhaltens

Falls das Regelverhalten der Achse durch die Geschwindigkeitsvorsteuerung zu hart wird, kann das Regelverhalten mit folgenden achsspezifischen NC-Maschinendaten optimiert werden:

- MD32410 \$MA_AX_JERK_TIME (Zeitkonstante für den axialen Ruckfilter)
- MD32610 \$MA_VELO_FFW_WEIGHT (Vorsteuerfaktor der Drehzahlvorsteuerung)

Zusätzliche Dämpfungsmöglichkeiten bieten die Geschwindigkeitsfilter des Antriebs SINAMICS S120:

- Parameter 1414 ff. (Zeitkonstante Drehzahlsollwertfilter 1, 2)

Hinweis

Jede dämpfende Maßnahme ergibt einen zusätzlichen Beitrag zur Vergrößerung der Gesamtzeitkonstante des Regelkreises!

Literatur:

Die vollständige Beschreibung zur Geschwindigkeitsvorsteuerung findet sich in: Funktionshandbuch Erweiterungsfunktionen; K3: Kompensationen, Kapitel "Dynamische Vorsteuerung (Schleppfehler-Kompensation)"

18.2.3 Regelkreisstruktur

Die folgenden Bilder geben einen Überblick über die Einbettung der Abstandsregelung in die Regelkreisstruktur des NC-Lagereglers und den internen Aufbau der Abstandsregelung.

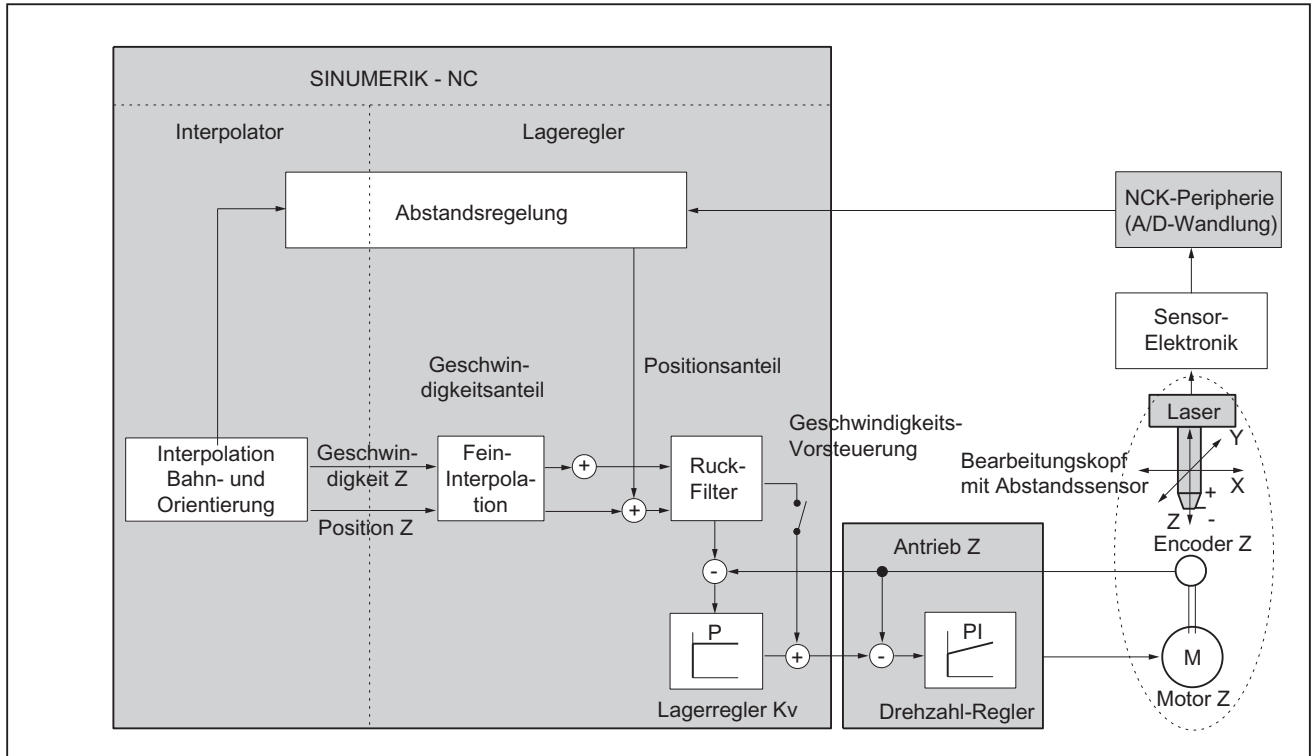


Bild 18-3 Regelstruktur Lagereglers mit Abstandsregelung (Prinzip)

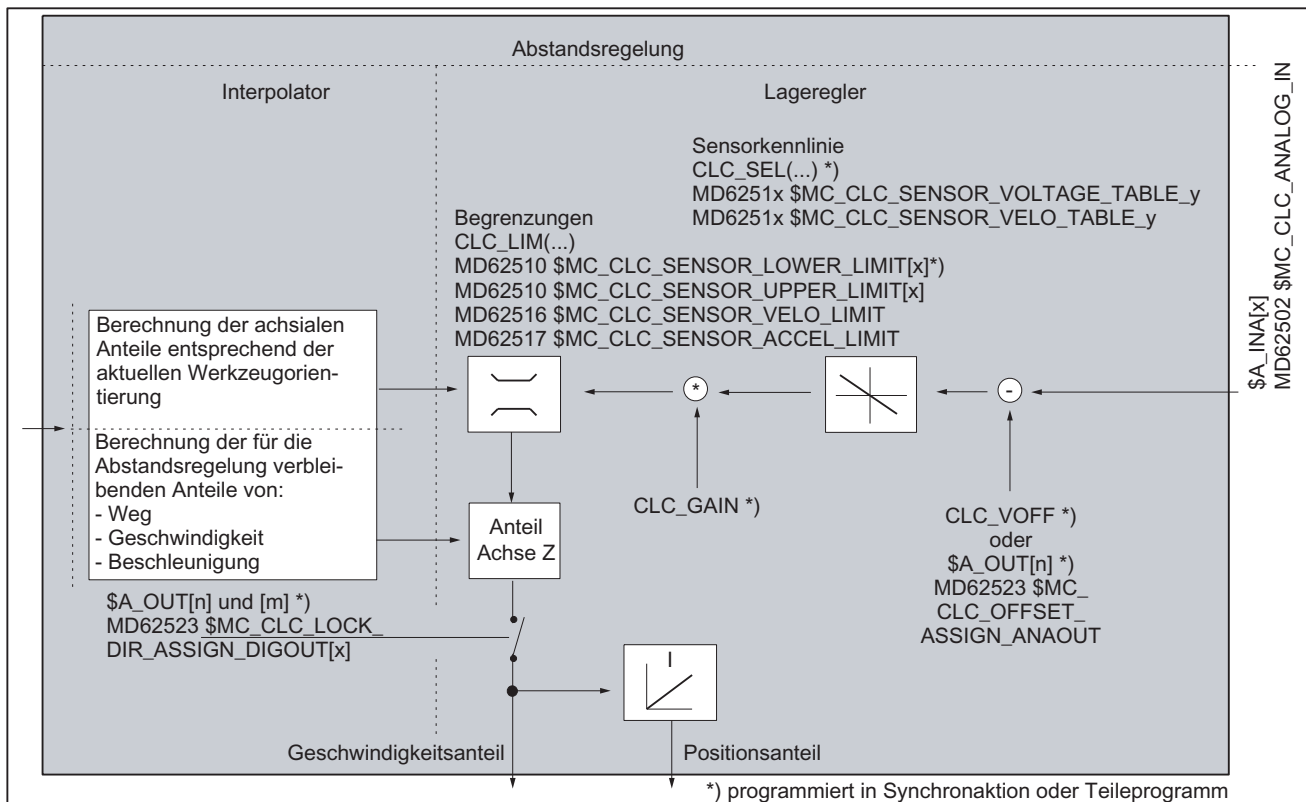


Bild 18-4 Regelstruktur Abstandsregelung (Prinzip)

18.2.4 Kompensationsvektor

Standard-Kompensationsvektor

Im Standardfall sind der Kompensationsvektor der Abstandsregelung und der Vektor der Werkzeugorientierung identisch. Demzufolge erfolgt die Ausgleichsbewegung der Abstandsregelung im Standardfall immer in Richtung der Werkzeugorientierung.

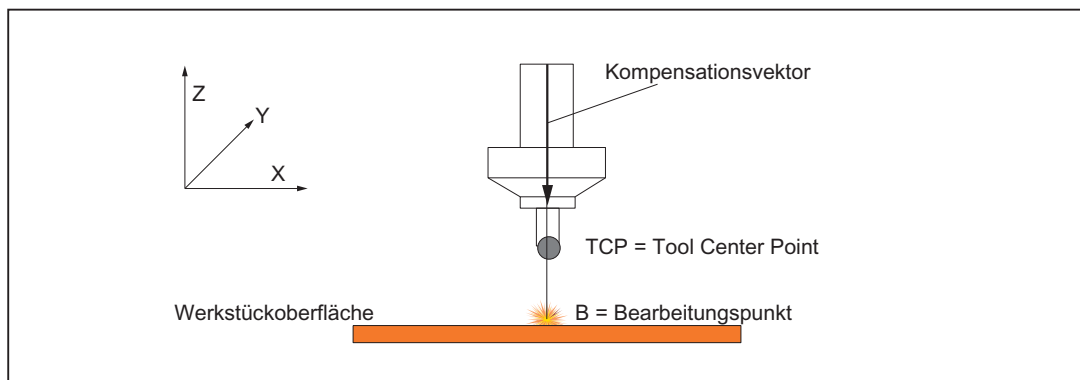


Bild 18-5 Abstandsregelung mit Standard-Kompensationsvektor

Hinweis

Die zur Bearbeitung des Werkstücks erforderliche Verfahrbewegung des Bearbeitungskopfes erfolgt in allen Bildern dieses Kapitels in Richtung der Y-Koordinate, d. h. senkrecht zur Zeichenebene.

Solange die Werkzeugorientierung und damit auch der Kompensationsvektor senkrecht zur Werkstückoberfläche ist, ergibt sich bei Ausgleichsbewegungen der Abstandsregelung kein nachteiliges Verhalten für den Bearbeitungsprozess.

Wird aus technologischen Gründen ein Werkzeuganstellwinkel erforderlich, sodass die Werkzeugorientierung nicht mehr senkrecht zur Werkstückoberfläche ist, ergibt sich bei Ausgleichsbewegungen der Abstandsregelung entlang des Standard-Kompensationsvektors ein Versatz des Bearbeitungspunktes auf der Werkstückoberfläche.

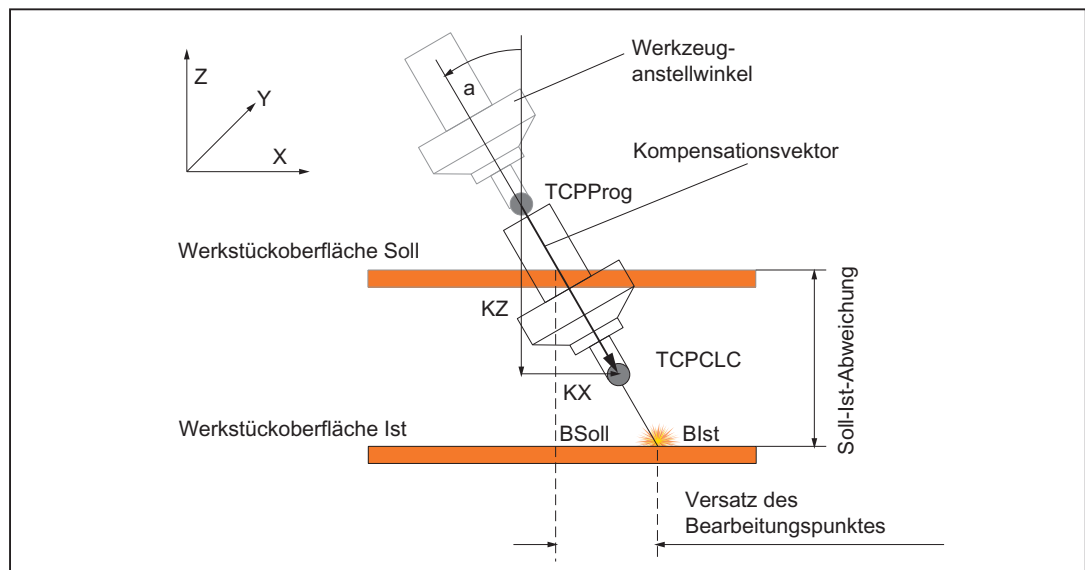


Bild 18-6 Standard-Kompensationsvektor

Die Ursache für den Versatz des Bearbeitungspunktes ist der X-Anteil (K_x) des Kompensationsvektors parallel zur Werkstückoberfläche. Um diesen Anteil verschiebt sich der TCP des Werkzeugs und damit der Bearbeitungspunkt B.

Programmierbarer Kompensationsvektor

Bei Verwendung des programmierbaren Kompensationsvektors erfolgen die Ausgleichsbewegungen der Abstandsregelung nicht in Richtung der Werkzeugorientierung, sondern in Richtung des programmierten Vektors.

Durch Vorgabe des programmierbaren Kompensationsvektors senkrecht zur Werkstückoberfläche, entfällt der oben genannte X-Anteil (K_x). Dadurch entsteht kein Versatz des Bearbeitungspunktes (B) durch die Ausgleichsbewegung der Abstandsregelung.

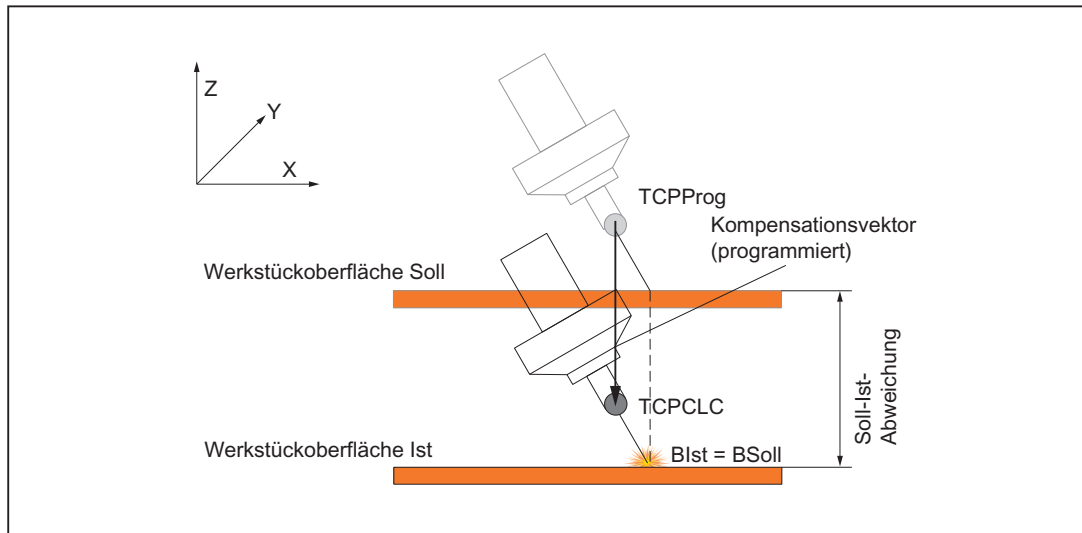


Bild 18-7 Programmierbarer Kompensationsvektor

Orientierungsänderungen

Entsprechend den oben gemachten Aussagen entsteht auch bei einer Orientierungsänderung des Bearbeitungskopfes mit aktiver Abstandsregelung ein unterschiedliches Verhalten.

Im folgenden Bild links der Standardfall (Kompensationsvektor == Vektor der Werkzeugorientierung), rechts mit programmiertem Kompensationsvektor.

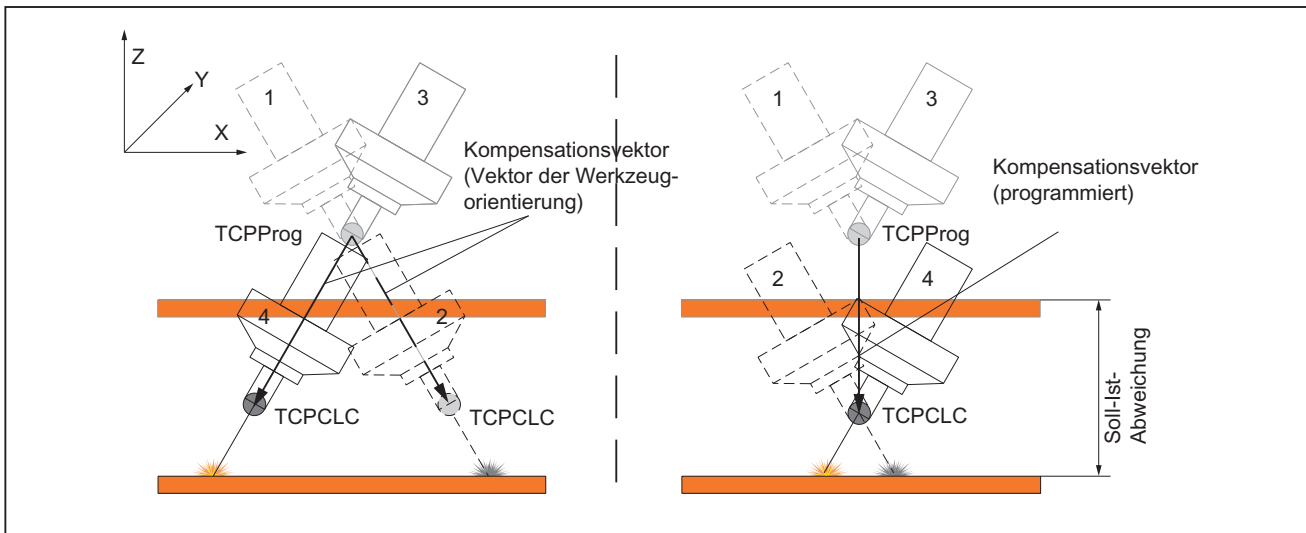


Bild 18-8 Orientierungsänderung des Bearbeitungskopfes

Die einzelnen Positionen des Bearbeitungskopfes bedeuten:

1. Programmierte Position des Bearbeitungskopfes
2. Tatsächliche Position des Bearbeitungskopfes mit aktiver Abstandsregelung vor der Orientierungsänderung

3. Programmierte Position des Bearbeitungskopfes nach der Orientierungsänderung
4. Tatsächliche Position des Bearbeitungskopfes mit aktiver Abstandsregelung nach der Orientierungsänderung

Die an der Maschine sichtbare Bewegung des Bearbeitungskopfes erfolgt bei der Orientierungsänderung direkt von Position 2 nach Position 4.

18.3 Technologische Eigenschaften der Abstandsregelung

Die Abstandsregelung ist durch folgende technologische Eigenschaften gekennzeichnet:

- **Dynamik**
Die überlagerte Sensor-Bewegung nutzt die aktuell von der programmierten Achsbewegung verbleibende Rest-Dynamik (Geschwindigkeit und Beschleunigung) aus. Der von der Rest-Beschleunigung zu nutzende Anteil kann als Prozentwert über ein Maschinendatum eingestellt werden.
- **Sensor-Kennlinie**
Die Verstärkungskennlinie eines Sensors kann mit bis zu 10 Stützpunkten vorgegeben werden.
- **Sensoren**
Es können zwei Sensoren mit unterschiedlichen Verstärkungskennlinien (z. B. ein mechanischer und ein kapazitiver Sensor) verwendet werden. Die aktive Sensorkennlinie ist durch einen Sprachbefehl im Teileprogramm Satz-synchron umschaltbar.
- **Regelkreisverstärkung der Abstandsregelung**
Die per NC-Maschinendaten parametrisierte Regelkreisverstärkung der Abstandsregelung ist durch einen Sprachbefehl im Teileprogramm Satz-synchron änderbar.
- **Bewegungsbegrenzung**
Die per NC-Maschinendaten parametrisierte untere und obere Begrenzung der durch die Abstandsregelung erzeugten Achsbewegungen ist durch einen Sprachbefehl im Teileprogramm Satz-synchron änderbar.
Das Erreichen einer Begrenzung wird durch einen Alarm angezeigt. Die Alarmreaktion (Stopp aller Verfahrbewegungen oder nur Anzeige) ist parametrierbar. Über ein PLC-Signal kann der aktuelle Positionsoffset eingefroren werden.
- **Ausschaltverhalten**
Das Ausschaltverhalten der Abstandsregelung kann wahlweise programmiert werden mit Synchronisation auf die aktuellen Achspositionen (keine Ausgleichsbewegung) oder mit Ausgleichsbewegung der Achsen auf die zuletzt programmierten Achspositionen (Achspositionen ohne Abstandsregelung).
- **Programmierbarer Sollabstand**
Um den an der Sensor-Elektronik eingestellten Soll-Abstand satzbezogen zu verändern, kann ein zusätzlicher Spannungswert programmiert werden.

- **Einflussmöglichkeiten über die PLC-Nahtstelle**
An der PLC-Nahtstelle sind folgende Signale verfügbar:
Zustandssignale:
 - Regelung aktiv
 - überlagerte Bewegung im Stillstand
 - untere Begrenzung erreicht
 - obere Begrenzung erreichtSteuersignale:
 - Bahnoverride für Sensorbewegung wirksam
- **Zustandsdaten der Abstandsregelung**
Sowohl die aktuellen als auch die Min/Max-Werte des Sensorsignals und des Positionsoffsets sind als GUD- und/oder BTSS-Variablen verfügbar.
- **Sensor-Signal**
Das Sensor-Signal kann über ein PT1-Filter mit einstellbarer Zeitkonstante geglättet werden.

18.4 Sensor-Kollisionsüberwachung

Sensor-Signal

Verfügt der verwendete Abstandssensor über ein zusätzliches Signal "Sensor-Kollision" zur Erkennung einer Kollision des Sensors mit dem bearbeitenden Werkstück, kann dieses Signal über einen digitalen NC-Peripherieeingang der Abstandsregelung zur Verfügung gestellt werden.

Als Reaktion auf dieses Signal führt die Abstandsregelung eine Rückzugsbewegung in allen abstandsgeregelten Achsen durch. Die Rückzugsbewegung wird dabei unabhängig vom aktuellen Wert des Geschwindigkeits-Overrides mit maximaler Verfahrgeschwindigkeit in positiver Regelrichtung bis zum Erreichen der aktuell gültigen oberen Begrenzung des Regelbereiches durchgeführt. Gleichzeitig erfolgt ein Stopp der Bahnbewegung.

Nach dem Stopp aller Verfahrbewegungen kann die Teileprogrammbearbeitung mit NC-START fortgesetzt werden.

Parametrierung

Der digitale Peripherieeingang, auf den das Signal "Sensor-Kollision" verdrahtet ist, wird der Abstandsregelung über folgendes Maschinendatum zugeordnet:

MD62504 \$MC_CLC_SENSOR_TOUCHED_INPUT (digitaler Peripherieeingang für Signal "Sensor-Kollision")

Die Angabe des digitalen Peripherieeingangs erfolgt dabei über die Eingangsnummer, analog der Angabe für die Systemvariablen der digitalen I/O-Peripherie \$A_IN/\$A_OUT mit \$A_IN[*Eingangsnummer*].

Eine negative Eingangsnummer bewirkt, dass das Signal "Sensor-Kollision" von der Abstandsregelung intern invertiert verarbeitet wird (Drahtbruchsicherung).

18.5 Inbetriebnahme

Compile-Zyklus

Vor Inbetriebnahme der Technologiefunktion ist sicherzustellen, dass der entsprechende Compile-Zyklus geladen und aktiviert ist (siehe Kapitel "TE01: Installation und Aktivierung ladbarer Compile-Zyklen (Seite 689)").

18.5.1 Aktivieren der Technologiefunktion

Die Technologiefunktion wird aktiviert über das Maschinendatum:

MD60940 \$MN_CC_ACTIVE_IN_CHAN_CLC[0], Bit n = 1

n = Kanal-Nummer - 1; Bit0 = 1. Kanal, Bit1 = 2. Kanal, usw.

Hinweis

Die Technologiefunktion kann für mehrere Kanäle gleichzeitig aktiviert werden.

18.5.2 Speicherkonfiguration durchführen

Die Technologiefunktion benötigt **zusätzliche** Daten im NC-internen Satzspeicher. Für folgende speicherkonfigurierende kanalspezifische Maschinendaten sind die Werte zu erhöhen:

- MD28090 \$MC_MM_NUM_CC_BLOCK_ELEMENTS += **4** (Anzahl Satzelemente für Compile-Zyklen)
- MD28100 \$MN_MM_NUM_CC_BLOCK_USER_MEM += **20** (Größe des Satzspeichers für Compile-Zyklen (DRAM) in kByte)

18.5.3 Parametrierung der Eingangssignale

Folgende Eingangssignale sind in den Maschinendaten zu parametrieren:

- Eingangsspannung des Abstandssensors
 - 1 analoger Eingang
- Eingangssignal "Sensor-Kollision (Optional)"
 - 1 digitaler Eingang

Analoger Eingang

Folgende Maschinendaten sind für den analogen Eingang zu parametrieren:

- MD10300 \$MN_FASTIO_ANA_NUM_INPUTS (Anzahl der aktiven analogen NC-Eingänge)
- MD10362 \$MN_HW_ASSIGN_ANA_FASTIN (pro Analog-Modul) (Hardwarezuordnung der schnellen analogen NC-Eingänge)
Die Spezifikation der physikalischen Adresse aktiviert das analoge Eingangsmodul

Digitaler Eingang

Folgende Maschinendaten sind für den digitalen Eingang zu parametrieren:

- MD10350 \$MN_FASTIO_DIG_NUM_INPUTS (Anzahl der aktiven digitalen NC-Eingangsbytes)
- MD10366 \$MN_HW_ASSIGN_DIG_FASTIN (pro Digital-Modul) (Hardwarezuordnung der externen digitalen NC-Eingänge)
Die Spezifikation der physikalischen Adresse aktiviert das digitale Eingangsmodul.

Die vollständige Beschreibung der analogen und digitalen Eingänge findet sich in:

Literatur:

Funktionshandbuch Erweiterungsfunktionen; Digitale und analoge NC-Peripherie (A4)

18.5.4 Parametrierung des programmierbaren Kompensationsvektors

Bezugs-Koordinatensystem

Über den programmierbaren Kompensationsvektor wird die Richtung vorgegeben, in der die Kompensationsbewegung der Abstandsregelung erfolgt. Der Kompensationsvektor bezieht sich immer auf das Basiskoordinatensystems (Maschinenkoordinatensystem).

Die Anfangskordinaten $[X_a, Y_a, Z_a]$ des Kompensationsvektors entsprechen dem Koordinatenursprung des Basiskoordinatensystems und sind daher immer gleich $[0, 0, 0]$.

Die Endkoordinaten $[X_e, Y_e, Z_e]$ des Kompensationsvektor werden durch die Istpositionen von 3 Kanalachsen, den sogenannten Richtungsachsen gebildet.

Richtungsachsen

Die Richtungsachsen müssen folgende Bedingungen erfüllen:

1. Die Richtungsachsen müssen Kanalachsen des Kanals sein in dem die Abstandsregelung aktiviert wird.
2. Die Richtungsachsen müssen Linearachsen sein.

Hinweis:

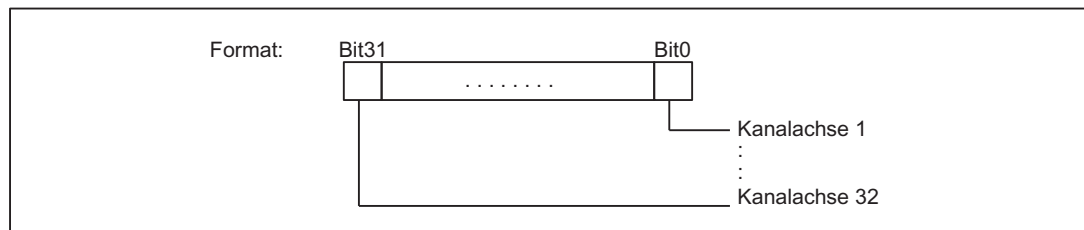
Da die Richtungsachsen nur zur Interpolation der Richtungskomponenten eingesetzt werden, benötigen sie keine mechanischen Achsen und können deshalb als Simulationsachsen parametrierung werden.

3. Als Maßeinheit der Richtungsachsen muss [mm] oder [inch] angewählt sein.
4. Die Richtungsachsen dürfen nicht Bestandteil einer Achskopplung wie z. B. Transformation, elektronisches Getriebe etc. sein.
5. Um sicherzustellen, dass die Bahndynamik nicht aufgrund der Achsdynamik der Richtungsachsen begrenzt wird, sind folgenden Maschinendaten der Richtungsachsen gleich oder höher als die entsprechenden Werte der Geometrieachsen des Kanals einzustellen:
 - MD32000 \$MA_MAX_AX_VELO[x] (Maximale Achsgeschwindigkeit)
 - MD32200 \$MA_POSCTRL_GAIN[x] (KV-Faktor)
 - MD32230 \$MA_MAX_AX_ACCEL[x] (Konfiguration Lageregler-Struktur)
x = Achsnummer

Die Festlegung welche Kanalachse Richtungsachse ist, erfolgt über das Maschinendatum:

- MD62528 \$MC_CLC_PROG_ORI_AX_MASK (Progr. Orientierungsvektor: Achsmaske)

Jedes Maschinendatenbit entspricht einer Kanalachse.



- Koordinate X = Kanalachse entsprechend Bit a
- Koordinate Y = Kanalachse entsprechend Bit b
- Koordinate Z = Kanalachse entsprechend Bit c
mit $a < b < c$

Aktueller Differenzwinkel

Der Differenzwinkel ist der Winkel zwischen dem Vektor der Werkzeugorientierung und dem Kompensationsvektor. Soll der aktuelle Differenzwinkel von der Abstandsregelung in eine Systemvariable \$AC_PARAM[n] ausgegeben werden, ist der Index n der Systemvariablen in folgendes Maschinendatum einzutragen:

MD65530 \$MC_CLC_PROG_ORI_ANGLE_AC_PARAM ()

Zulässiger Grenzwinkel

Über den zulässigen Grenzwinkel wird der maximal erlaubte Differenzwinkel zwischen dem Vektor der Werkzeugorientierung und dem Kompensationsvektor vorgegeben. Der Grenzwinkel wird über das folgende Maschinendatum parametrisiert:

MD65520 \$MC_CLC_PROG_ORI_MAX_ANGLE ()

18.5.5 Parametrierung der Abstandsregelung

Teileprogrammname

Zur Deklaration der funktionsspezifischen Teileprogrammname CLC_GAIN und CLC_VOFF sind folgende Maschinendaten zu parametrieren:

- MD10712 \$MN_NC_USER_CODE_CONF_NAME_TAB[0] = "OMA1" (Liste umprojektierter NC-Codes)
- MD10712 \$MN_NC_USER_CODE_CONF_NAME_TAB[1] = "CLC_GAIN"
- MD10712 \$MN_NC_USER_CODE_CONF_NAME_TAB[2] = "OMA2"
- MD10712 \$MN_NC_USER_CODE_CONF_NAME_TAB[3] = "CLC_VOFF"

1D-/3D-Abstandsregelung

Die Auswahl zwischen 1D- oder 3D-Abstandsregelung erfolgt über das Maschinendatum:

- MD62500 \$MC_CLC_AXNO = <n> (Achszuordnung der Abstandsregelung)
 - <n> > 0: 1D-Abstandsregelung mit <n> = Achsnummer der abstandsgeregelten Kanalachse
 - <n> = -1: 1. im Kanal konfigurierte 5-Achstransformation
 - <n> = -2: 2. im Kanal konfigurierte 5-Achstransformation

Eingangssignale

Die oben parametrierten Eingangssignale des Abstandssensors werden über folgende Maschinendaten der Abstandsregelung bekannt gemacht (siehe auch Kapitel "Parametrierung der Eingangssignale (Seite 715)"):

- MD62502 \$MN_CLC_ANALOG_IN = <n> (Analogeingang für die Anstandsregelung)
<n> = Eingangsnummer, analog der Adressierung der Systemvariablen \$A_INA[<n>]
- MD62504 \$MN_CLC_SENSOR_TOUCHED_INPUT = <n> (Zuordnung eines Eingangsbits für das Signal "Sensor-Kollision")
<n> = Eingangsnummer, analog der Adressierung der Systemvariablen \$A_IN[<n>]

Genauhalt

Um eine programmierte Satzwechselbedingung "Genauhalt grob/fein erreicht" (G601/G602) erfüllen zu können, muss die von der Abstandsregelung erzeugte Verfahrgeschwindigkeit in den abstandsgeregelten Achsen mindestens für die Dauer der Stillstand-Verzögerungszeit kleiner als die Stillstand-Geschwindigkeitstoleranz sein.

Zur Optimierung der Satzwechselzeit sind folgende Maschinendaten anzupassen:

- MD36000 \$MA_STOP_LIMIT_COARSE[<x>] (Genauhalt grob)
- MD36010 \$MA_STOP_LIMIT_FINE[<x>] (Genauhalt fein)
- MD36020 \$MA_POSITIONING_TIME[<x>] (Verzögerungszeit Genauhalt fein)

- MD36040 \$MA_STANDSTILL_DELAY_TIME[<x>] (Verzögerungszeit Stillstandsüberwachung)
- MD36060 \$MA_STANDSTILL_VELO_TOL[<x>] (Schwellgeschwindigkeit/Drehzahl "Achse/Spindel steht")
<x> = Achsnummer der abstandsgeregelten Maschinenachse

18.5.6 Inbetriebnahme der Abstandsregelung

Abstandssensor

Die Ausgänge des Abstandssensors sind an den Peripheriebaugruppen anzuschließen, die über folgende Maschinendaten aktiviert wurden:

- MD10362 \$MN_HW_ASSIGN_ANA_FASTIN (E/A-Adresse der Peripheriebaugr.) (Hardwarezuordnung der schnellen analogen NC-Eingänge)
- MD10366 \$MN_HW_ASSIGN_DIG_FASTIN (E/A- Adresse der Peripheriebaugr.) (Hardwarezuordnung der externen digitalen NC-Eingänge)

(Siehe auch Kapitel "Randbedingungen > Peripheriebaugruppen (Seite 738)")

Regelsinn testen

Zum Testen des Regelsinns der Abstandsregelung kann folgendermaßen vorgegangen werden:

- Einschalten der Anstandsregelung über ein Teileprogramm mit CLC(1) (siehe Kapitel "Ein- und Ausschalten der Abstandsregelung (CLC) (Seite 721)")
- Generieren einer Eingangsspannung z. B. durch folgende Synchronaktion:

Programmcode

```
N100 $AC_TIMER[1]=2.5
N110 ID = 1 EVERY $AC_TIMER[1] >= 2.5 DO $AC__TIMER[1]=0
N120 ID = 2 WHENEVER $AC_TIMER[1] < 2.0 DO $A_OUTA[6] = 100000.0 *
($AC_TIMER[1] - 1.0)
N130 ID = 3 WHENEVER $AC_TIMER[1] >= 2.0 DO $A_OUTA[6] = 0.0
```

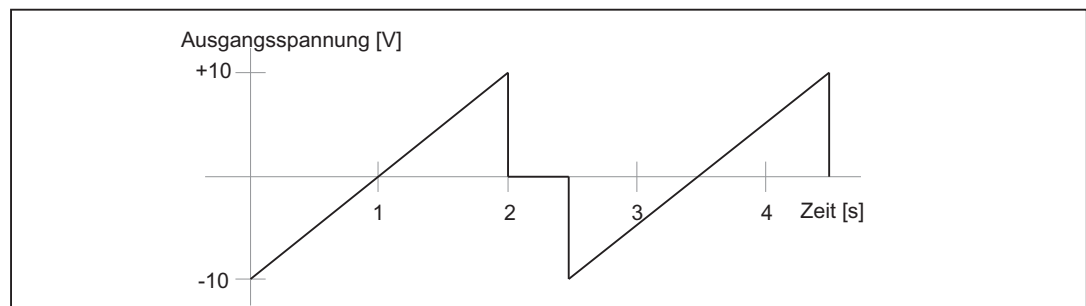


Bild 18-9 Ausgangsspannung der Synchronaktion

Die Spannungsvorgabe über den in der Synchronaktion verwendeten analogen Ausgang \$A_OUTA[6] wird von der Abstandsregelung von der Eingangsspannung des Abstandssensors subtrahiert, hat also die entgegengesetzte Polarität des Eingangssignals.

Damit die Abstandsregelung den Analogausgang 6 (\$A_OUTA[6]) als zusätzlichen, den Sensoreingang überlagernden Eingang verwendet, ist folgendes Maschinendatum zu setzen:

MD62522 \$MN_CLC_OFFSET_ASSIGN_ANAOUT = 6 (Hardwarezuordnung der externen digitalen NC-Eingänge)

Hinweis

Vor dem erstmaligen Einschalten der Abstandsregelung ist darauf zu achten, dass Kollisionsfreiheit im gesamten für die Abstandsregelung freigegebenen Arbeitsraum besteht:

- MD62505 \$MC_CLC_SENSOR_LOWER_LIMIT (Untere Bewegungsgrenze der Abstandsregelung)
- MD62506 \$MC_CLC_SENSOR_UPPER_LIMIT (Obere Bewegungsgrenze der Abstandsregelung)

Ein falscher Regelsinn kann durch jeweils eine der folgenden Maßnahmen korrigiert werden:

- Umpolung des Analogeingangs
- Vorzeichenänderung aller Werte in den Maschinendaten:
 - MD62511 \$MC_CLC_SENSOR_VELO_TABLE_1 (Koordinate Geschwindigkeit der Stützpunkte Sensorkennlinie 1)
 - MD62513 \$MC_CLC_SENSOR_VELO_TABLE_2 (Koordinate Geschwindigkeit der Stützpunkte Sensorkennlinie 2)

Sensorsignal

Signalqualität

Die Qualität des analogen Eingangssignals kann mittels der funktionsspezifischen Anzeigedaten überprüft werden (siehe Kapitel "Funktionsspezifische Anzeigedaten (Seite 734)").

Eingangsspannungsbereich

Der Eingangsspannungsbereich für das Messsignal des Sensors kann über das Maschinendatum für den Bewertungsfaktor für die analogen NC-Eingänge angepasst werden:

MD10320 \$MN_FASTIO_ANA_INPUT_WEIGHT[<Analogeingang>] = <Bewertungsfaktor>

Damit der Bewertungsfaktor von der Abstandsregelung eingerechnet wird, muss folgendes Maschinendatum gesetzt werden:

MD62508 \$MC_CLC_SPECIAL_FEATURE_MASK, Bit 13 = 1

Funktionsspezifische Alarmtexte

Zur Anzeige der funktionsspezifischen Alarmtexte müssen diese zuerst in die entsprechende HMI-Datenhaltung eingebracht werden (siehe Kapitel "Anlegen von Alarmtexten (Seite 696)").

Abschluss

Es wird empfohlen, nach Abschluss der Inbetriebnahme eine Datensicherung durchzuführen.

Literatur:

Inbetriebnahmehandbuch IBN CNC: NC, PLC, Antrieb

Hinweis

Es wird empfohlen, nach Abschluss der Inbetriebnahme eine Datensicherung durchzuführen.

18.6 Programmierung

18.6.1 Ein- und Ausschalten der Abstandsregelung (CLC)

Syntax

CLC(*Mode*)

Mode

- Format: Integer
- Wertebereich: -1, 0, 1, 2, 3

CLC(...) ist ein Prozedur-Aufruf und muss daher in einem eigenen Teileprogramm-Satz programmiert werden.

Funktionalität

Zum Ein-/Ausschalten der Abstandsregelung stehen folgende Modi zur Verfügung:

- CLC(1)
Einschalten der Abstandsregelung mit Kompensationsvektor in Richtung der Werkzeugorientierung
Die Auswertung des Sensor-Kollisionssignals ist ausgeschaltet.
- CLC(2)
Einschalten der Abstandsregelung mit Kompensationsvektor in Richtung der Werkzeugorientierung
Die Auswertung des Sensor-Kollisionssignals ist eingeschaltet.
- CLC(3)
Einschalten der Abstandsregelung mit programmiertem Kompensationsvektor
Die Auswertung des Sensor-Kollisionssignals ist ausgeschaltet.

- CLC(0)
Ausschalten der Abstandsregelung ohne Herausfahren des Positionsoffsets. Wenn sich die abstandsgeregelten Achsen aufgrund des Sensorsignals zum Ausschaltzeitpunkt noch bewegen, werden sie gestoppt. Das Werkstückkoordinatensystem (WKS) wird anschließend auf die entsprechenden Stillstandpositionen synchronisiert. Dabei wird ein automatischer Vorlaufstopp ausgeführt.
- CLC(-1)
Ausschalten der Abstandsregelung mit Herausfahren des Positionsoffsets. Wenn sich die abstandsgeregelten Achsen aufgrund des Sensorsignals zum Ausschaltzeitpunkt noch bewegen, werden sie gestoppt. Ein Positionsoffset zur zuletzt programmierten Position wird mit dem Ausschaltbefehl automatisch herausgefahren.


RESET-Verhalten

Bei Reset (NC-RESET oder Programmende) wird implizit CLC(0) ausgeführt.

Parametrierbares RESET-Verhalten

Das Restverhalten einer 1D-Abstandsregelung kann festgelegt werden über das kanalspezifische NC-OEM Maschinendatum:

- MD62524 \$MC_CLC_ACTIVE_AFTER_RESET (Restverhalten bei aktiver CLC)

 VORSICHT
Abstandsregelung
Nur im Zusammenhang mit einer 1D-Abstandsregelung ist das kanalspezifische NC-OEM-Maschinendatum MD62524 wirksam.
Bei einer 3D-Abstandsregelung wird bei Reset immer CLC(0) wirksam.

Folgendes Verhalten ist parametrierbar:

- MD62524 \$MC_CLC_ACTIVE_AFTER_RESET = 0
Die Abstandsregelung verhält sich bei Reset wie beim Ausschalten mit CLC(0) (siehe Abschnitt "Funktionalität").
- MD62524 \$MC_CLC_ACTIVE_AFTER_RESET = 1
Der aktuelle Zustand der Abstandsregelung bleibt erhalten.

Randbedingungen

Folgende Randbedingungen sind zu beachten:

Bahnsteuerbetrieb

Ein-/Ausschalten der Abstandsregelung (CLC(*Mode*)) während aktivem Bahnsteuerbetriebes (G64/G64x) führt zu einem Geschwindigkeitseinbruch der Bahnbewegungen. Zur Vermeidung derartiger Geschwindigkeitseinbrüche, ist die Abstandsregelung vor einem Bahnabschnitt mit konstanter Bahngeschwindigkeit einzuschalten. Während des entsprechenden Bahnabschnittes kann dann gegebenenfalls über den programmierbaren Verstärkungsfaktor der Abstandsregelung (CLC_GAIN) die Abstandsregelung blockiert und wieder freigegeben werden.

Satzwechsel mit Genauhalt

Ist Genauhalt zum Satzende (G60/G09 mit G601/G602) aktiv, kann der Satzwechsel aufgrund von Achsbewegungen, ausgelöst durch das Sensorsignal der Abstandsregelung, verzögert werden.

Sensorkollisionsüberwachung

Ein digitaler Eingang für ein zusätzliches Kollisionssignal kann vom Sensor mit dem folgenden Maschinendatum konfiguriert werden:

MD62504 \$MC_CLC_SENSOR_TOUCHED_INPUT (Zuordnung eines Eingangssignals für das Signal "Sensor-Kollision")

Diese Kollisionsüberwachung kann durch abwechselnde Programmierung von CLC(1)/CLC(2) satzsynchron ein- und ausgeschaltet werden.

Als Reaktion auf das Kollisionssignal des Sensors fährt die Abstandsregelung unabhängig von der Feedrate-Override-Einstellung mit maximaler Geschwindigkeit in Plus-Richtung, bis sie die aktuell gültige obere Begrenzung erreicht. Gleichzeitig wird die Bahnbewegung gestoppt.

Mit NC-START ist das Fortsetzen der Bearbeitung möglich.

3D-Abstandsregelung und 5-Achs-Transformation

Wird eine 3D-Abstandsregelung eingeschaltet, bevor die zur Abstandsregelung in Richtung der Werkzeugorientierung benötigte 5-Achs-Transformation aktiviert wurde, arbeitet die Abstandsregelung abhängig von der aktiven Arbeitsebene (G17/G18/G19):

- G17: Richtung der Abstandsregelung = Z
- G18: Richtung der Abstandsregelung = Y
- G19: Richtung der Abstandsregelung = X

Einschalten der 5-Achs-Transformation

Beim Einschalten der 5-Achs-Transformation muss die durch die Rundachspositionen vorgegebene Werkzeugorientierung mit der bei Einschalten der Abstandsregelung durch die aktive Arbeitsebene festgelegten Regelrichtung übereinstimmen.

Stimmen Werkzeugorientierung der 5-Achs-Transformation und Regelrichtung der Abstandsregelung nicht überein, wird folgender CLC-Alarm angezeigt:

- Alarm "75016 Kanal *Nummer* Satz *Nummer* CLC: Orientierung geändert bei TRAFOOF."

Ausschalten der 5-Achs-Transformation

Wird bei aktiver Abstandsregelung die 5-Achs-Transformation ausgeschaltet, wird die letzte Regelrichtung vor dem Ausschalten der 5-Achs-Transformation beibehalten.

Werkzeugradius-Korrektur

Eine 3D-Abstandsregelung darf nur ausgeschaltet werden, wenn zum Ausschaltzeitpunkt keine Werkzeugradiuskorrektur im Kanal aktiv ist (G40). Bei aktiver Werkzeugradiuskorrektur (G41/G42) wird folgender Alarm angezeigt:

- Alarm "75015 Kanal *Nummer* Satz *Nummer* CLC(0) bei aktiver WRK."

Kompensationsvektor

Istposition der Richtungsachsen

Wird die Abstandsregelung mit programmierbarem Kompensationsvektor bei einer Position von 0 in allen 3 Richtungsachsen eingeschaltet, kann daraus kein Kompensationsvektor berechnet werden. Es wird dann folgender Alarm angezeigt:

- Alarm "75019 Kanal *Nummer*, Fehler-ID: 1, Winkel 0.0"

Referenzieren der Richtungsachsen

Vor dem Aktivieren der Abstandsregelung mit programmierbarem Kompensationsvektor CLC(3) müssen die Richtungsachsen referenziert werden.

Nahtstellensignale der Richtungsachsen

Vor dem Aktivieren der Abstandsregelung mit programmierbarem Kompensationsvektor CLC(3) sind durch das PLC-Anwenderprogramm für alle 3 Richtungsachsen folgende Nahtstellensignale zu setzen:

- DBX31, ... DBX1.5 = 1 (Lagemesssystem 1)
- DBX31, ... DBX2.1 = 1 (Reglerfreigabe)
- DBX31, ... DBX21.7 = 1 (Impulsfreigabe)
x = Achsnummer

Umschalten der Abstandsregelung

Ein direktes Umschalten der Abstandsregelung von CLC(1) oder CLC(2) nach CLC(3) oder umgekehrt ist nicht möglich. Eine derartige Umschaltung wird ohne Rückmeldung ignoriert. Wird eine Umschaltung notwendig, ist die Abstandsregelung mit CLC(0) oder CLC(-1) zuerst auszuschalten und dann im gewünschten Mode wieder einzuschalten.

Interpolation des Kompensationsvektors

Soll der Kompensationsvektor bezüglich seiner Orientierung einer nichtlinearen Werkstückoberfläche z. B. Kreisbogen folgen, kann dies durch entsprechende Programmierung der Richtungsachsen erreicht werden.

Beispiel

Orientierung des Kompensationsvektors senkrecht zu einer halbkreisförmigen Werkstückoberfläche. Die Programmierung der Verfahrbewegung ist nicht berücksichtigt.

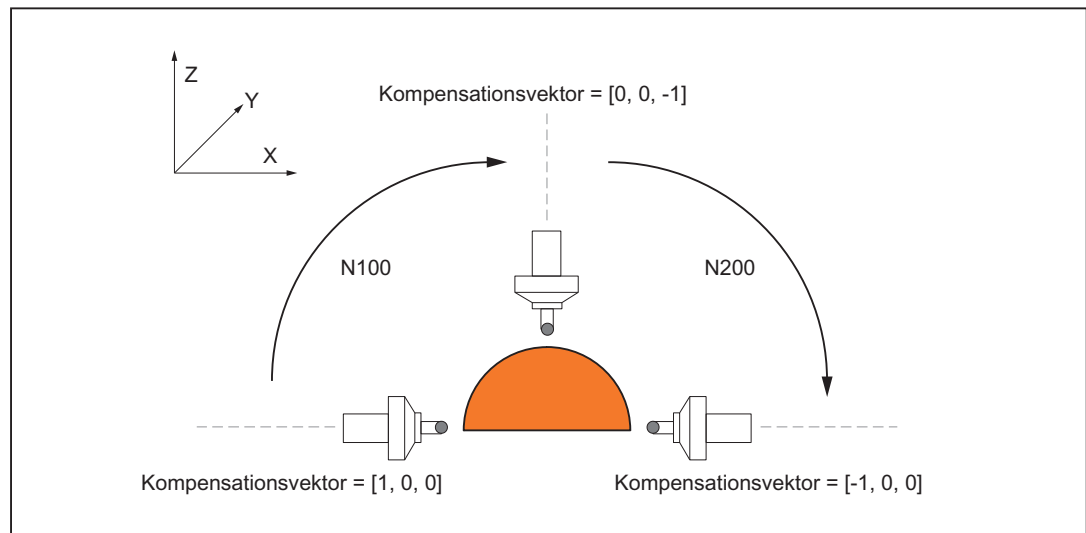


Bild 18-10 Interpolation des Kompensationsvektors

Vor dem Teileprogrammsatz N100 ist der Kompensationsvektor durch Programmierung der Richtungsachsen auf $[1, 0, 0]$ orientiert worden. Im Teileprogrammsatz N100 wird die Endposition des Kompensationsvektors durch Programmierung der Richtungsachsen auf $[0, 0, -1]$ orientiert.

Die Zwischenwerte entstehen durch Bahninterpolation aller im Teileprogrammsatz programmierten Achsen:

- Geometrieachsen für die Bewegung des Bearbeitungskopfes
- Richtungsachsen des Kompensationsvektors

Eine Zerlegung der Bewegung in die Teileprogrammsätze N100 und N200 ist notwendig, da sich ansonsten eine antiparallele Orientierung des Kompensationsvektors von $[1, 0, 0]$ am Anfang der Bewegung und $[-1, 0, 0]$ am Ende der Bewegung (Halbkreis) ergäbe. Der Interpolator würde in diesem Fall nur die X-Koordinate des Kompensationsvektors interpolieren und die Orientierung des Kompensationsvektors würde sich nicht ändern.

Antiparallele Orientierung des Kompensationsvektors

Wird in einem Teileprogrammsatz eine antiparallele Orientierung des Kompensationsvektors programmiert, wird folgender Alarm angezeigt:

- Alarm "75018 Kanal NummerSatz NummerCLC in programmierbare Richtung, Fehler-ID: 1"

Hinweis

Interpolation des Kompensationsvektors

Die Interpolation des Kompensationsvektors ist wie oben beschrieben keine echte Vektorinterpolation, sondern entsteht durch Interpolation der Istpositionen der Richtungsachsen.

Dies hat zur Folge, dass bei einer Änderung des Kompensationsvektors aufgrund der Werkstückkontur, die Interpolation der Richtungsachsen mit in die Bahninterpolation der Geometrieachsen eingeht. Um den Einfluss der Richtungsachsen auf die Bahninterpolation möglichst gering zu halten, wird empfohlen, die Dynamik der Richtungsachsen mindestens

gleich groß oder größer (ca. Faktor 10) als die Dynamik der Geometrieachsen zu parametrieren.

Weiter ist bei einer Umorientierung (Drehung) des Kompensationsvektors das Verhältnis des programmierten Verfahrenweges zur parametrisierten Dynamik der Richtungsachsen zu beachten. Das Verhältnis sollte so gewählt werden, dass der programmierte Verfahrenweg aufgrund der Achsdynamik nicht in einem bzw. einigen wenigen Interpolationstakten abgefahren wird. Dies führt zu hohen Belastungen an der Maschine und unter Umständen zu axialen Alarmen mit Abbruch der Teileprogrammabarbeitung.

Beispiel

Drehung des Kompensationsvektors und damit des Bearbeitungskopfes um 90°:

- Ausgangsorientierung: Parallel zur Koordinatenachse X
- Zielorientierung Parallel zur Koordinatenachse Y
Ungünstige Programmierung der Umorientierung:
- $[1, 0, 0] \rightarrow [0, 1, 0]$
Vorteilhafte Programmierung der Umorientierung:
- $[100, 0, 0] \rightarrow [0, 100, 0]$

Drehung des Werkstückkoordinatensystems

Der Kompensationsvektor bezieht sich, wie oben beschrieben, immer auf das Basiskoordinatensystem (Maschinenkoordinatensystem). Erfolgt zur Bearbeitung des Werkstücks eine Transformation des Werkstückkoordinatensystems (Drehung, Spiegelung) so, dass die Koordinatenachsen der beiden Koordinatensysteme nicht mehr parallel und gleichorientiert verlaufen, ist eine entsprechende Transformation für den Kompensationsvektor vorzunehmen.



VORSICHT

Kein gleichorientierter Verlauf

Erfolgt eine Transformation des Werkstückkoordinatensystems so, dass die Koordinatenachsen des Basis- und Werkstückkoordinatensystems nicht mehr parallel und gleichorientiert verlaufen, liegt die entsprechende Transformation des Kompensationsvektors in der alleinigen Verantwortung des Anwenders.

18.6.2 Regelkreisverstärkung (CLC_GAIN)

Syntax

$CLC_GAIN = Faktor$

Faktor

- Format: Real
- Wertebereich: $y \geq 0.0$

CLC_GAIN ist eine NC-Adresse und kann daher zusammen mit anderen Anweisungen in einem Teileprogrammsatz geschrieben werden.

Bei Programmierung eines negativen Faktors wird ohne Alarm der Betragswert verwendet.

Funktionalität

Die aktuelle Regelkreisverstärkung der Abstandsregelung ergibt sich aus der aktiven, über Maschinendaten vorgegebenen Kennlinie:

- MD62510 \$MC_CLC_SENSOR_VOLTAGE_TABLE1 (Koordinate Spannung der Stützpunkte Sensorkennlinie 1)
- MD62511 \$MC_CLC_SENSOR_VELO_TABLE1 (Koordinate Geschwindigkeit der Stützpunkte Sensorkennlinie 1) bzw.
- MD62512 \$MC_CLC_SENSOR_VOLTAGE_TABLE2 (Koordinate Spannung der Stützpunkte Sensorkennlinie 2)
- MD62513 \$MC_CLC_SENSOR_VELO_TABLE2 (Koordinate Geschwindigkeit der Stützpunkte Sensorkennlinie 2)

Über CLC_GAIN kann die Regelkreisverstärkung der Kennlinie multiplikativ mit einem programmierbaren Faktor beaufschlagt werden.

ACHTUNG

Unpräzise Kennlinie

Ein Vergrößern der Verstärkung ($CLC_GAIN > 1.0$) kann zu Schwingungen der geregelten Achsen führen!

Aktivierungszeitpunkt

Die geänderte Regelkreisverstärkung wird in dem Teileprogrammsatz wirksam, in dem CLC_GAIN programmiert wurde, bzw. wenn dieser Satz keine ausführbaren Anweisungen enthält, im nächsten Teileprogrammsatz mit ausführbaren Anweisungen.

Verhalten bei Kennlinienumschaltung

Der programmierte Faktor bleibt auch nach einem Umschalten der Verstärkungskennlinie (CLC_SEL) wirksam, d. h. er wirkt sofort auf die neu angewählte Kennlinie.

Verhalten bei CLC_GAIN=0.0

Wird die Regelkreisverstärkung der Abstandsregelung mit CLC_GAIN=0.0 ausgeschaltet, bleibt der zum Ausschaltzeitpunkt vorhandene CLC-Positionsoffset unverändert erhalten. Dies kann z. B. beim Laserschneiden von Blechen dazu verwendet werden, um ohne einzusinken bereits herausgefallene Blechausschnitte zu "überfliegen".

Wird bei aktiver 3D-Abstandsregelung und ausgeschalteter Regelkreisverstärkung (CLC_GAIN=0.0) die Werkzeugorientierung verändert, wird der CLC-Offsetvektor mitgedreht. Dies führt in der Regel zu einer Verschiebung des CLC-Arbeitspunktes auf der Werkstückoberfläche (siehe folgendes Bild).

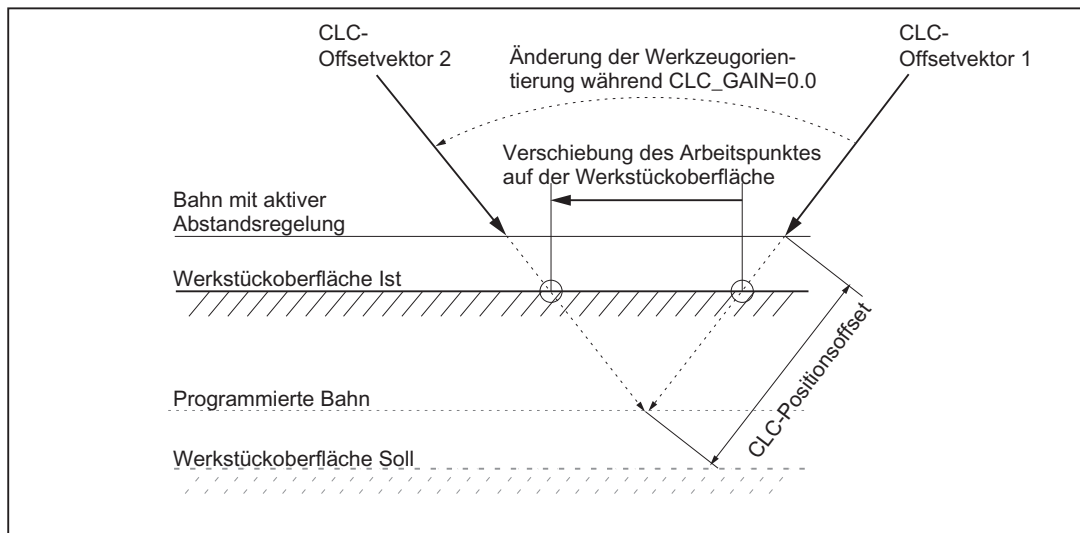


Bild 18-11 Verhalten des CLC-Versatzvektors während CLC_GAIN=0.0

Rücksetzen

Innerhalb eines Teileprogramms muss ein geänderter Verstärkungsfaktor durch explizite Programmierung von CLC_GAIN=1.0 rückgesetzt werden.

RESET-Verhalten

Nach Power-On-Reset, NC-RESET oder Programmende ist CLC_GAIN=1.0 wirksam.

18.6.3 Begrenzung des Regelbereichs (CLC_LIM)

Syntax

CLC_LIM(*Untergrenze, Obergrenze*)

Untergrenze, Obergrenze

Format und Wertebereich wie Maschinendaten:

- MD62505 \$MC_CLC_SENSOR_LOWER_LIMIT[n] (Untere Bewegungsgrenze der Abstandsregelung)
- MD62506 \$MC_CLC_SENSOR_UPPER_LIMIT[n] (Obere Bewegungsgrenze der Abstandsregelung)

CLC_LIM(...) ist ein Prozedur-Aufruf, und muss daher in einem eigenen Teileprogramm-Satz programmiert werden.

Funktionalität

Der maximale Regelbereich der Abstandsregelung kann über CLC_LIM satzspezifisch angepasst werden. Begrenzt wird die maximal programmierbare Unter- bzw. Obergrenze durch den im jeweiligen Maschinendatum vorgegebenen Grenzwert:

- MD62505 \$MC_CLC_SENSOR_LOWER_LIMIT[1] (Untere Bewegungsgrenze der Abstandsregelung)
- MD62506 \$MC_CLC_SENSOR_UPPER_LIMIT[1] (Obere Bewegungsgrenze der Abstandsregelung)

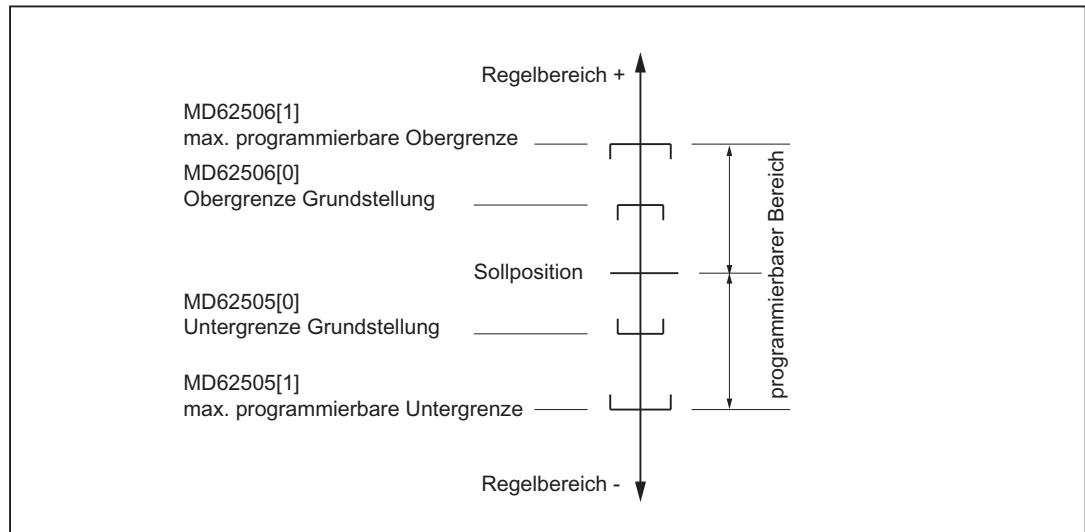


Bild 18-12 Wertebereichsgrenzen für Unter- und Obergrenze

Die Begrenzung des Regelbereiches wirkt relativ zur aktuell programmierten Sollposition der Achse. Werden die Begrenzungen so verändert, dass die Istposition außerhalb der Begrenzung liegt, wird von der Abstandsregelung automatisch in den Begrenzungsbereich zurückgefahren.

Rücksetzen

Innerhalb eines Teileprogramms kann eine veränderte Begrenzung des Regelbereiches durch explizite Programmierung von CLC_LIM ohne Argumente "CLC_LIM()" rückgesetzt werden. Anschließend sind wieder die Begrenzungen aus den folgenden Maschinendaten wirksam:

- MD62505 \$MC_CLC_SENSOR_LOWER_LIMIT[0] (Untere Bewegungsgrenze der Abstandsregelung)
- MD62506 \$MC_CLC_SENSOR_UPPER_LIMIT[0] (Obere Bewegungsgrenze der Abstandsregelung)

RESET-Verhalten

Nach Power-On-Reset, NC-RESET und Programmende wird die Voreinstellung aus den oben aufgeführten Maschinendaten wirksam.

Fehlermeldungen

Folgende Programmierfehler werden mit einem Alarm angezeigt:

- Programmierung von mehr als 2 Argumenten
 - CLC-Alarm "75005 Kanal *Nummer* Satz *Nummer* CLC_LIM: allgemeiner Programmierfehler"
- Programmierung von Argumenten außerhalb der zulässigen Grenzen
 - CLC-Alarm "750010 Kanal *Nummer* Satz *Nummer* CLC_LIM Wert größer als MD-Grenze"

18.6.4 Richtungsabhängiges Sperren der Verfahrbewegung

Syntax

$\$A_OUT[Nummer] = Freigabesignal$

Nummer

Nummer des parametrisierten digitalen Ausganges (siehe Abschnitt "Parametrierung")

- Format: Integer
- Wertebereich: 1, 2, . . . max. Anzahl digitaler Ausgänge

Freigabesignal

Invertierbares Freigabesignal (siehe Abschnitt "Parametrierung")

- Format: Integer
- Wertebereich: 0, 1

Die Systemvariable $\$A_OUT[n]$ kann satzsynchron in Teileprogramm oder asynchron über Synchronaktionen gesetzt werden.

Funktionalität

Über parametrierbare digitale Ausgänge (Systemvariable $\$A_OUT$), lässt sich die durch die Abstandsregelung erzeugte Verfahrbewegung (Stellgröße) richtungsabhängig sperren. So lange z. B. die negative Verfahrrichtung gesperrt ist, verfahren die abstandsgeregelten Achsen aufgrund des Sensorsignals nur in positiver Verfahrrichtung.

Dies kann z. B. beim Laserschneiden von Blechen dazu verwendet werden, um ohne einzusinken bereits herausgefallene Blechausschnitte zu "überfliegen".

Parametrierung

Die Parametrierung der digitalen Ausgänge erfolgt über das Maschinendatum:

- MD62523 \$MC_CLC_LOCK_DIR_ASSIGN_DIGOUT[n] (Zuordnung der Digitalausgänge für Verriegelung der CLC-Bewegung)
 - n = 0 → digitaler Ausgang zum Sperren der negativen Verfahrrichtung
 - n = 1 → digitaler Ausgang zum Sperren der positiven Verfahrrichtung

Beispiel

Folgende digitale Ausgänge sollen verwendet werden:

- \$A_OUT[3] zum Sperren der negativen Verfahrrichtung
- \$A_OUT[4] zum Sperren der positiven Verfahrrichtung

Parametrierung in den Maschinendaten:

- MD62523 \$MC_CLC_LOCK_DIR_ASSIGN_DIGOUT[0] = 3 (Zuordnung der Digitalausgänge für Verriegelung der CLC-Bewegung)
- MD62523 \$MC_CLC_LOCK_DIR_ASSIGN_DIGOUT[1] = 4

Wirkung:

- \$A_OUT[3] = 0 → Die negative Verfahrrichtung ist freigegeben
- \$A_OUT[3] = 1 → Die negative Verfahrrichtung ist gesperrt
- \$A_OUT[4] = 0 → Die positive Verfahrrichtung ist freigegeben
- \$A_OUT[4] = 1 → Die positive Verfahrrichtung ist gesperrt

Invertierung der Auswertung

Durch Eingabe der negativen Nummer des digitalen Ausgangs, wird das digitale Ausgangssignal invertiert ausgewertet:

Parametrierung in den Maschinendaten:

- MD62523 \$MC_CLC_LOCK_DIR_ASSIGN_DIGOUT[0] = -3 (Zuordnung der Digitalausgänge für Verriegelung der CLC-Bewegung)
- MD62523 \$MC_CLC_LOCK_DIR_ASSIGN_DIGOUT[1] = -4

Wirkung:

- \$A_OUT[3] = 0 → Die negative Verfahrrichtung ist gesperrt
- \$A_OUT[3] = 1 → Die negative Verfahrrichtung ist freigegeben
- \$A_OUT[4] = 0 → Die positive Verfahrrichtung ist gesperrt
- \$A_OUT[4] = 1 → Die positive Verfahrrichtung ist freigegeben

18.6.5 Satzweise vorgebbarer Spannungsoffset (CLC_VOFF)

Syntax

$CLC_VOFF = \text{Spannungsoffset}$

Spannungsoffset

- Format: Real
- Einheit: Volt
- Wertebereich: keine Einschränkungen

CLC_VOFF ist eine NC-Adresse und kann daher zusammen mit anderen Anweisungen in einem Teileprogrammsatz geschrieben werden.

Funktionalität

Über CLC_VOFF kann der Abstandsregelung ein konstanter Spannungsoffset vorgegeben werden, der von der Eingangsspannung des Abstandssensors subtrahiert wird. Der programmierte Spannungsoffset bewirkt somit eine Änderung des Sollabstandes zwischen Wertstück und Abstandssensor bzw. eine Verschiebung des Arbeitspunktes der Abstandsregelung.

Der quantitative Einfluss des Spannungsoffsets ist abhängig von den weiteren Parametern der Abstandsregelung und kann daher nicht allgemeingültig normiert werden.

Aktivierungszeitpunkt

Der Spannungsoffset wird in dem Teileprogrammsatz wirksam, in dem CLC_VOFF programmiert wurde, bzw. wenn dieser Satz keine ausführbaren Anweisungen enthält, im nächsten Teileprogrammsatz mit ausführbaren Anweisungen.

Rücksetzen

Innerhalb eines Teileprogramms muss ein Spannungsoffset durch explizite Programmierung von CLC_VOFF =0.0 rückgesetzt werden.

RESET-Verhalten

Nach Power-On-Reset, NC-RESET oder Programmende ist CLC_VOFF =0.0 wirksam.

18.6.6 Per Synchronaktion vorgebbarer Spannungsoffset

Syntax

$\$A_OUTA[\text{Nummer}] = \text{Spannungsoffset}$

Nummer

Nummer des parametrierbaren analogen Ausgangs (siehe Abschnitt "Parametrierung")

- Format: Integer
- Wertebereich: 1, 2, . . . max. Anzahl analoger Ausgänge

Spannungsoffset

Wie Spannungsoffset bei CLC_VOFF (siehe Kapitel "Satzweise vorgegebener Spannungsoffset (CLC_VOFF) (Seite 732)").

Funktionalität

Über einen parametrierbaren analogen Ausgang (Systemvariable \$A_OUTA), lässt sich der Abstandsregelung ein Spannungsoffset vorgeben, der wie CLC_VOFF von der Eingangsspannung des Abstandssensors subtrahiert wird.

Durch Programmierung des analogen Ausgangs innerhalb einer Synchronaktion kann der Spannungsoffset im Interpolatortakt angepasst werden.

Parametrierung

Die Parametrierung des analogen Ausgangs erfolgt über das Maschinendatum:

MD62522 \$MC_CLC_OFFSET_ASSIGN_ANAOUT (Änderung des Soll-Abstands durch Überlagerung des Sensorsignals)

Beispiel

Am analogen Eingang \$A_INA[3] liegt eine externe Spannung Uext, die als kontinuierlich veränderbarer Spannungsoffset z. B. zu Test- oder Inbetriebnahmезwecken der Sensorspannung überlagert werden soll. Als analoger Ausgang für den Spannungsoffset der Abstandsregelung wird \$A_OUTA[2] verwendet.

Parametrierung des analogen Ausgangs für den Spannungsoffset der Abstandsregelung:

MD62522 \$MC_CLC_OFFSET_ASSIGN_ANAOUT = 2 (Änderung des Soll-Abstands durch Überlagerung des Sensorsignals)

Die Zuweisung des analogen Eingangs \$A_INA[3] auf den analogen Ausgang der Abstandsregelung \$A_OUTA[2] erfolgt innerhalb einer Synchronaktion:

```
ID=1 DO $A_OUTA[2] = $A_INA[3]
```

18.6.7 Auswahl der aktiven Sensorkennlinie (CLC_SEL)

Syntax

CLC_SEL(*Kennliniennummer*)

Kennliniennummer

- Format: Integer
- Wertebereich: 1, 2

CLC_SEL(...) ist ein Prozedur-Aufruf und muss daher in einem eigenen Teileprogramm-Satz programmiert werden.

Mit *Kennliniennummer*= 2 wird die Kennlinie 2 angewählt. Bei jedem anderen Wert wird ohne Alarm die Kennlinie 1 angewählt.

Funktionalität

Mit CLC_SEL kann zwischen den in den Maschinendaten definierten Sensorkennlinien umgeschaltet werden.

- Kennlinie 1:
 - MD62510 \$MC_CLC_SENSOR_VOLTAGE_TABLE_1 (Koordinate Spannung der Stützpunkte Sensorkennlinie 1)
 - MD62511 \$MC_CLC_SENSOR_VELO_TABLE_1 (Koordinate Geschwindigkeit der Stützpunkte Sensorkennlinie 1)
- Kennlinie 2:
 - MD62512 \$MC_CLC_SENSOR_VOLTAGE_TABLE_2 (Koordinate Spannung der Stützpunkte Sensorkennlinie 2)
 - MD62513 \$MC_CLC_SENSOR_VELO_TABLE_2 (Koordinate Geschwindigkeit der Stützpunkte Sensorkennlinie 2)

RESET-Verhalten

Nach Power-On-Reset, NC-RESET oder Programmende ist Kennlinie 1 wirksam.

18.7 Funktionsspezifische Anzeigedaten

Die Technologiefunktion "Abstandsregelung" stellt spezifische Anzeigedaten zur Unterstützung der Inbetriebnahme bzw. zu Service-Zwecken bereit.

Anwendungsmöglichkeiten

Anwendungsmöglichkeiten der Anzeigedaten sind z. B.:

- Ermittlung der Formabweichungen und kurzzeitig auftretender Regelfehler über die Variablen für den maximalen und minimalen Positionsoffset bzw. der Sensoreingangsspannung.
- Ermittlung des vom A/D-Wandlers erfasste Spannungsrauschens über die Variablen für die maximale und minimale Sensoreingangsspannung. Voraussetzungen sind dabei ein konstanter Abstand des Abstandssensors von der Werkstückoberfläche und das Ausschalten der Abstandsregelung mittels CLC_GAIN = 0.0.

Die Erfassung der minimalen und maximalen Werte erfolgt im Lagereglertakt.

Variablen-Typen

Die Anzeigedaten stehen sowohl als kanalspezifische GUD- (Global User Data) als auch als BTSS-Variable zur Verfügung.

18.7.1 Kanalspezifische GUD-Variable

Als Anzeigedaten stellt die Technologiefunktion "Abstandsregelung" folgende kanalspezifische GUD-Variable für die HMI-Applikationen zur Verfügung:

- SINUMERIK HMI Advanced
- SINUMERIK Operate

Tabelle 18-1 Kanalspezifische GUD-Variable

GUD-Variable	Bezeichnung	Einheit	Zugriff
CLC_DISTANCE[0]	aktueller Positionsoffset	mm	nur lesen
CLC_DISTANCE[1]	absolutes Minimum des Positionsoffsets	mm	lesen/schreiben
CLC_DISTANCE[2]	absolutes Maximum des Positionsoffsets	mm	lesen/schreiben
CLC_VOLTAGE[0]	aktuelle Sensoreingangsspannung	V	nur lesen
CLC_VOLTAGE[1]	absolutes Minimum der Sensoreingangsspannung	V	lesen/schreiben
CLC_VOLTAGE[2]	absolutes Maximum der Sensoreingangsspannung	V	lesen/schreiben

Nach erfolgter Inbetriebnahme der Technologiefunktion werden die genannten GUD-Variablen nicht automatisch auf der HMI-Oberfläche angezeigt.

SINUMERIK HMI Advanced

Zum Anlegen und Anzeigen der GUD-Variablen sind bei HMI Advanced folgende Bedienhandlungen durchzuführen.

1. Kennwort setzen
Es ist das Kennwort der Schutzstufe 1: (Maschinenhersteller) einzugeben.
2. Anzeige der "Definitionen" aktivieren
Bedienbereichsumschaltung > Dienste > Datenauswahl
3. Falls noch keine SGUD.DEF Datei vorhanden ist:
Bedienbereichsumschaltung > Dienste > Daten verwalten > Neu...
 - Name: SGUD
 - Typ: GlobaleDaten/System
mit **OK** bestätigen.
Die Datei wird daraufhin im Editor geöffnet.

1. GUD-Variablen-Definitionen editieren
DEF CHAN REAL CLC_DISTANCE[3] ; Array of real, 3 elements
DEF CHAN REAL CLC_VOLTAGE[3] ; Array of real, 3 elements
M30
2. Datei speichern und Editor schließen
3. Datei SGUD.DEF aktivieren

Die GUD-Variablen der Abstandsregelung werden jetzt angezeigt unter:

Bedienbereichsumschaltung > Parameter > Anwenderdaten > Kanal Anw.Daten

SINUMERIK Operate

Zum Anlegen und Anzeigen der GUD-Variablen sind bei SINUMERIK Operate folgende Bedienhandlungen durchzuführen.

1. Kennwort setzen
Es ist das Kennwort der Schutzstufe 1: (Maschinenhersteller) einzugeben.
2. Falls noch keine SGUD.DEF Datei vorhanden ist:
Bedienbereichsumschaltung > Inbetriebnahme > Systemdaten > Verzeichnis NC-Daten
öffnen: Cursor auf Definitionen stellen > **Neu...**
 - Name: SGUD
 - Typ: DEF
mit **OK** bestätigen.
Die Datei wird daraufhin im Editor geöffnet.
1. GUD-Variablen-Definitionen editieren
DEF CHAN REAL CLC_DISTANCE[3] ; Array of real, 3 elements
DEF CHAN REAL CLC_VOLTAGE[3] ; Array of real, 3 elements
M30
2. Datei speichern und Editor schließen
3. Datei SGUD.DEF aktivieren

Die GUD-Variablen der Abstandsregelung werden jetzt angezeigt unter:

Bedienbereichsumschaltung > Parameter > Anwendervariablen > Kanal GUD

SINUMERIK NC

Die neu angelegten und bereits angezeigten GUD-Variablen werden von der Abstandsregelung erst nach einem NC-POWER ON-RESET erkannt und mit aktuellen Werten versorgt.

Hinweis

Nach dem Anlegen der GUD-Variablen muss ein NC-POWER ON-RESET ausgelöst werden, damit die Abstandsregelung die GUD-Variablen aktualisiert.

18.7.2 BTSS-Variable

Die Technologiefunktion "Abstandsregelung" stellt folgende kanalspezifischen BTSS-Variablen als Anzeigedaten für die HMI-Applikation zur Verfügung:

BTSS-Variable	Bezeichnung	Einheit	Zugriff
CLC[0]	aktueller Positionsoffset	mm	nur lesen
CLC[1]	absolutes Minimum des Positionsoffsets	mm	lesen/schreiben
CLC[2]	absolutes Maximum des Positionsoffsets	mm	lesen/schreiben
CLC[3]	aktuelle Sensoreingangsspannung	V	nur lesen
CLC[4]	absolutes Minimum der Sensoreingangsspannung	V	lesen/schreiben
CLC[5]	absolutes Maximum der Sensoreingangsspannung	V	lesen/schreiben
CLC[6]	1. Komponente des normierten Werkzeugorientierungsvektors	-	nur lesen
CLC[7]	2. Komponente des normierten Werkzeugorientierungsvektors	-	nur lesen
CLC[8]	3. Komponente des normierten Werkzeugorientierungsvektors	-	nur lesen

Damit auf die BTSS-Variablen zugegriffen werden kann, müssen sie noch im System bekannt gemacht bzw. definiert werden.

Definieren der BTSS-Variablen

Zum Definieren der BTSS-Variablen sind folgenden Bedienhandlungen durchzuführen.

1. Anlegen der CLC-spezifischen Definitions-Datei: **CLC.NSK**

Hinweis:

Es wird empfohlen, die Datei nicht im Verzeichnis \MMC2 sondern unter \OEM anzulegen, damit sie nicht durch die Installation eines neuen Softwarestandes überschrieben wird.

2. Definieren der CLC-spezifischen BTSS-Variablen

In die Datei CLC.NSK ist folgende Zeile einzufügen:

```
LINK("CLC" ,200,"2 1 1 1 1F# /NC 5 0 1",100)
```

3. Anlegen bzw. Erweitern der Anwender-spezifischen Definitions-Datei: **USER.NSK**
(siehe Punkt 1.: Hinweis)

4. In der Datei USER.NSK ist der Aufruf der CLC-spezifischen Definitions-Datei CLC.NSK zu ergänzen. Dazu ist folgende Zeile einzufügen:

```
CALL (CLC .NSK)
```

LinkItem verwenden

Zur Verwendung der BTSS-Variablen in einem DDE-Control, muss die Eigenschaft "LinkItem" des DDE-Controls gemäß dem folgenden Beispiel gesetzt werden:

```
label1.LinkItem = "CLC[u1,1,9] (" "!d%15.4lf" ") "
```

Der Format-String kann gegebenenfalls angepasst werden.

Die folgenden Code-Zeilen zeigen beispielhaft die Verteilung der durch den NCDDE-Zugriff gelieferten Variablen auf ein Feld von Labels:

```
FOR i = 0 To 8  
    label2.Caption[i] = Trim$(Mid$(label1.Caption, 1+15*i, 15))  
NEXT
```

18.8 Funktionsspezifische Alarmtexte

Zum Vorgehen beim Anlegen von funktionsspezifischen Alarmtexten siehe Kapitel "Anlegen von Alarmtexten (Seite 696)".

18.9 Randbedingungen

18.9.1 Peripheriebaugruppen

Die analoge Ausgangsspannung des Abstandssensors muss zur A/D-Wandlung über eine Peripheriebaugruppe mit analogem Eingang an die NC angeschlossen werden.

Anschlussmöglichkeiten

Der Anschluss der Peripherie SIMATIC ET 200S erfolgt bei SINUMERIK 840D sl über PROFIBUS-DP. Der Abstandssensor wird über eine analoge S7 Peripheriebaugruppe angeschlossen.

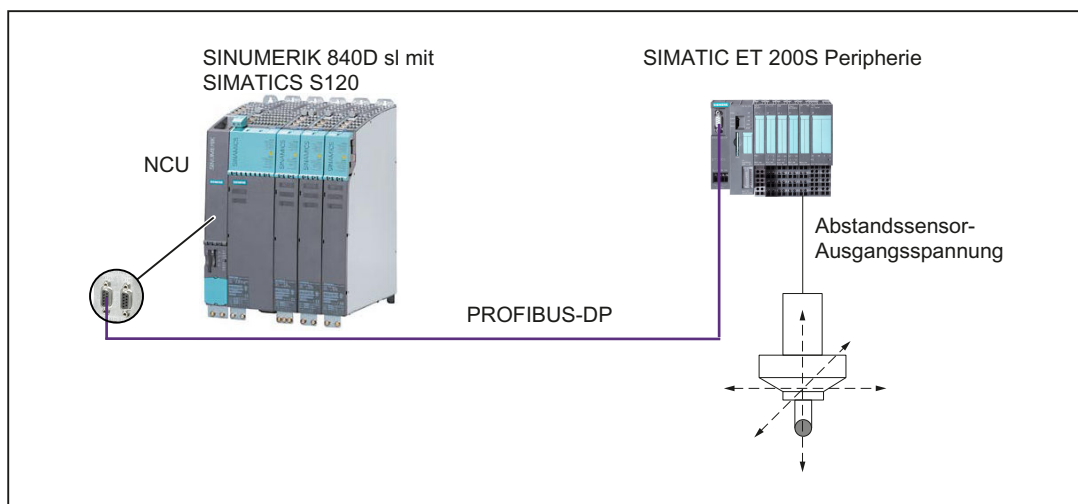


Bild 18-13 Peripheriebaugruppen-Anschluss bei SINUMERIK 840D sl

Geeignete Peripheriebaugruppen

Da die A/D-Wandlungszeit direkt in die Totzeit des Regelkreises der Abstandsregelung eingeht, darf nur eine Peripheriebaugruppe mit kleiner Wandlungszeit verwendet werden.

Für die Abstandsregelung geeignete SIMATIC S7 Peripheriebaugruppen sind:

- Analoges Peripheriemodul 2 AI, U, High Speed für ET 200S
- Analoges Peripheriemodul 2 AO, U, High Speed für ET 200S

Anschluss der Peripheriebaugruppen

Die SIMATIC Peripheriegeräte der Baureihe ET200, z. B. ET200M, werden mittels HW-Konfig wie üblich in das S7-Projekt eingebracht und konfiguriert.

Hinweis

Zur Überprüfung, ob eine im Hardware-Katalog angewählte Baugruppe mit der in der Automatisierungsanlage vorhandenen Baugruppe übereinstimmt, wird folgendes Vorgehen empfohlen:

1. Notieren Sie sich die Artikelnummern aller in der Automatisierungsanlage verwendeten Baugruppen.
 2. Wählen Sie im Hardware-Katalog die entsprechende Baugruppe an und vergleichen sie die Artikelnummer der in der Automatisierungsanlage verwendeten Baugruppe mit der Artikelnummer, die im Hardware-Katalog angezeigt wird. Beide Artikelnummern müssen übereinstimmen.
-

18.9.1.1 Externe Glättungsfilter

Soll zur Glättung der Ausgangsspannung des Abstandssensors vor der A/D-Wandlung der Ausgangsspannung durch die Peripheriebaugruppe ein externer Filter zwischengeschaltet werden, ist dabei sicherzustellen, dass die resultierende Zeitkonstante klein gegenüber dem Lagereglertakt der NC ist.

Hinweis

Es ist regelungstechnisch günstiger ein großes Signal-Rausch-Verhältnis über elektromagnetische Abschirmmaßnahmen zu erzielen, als durch die Verwendung von Glättungsfiltern im Signalpfad.

18.9.2 Funktionsspezifische Randbedingungen

Vollständiger NC-Stop

Soll im Zusammenhang mit NC-Stop nicht nur die programmierte Bahnbewegung, sondern auch die Verfahrbewegung der abstandsgeregelten Achsen gestoppt werden, sind dazu folgende NC/PLC-Nahtstellensignale zu setzen:

- DB21, ... DBX7.3 = 1 (NC-Stop)
- DB21, ... DBX7.4 = 1 (NC-Stop Achsen und Spindeln)

Nachführen

Wird eine abstandsgeregelte Achse als Alarmreaktion oder über das entsprechende NC/PLC-Nahtstellensignal in den Zustand "Nachführen" geschaltet, erfolgt auch keine Sollwertausgabe seitens der Abstandsregelung mehr an diese Achse.

Fahren ohne Software-Endschalter

Sollen die abstandsgeregelten Achsen unreferenziert verfahren werden (Fahren ohne Software-Endschalter), müssen dennoch für die achsspezifischen Software-Endschalter Werte außerhalb des benutzten Verfahrbereichs parametrisiert werden:

- MD36100 \$MA_POS_LIMIT_MINUS (1. Software-Endschalter minus)
- MD36110 \$MA_POS_LIMIT_PLUS (1. Software-Endschalter plus)
- MD36120 \$MA_POS_LIMIT_MINUS2 (2. Software-Endschalter minus)
- MD36130 \$MA_POS_LIMIT_PLUS2 (2. Software-Endschalter plus)

Die Abstandsregelung berücksichtigt die Maschinendaten auch im unreferenzierten Zustand einer Achse.

Digital-/Analog-Eingänge sperren

Sowohl der Analogeingang für die Eingangsspannung des Abstandssensors als auch der Digitaleingang den die Abstandsregelung im Rahmen der Sonderfunktion "Schnellabheben im Lagereglertakt" verwendet, sind von der PLC aus nicht beeinflussbar (sperrbar):

DB10, DBB0 (Sperrung der digitalen NC-Eingänge)

DB10, DBB146 (Sperrung der analogen NC-Eingänge)

Siehe dazu auch die Beschreibung zum Maschinendatum:

- MD62508 \$MC_CLC_SPECIAL_FEATURE_MASK, Bit 4 und 5 (Spezielle Funktionen und Betriebsarten der Abstandsregelung)

Gantry-Achsen: Nur Leitachsen

Nur eine der abstandsgeregelten Achsen darf als Leitachse eines Gantry-Verbunds konfiguriert sein:

MD37100 \$MA_GANTRY_AXIS_TYPE (Gantry-Achsdefinition)

Die Verwendung von Folgeachsen eines Gantry-Verbunds ist nicht zulässig.

Anzeige der Achspositionen

Die tatsächliche aktuelle Achsposition einer abstandsgeregelten Achse als Summe aus interpolatorischer Achsposition und dem aktuellen Positionsoffset der Abstandsregelung wird im Maschinen-Grundbild nicht angezeigt:

- SINUMERIK HMI Advanced:
Die tatsächliche aktuelle Achsposition wird im Service-Bild: **Bedienbereichsumschaltung > Diagnose > Service Anzeigen > Achse/Spindel** als "Lageistwert" angezeigt.
- SINUMERIK Operate:
Die tatsächliche aktuelle Achsposition wird im Service-Bild: **Bedienbereichsumschaltung > Diagnose > Achsdialog > Service Achse** > als "Lageistwert-Messsystem 1 und 2" angezeigt.

Keine virtuellen Achsen

Abstandsgeregelte Achsen dürfen nicht als virtuellen Achsen parametrieren werden:

MD30132 \$MA_IS_VIRTUAL_AX[<Achse>] (Achse ist virtuelle Achse)

Rechenzeitbedarf

An Steuerungen, bei denen die parametrieren Taktzeiten von Interpolations- und Lageregeltakt gegenüber der jeweiligen Standardeinstellung stark optimiert wurde, ist der zusätzliche Rechenzeitbedarf der Technologiefunktion "Abstandsregelung" zu berücksichtigen.

Der zusätzliche Rechenzeitbedarf entsteht ab der Aktivierung der Abstandsregelung im Teileprogramm `CLC (. . .)`. Bei Überschreitung des Interpolations- oder Lageregeltaktes wird folgender Alarm angezeigt:

- Alarm: "4240 Rechenzeitüberlauf auf der IPO- oder Lagereglerebene, IP
Teileprogrammstelle"

Die Bearbeitung des Teileprogramms wird abgebrochen.

1D-Abstandsregelung

In folgender Situation kann der Alarm "1016: Systemfehler, ID550010" auftreten:

- Die abstandsgeregelte Achse (z.B. Z-Achse) ist als Geometrieachse parametrieren
- Innerhalb einer beliebigen Befehlsfolge in der implizit oder explizit `STOPRE` ausgelöst wird, wird die Abstandsregelung mit `CLC(0)` ausgeschaltet

Es wird daher empfohlen, die abstandsgeregelte Achse einer 1D-Abstandsregelung (z.B. Z-Achse) so zu parametrieren, dass sie keine Geometrieachse des Kanals mehr ist.

Parametrierung: Maschinendaten

- MD20050 \$MC_AXCONF_GEOAX_ASSIGN_TAB[<Z-Achse>] = 0
- MD20060 \$MC_AXCONF_GEOAX_NAME_TAB[<Z-Achse>] = "NO_Z_AXIS"

Programmierung: Drehungen um die Z-Achse

Da die Z-Achse nach der Umparametrierung keine Geometrieachse mehr ist, muss für Drehungen um die Z-Achse statt der vordefinierten Funktion CROT () die vordefinierte Prozedur CRPL () verwendet werden:

CROT (Z, <Winkel>) → CRPL (1, <Winkel>)

18.10 Datenlisten**18.10.1 Maschinendaten****18.10.1.1 NC-spezifischen Maschinendaten**

Nummer	Bezeichner: \$MN_	Beschreibung
10300	FASTIO_ANA_NUM_INPUTS	Anzahl der aktiven analogen NC-Eingänge
10350	FASTIO_DIG_NUM_INPUTS	Anzahl der aktiven digitalen NC-Eingangsbytes
10362	HW_ASSIGN_ANA_FASTIN	Hardware-Zuordnung der externen analogen NC-Eingänge: 0...7
10712	NC_USER_CODE_CONF_NAME_TAB	Liste der umbenannten NC-Bezeichner

18.10.1.2 Kanal-spezifische Maschinendaten

Nummer	Bezeichner: \$MC_	Beschreibung
28090	MM_NUM_CC_BLOCK_ELEMENTS	Anzahl der Compile-Zyklen-Satzelemente (DRAM)
28100	MM_NUM_CC_BLOCK_USER_MEM	Speicherplatz für Compile-Zyklen-Satzelemente (DRAM) in kB
28254	MM_NUM_AC_PARAM	Anzahl Parameter für Synchronaktionen

Abstandsregelung

62500	CLC_AXNO	Achszuordnung für die Abstandsregelung
62502	CLC_ANALOG_IN	Analogeingang für die Anstandsregelung
62504	CLC_SENSOR_TOUCHED_INPUT	Zuordnung eines Eingangsbits für das Signal "Sensor-Kollision"
62505	CLC_SENSOR_LOWER_LIMIT	Untere Bewegungsgrenze der Abstandsregelung
62506	CLC_SENSOR_UPPER_LIMIT	Obere Bewegungsgrenze der Abstandsregelung
62508	CLC_SPECIAL_FEATURE_MASK	Spezielle Funktionen und Betriebsarten der Abstandsregelung

Nummer	Bezeichner: \$MC_	Beschreibung
62510	CLC_SENSOR_VOLTABE_TABLE_1	Koordinate Spannung der Stützpunkte Sensorkennlinie 1
62511	CLC_SENSOR_VELO_TABLE_1	Koordinate Geschwindigkeit der Stützpunkte Sensorkennlinie 1
62512	CLC_SENSOR_VOLTAGE_TABLE_2	Koordinate Spannung der Stützpunkte Sensorkennlinie 2
62513	CLC_SENSOR_VELO_TABLE_2	Koordinate Geschwindigkeit der Stützpunkte Sensorkennlinie 2
62516	CLC_SENSOR_VELO_LIMIT	Geschwindigkeit der Abstandsregelbewegung
62516	CLC_SENSOR_ACCEL_LIMIT	Beschleunigung der Abstandsregelbewegung
62520	CLC_SENSOR_STOP_POS_TOL	Positionstoleranz für die Zustandsmeldung "Stillstand Abstandsregelung"
62521	CLC_SENSOR_STOP_DWELL_TIME	Wartezeit für die Zustandsmeldung "Stillstand Abstandsregelung"
62522	CLC_OFFSET_ASSIGN_ANAOUT	Änderung des Soll-Abstands durch Überlagerung des Sensorsignals
62523	CLC_LOCK_DIR_ASSIGN_DIGOUT	Zuordnung der Digitalausgänge für Verriegelung der CLC-Bewegung
62524	CLC_ACTIVE_AFTER_RESET	Abstandsregelung bleibt nach RESET weiterhin aktiv
62525	CLC_SENSOR_FILTER_TIME	Zeitkonstante der PT1-Filterung des Sensorsignals
62528	CLC_PROG_ORI_AX_MASK	Progr. Orientierungsvektor: Achsmaske
62529	CLC_PROG_ORI_MAX_ANGLE	Progr. Orientierungsvektor: Maximaler Differenzwinkel
62530	CLC_PROG_ORI	Progr. Orientierungsvektor: Index der \$AC_PARAM-Variablen zur Ausgabe des aktuellen Differenzwinkels

18.10.1.3 Achs-/Spindel-spezifische Maschinendaten

Nummer	Bezeichner: \$MA_	Beschreibung
32070	CORR_VELO	Achsgeschwindigkeit für Handrad, externe Nullpunktverschiebung, SA-Abstandsregelung
32410	AX_JERK_TIME	Zeitkonstante für axiales Rückfilter
32610	VELO_FFW_WEIGHT	Vorsteuerfaktor für die Geschwindigkeitsvorsteuerung
36000	STOP_LIMIT_COARSE	Genauhalt grob
36010	STOP_LIMIT_FINE	Genauhalt fein
36040	STANDSTILL_DELAY_TIME	Verzögerungszeit Stillstandsüberwachung
36060	STANDSTILL_VELO_TOL	Geschwindigkeit für Achse/Spindel gestoppt
36750	AA_OFF_MODE	Art der Werte-Verrechnung bei axialer Positionsüberlagerung

18.10.2 Parameter Antrieb (SINAMICS S120)

Nummer	Kurzname	Langname
p1414[0...n]	n_soll_filt Akt	Drehzahlsollwertfilter Aktivierung 1, 2
p1415[0...n]	n_soll_filt 1 Typ	Drehzahlsollwertfilter 1 Typ
p1416[0...n]	n_soll_filt 1 T	Drehzahlsollwertfilter 1 Zeitkonstante
p1417[0...n]	n_soll_filt 1 fn_n	Drehzahlsollwertfilter 1 Nenner-Eigenfrequenz
p1418[0...n]	n_soll_filt 1 D_n	Drehzahlsollwertfilter 1 Nenner-Dämpfung
p1419[0...n]	n_soll_filt 1 fn_z	Drehzahlsollwertfilter 1 Zähler-Eigenfrequenz
p1420[0...n]	n_soll_filt 1 D_z	Drehzahlsollwertfilter 1 Zähler-Dämpfung
p1421[0...n]	n_soll_filt 2 Typ	Drehzahlsollwertfilter 2 Typ
p1422[0...n]	n_soll_filt 2 T	Drehzahlsollwertfilter 2 Zeitkonstante
p1423[0...n]	n_soll_filt 2 fn_n	Drehzahlsollwertfilter 2 Nenner-Eigenfrequenz
p1424[0...n]	n_soll_filt 2 D_n	Drehzahlsollwertfilter 2 Nenner-Dämpfung
p1425[0...n]	n_soll_filt 2 fn_z	Drehzahlsollwertfilter 2 Zähler-Eigenfrequenz
p1426[0...n]	n_soll_filt 2 D_z	Drehzahlsollwertfilter 2 Zähler-Dämpfung

18.10.3 Signale

18.10.3.1 Signale an Kanal

DB-Nummer	Byte.Bit	Beschreibung
21, ...	1.4	Stopp CLC-Bewegung
21, ...	1.5	Feedrate-Override wirkt auf CLC

18.10.3.2 Signale von Kanal

DB-Nummer	Byte.Bit	Beschreibung
21, ...	37.3	CLC ist aktiv
21, ...	37.4-5	CLC-Bewegung ist im Stillstand
21, ...	37.4	CLC-Bewegung an unterer Bewegungsgrenze
21, ...	37.5	CLC-Bewegung an oberer Bewegungsgrenze

19.1 Kurzbeschreibung

Eine Master-Slave-Kopplung ist eine auf Lageregelebene durchgeführte Drehzollsollwertkopplung zwischen einer Master- und einer beliebigen Anzahl von Slave-Achsen mit und ohne Momentenausgleichsregelung. Die Kopplung kann statisch, d.h. permanent eingeschaltet, dynamisch ein-/ausgeschaltet und umkonfiguriert werden.

Hinweis

Für SINUMERIK 828D bestehen bezüglich der Funktion "Master-Slave-Kopplung" folgende Einschränkungen:

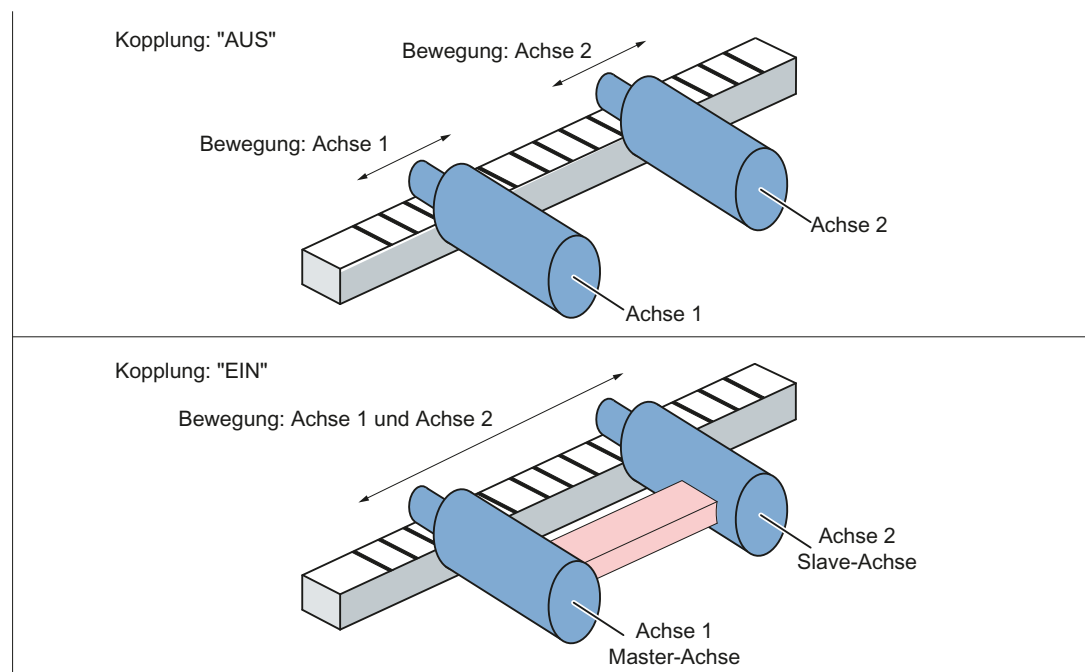
- Es ist nur **ein** Master-Slave-Verbund möglich (Ausnahme: Steuerungsvariante "SW28x(1) CNC-SW Fräsen Adv. Export (me822) mit max. **zwei** Master-Slave-Verbänden).
- Es kann nur **eine** Slave-Achse an die Master-Achse gekoppelt werden.

Anwendungen

Mögliche Anwendungen der Master-Slave-Kopplung sind:

- Leistungsverstärkung bei mechanisch gekoppelten Antrieben
- Kompensation von Getriebe- und Zahnflankenspiel durch Aufschalten eines Verspannmoments

Dynamische Master-Slave-Kopplung zweier Achsen



19.2 Kopplungsschaltbild

Bei geschlossener Kopplung wird die Slaveachse ausschließlich über den lastseitigen Drehzahlsollwert der Masterachse verfahren. Sie ist damit nur drehzahl- und nicht lagegeregelt. Zwischen Master- und Slaveachse erfolgt auch keine Differenzlageregung.

Über den Momentenausgleichsregler wird das geforderte Moment zwischen der Master- und der Slaveachse aufgeteilt. Die Aufteilung kann bei Verwendung unterschiedlicher Motoren über einen Wichtungsfaktor den spezifischen Anforderungen angepasst werden.

Durch Aufschalten eines Zusatzmoments (Verspannmoment) kann eine Verspannung zwischen Master- und Slaveachse erreicht werden.

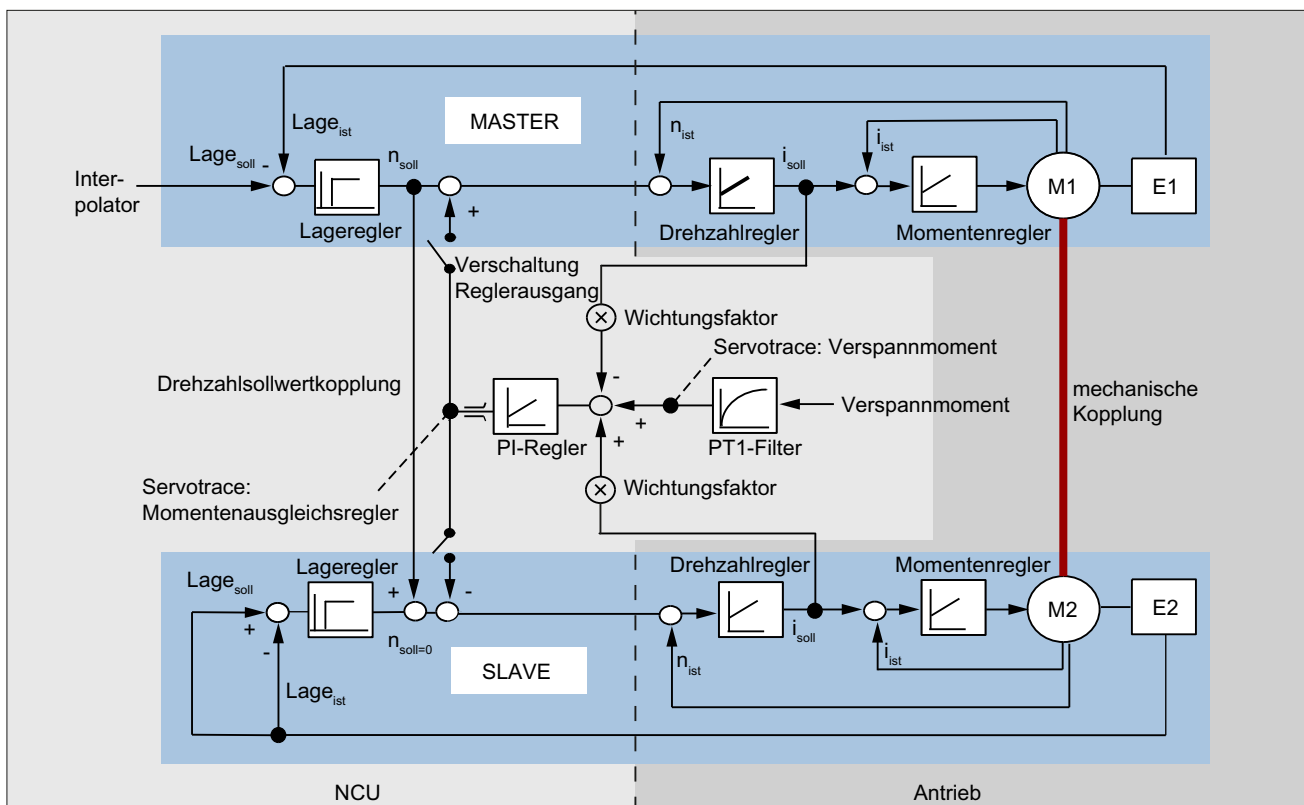


Bild 19-1 Reglerstruktur

19.3 Konfiguration einer Kopplung

Statische Zuordnung

Die statische Zuordnung von Master- und Slaveachse wird für Drehzahlsollwertkopplung und Momentenausgleichsregelung getrennt in folgenden Maschinendaten definiert:

- Drehzahlsollwertkopplung
MD37250 \$MA_MS_ASSIGN_MASTER_SPEED_CMD[<Slaveachse>] =
<Maschinenachsnnummer der Masterachse für Drehzahlsollwertkopplung>
- Momentenausgleichsregelung
MD37252 \$MA_MS_ASSIGN_MASTER_TORQUE_CTR[<Slaveachse>] =
<Maschinenachsnnummer der Master- oder Slaveachse für Momentenausgleichsregelung>
(Siehe Kapitel "Verspannmoment (Seite 752)")

Dynamische Zuordnung (nur 840D sl)

Mit den nachfolgenden Programmbefehlen kann die Zuordnung von Master- und Slaveachsen dynamisch verändert werden.

Zuordnung definieren

Zuordnung einer oder mehrerer Slaveachsen zu einer Masterachse:

```
MASLDEF (<Slaveachse_1>, <Slaveachse_2>, ..., <Masterachse>)
```

Die Anzahl der zuordenbaren Slaveachsen ist prinzipiell beliebig. Die Zuordnung bleibt auch nach Betriebsartenwechsel, Reset und Teileprogrammende erhalten.

Zuordnung löschen

Zuordnung einer oder mehrerer Slaveachsen zu ihrer jeweiligen Masterachse löschen:

```
MASLDEL (<Slaveachse_1>, <Slaveachse_2>, ...)
```

Vor dem Löschen der Zuordnung wird eine aktive Kopplung zuerst implizit ausgeschaltet.

Achszuordnung für Drehzahlsollwertkopplung und Momentenausgleichsregelung

Für die Drehzahlsollwertkopplung bezieht sich die Slaveachse auf die bei der Definition der Zuordnung (MASLDEF) angegebene Masterachse.

Auf welche Achse sich die Slaveachse für die Momentenausgleichsregelung beziehen soll, wird in folgendem Maschinendatum eingestellt:

```
MD37253 $MA_MS_FUNCTION_MASK[<Slaveachse>], Bit 1 = <Wert>
```

<Wert>	Beschreibung
0	Bei dynamischer Zuordnung (MASLDEF) bezieht sich die Slaveachse für die Momentenausgleichsregelung, ebenso wie für die Drehzahlsollwertkopplung, auf die Masterachse des Master-Slave-Verbunds.
1	Bei dynamischer Zuordnung (MASLDEF) bezieht sich die Slaveachse für die Momentenausgleichsregelung auf die im Maschinendatum angegebene Achse, Master- oder Slaveachse, des Master-Slave-Verbunds: MD37252 \$MA_MS_ASSIGN_MASTER_TORQUE_CTR<Slaveachse> = <Achse>

(Siehe Kapitel "Verspannmoment (Seite 752)")

Randbedingungen

Bei der dynamischen Zuordnung sind folgende Randbedingungen zu beachten:

- Eine Änderung der Zuordnung mit `MASLDEF` hat im eingeschalteten Zustand der Kopplung keine Auswirkung. Die Änderung wird erst mit dem nächsten Ausschalten der Kopplung wirksam.
- Für Drehzahlollwertkopplung und Momentenausgleichsregelung wird immer die gleiche Masterachse verwendet.
- Eine Plausibilitätskontrolle der Zuordnungen erfolgt erst beim Einschalten der Kopplung.

Anwenderspezifische Standard-Zuordnung nach Reset (nur 840D sl)

Eine anwenderspezifische Standard-Zuordnung, die immer nach Reset wirksam wird, kann im `PROG_EVENT.SPF` durch die Programmbefehle `MASLDEF` und `MASLDEL` definiert werden.

Damit `PROG_EVENT.SPF` bei Reset ausgeführt wird, ist folgendes Maschinendatum einzustellen: `MD20108 $MC_PROG_EVENT_MASK.Bit 2 = 1`

Beispiel: Dynamische Änderung der Zuordnung (nur 840D sl)

Die Zuordnung der Slaveachse AX3 wird von Masterachse AX1 zu Masterachse AX2 geändert. Die Kopplung muss dazu zwischenzeitlich ausgeschaltet werden (siehe Kapitel "Ein-/Ausschalten einer Kopplung (Seite 755)").

	Programmierung	Beschreibung
①	<code>MASLDEF (AX3, AX1)</code>	Zuordnung der Slaveachse AX3 zur Masterachse AX1
②	<code>MASLON (AX3)</code>	Einschalten der Kopplung
③	<code>MASLDEL (AX3)</code>	Ausschalten der Kopplung und Aufheben der Zuordnung von AX3 zu AX1
④	<code>MASLDEF (AX3, AX2)</code>	Zuordnung der Slaveachse AX3 zur Masterachse AX2

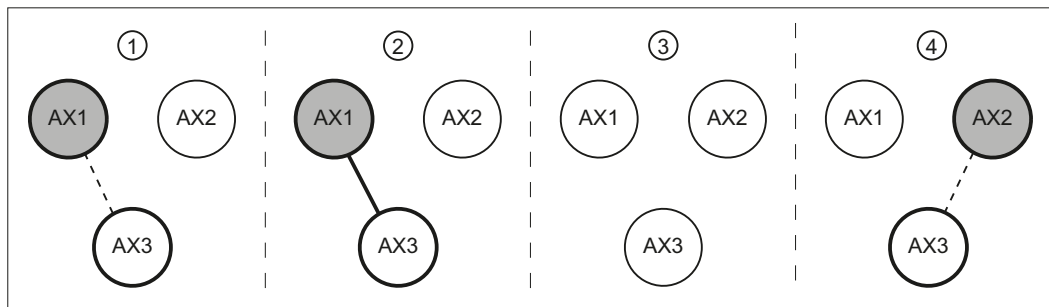


Bild 19-2 Wechselnde Zuordnung der Slaveachse

Allgemeine Randbedingungen

Folgende allgemeine Randbedingungen sind zu beachten:

- eine Slaveachse kann nur einer Masterachse zugeordnet werden
- einer Masterachse können mehrere Slaveachsen zugeordnet werden
- eine Slaveachse darf keine Masterachse einer anderen Master-Slave-Beziehung sein

Hinweis

Antrieboptimierung

An einem Antriebsgerät SINAMICS S120 können maximal 3 Antriebe gleichzeitig optimiert bzw. vermessen (Drehzahlregleroptimierung/Funktionsgenerator) werden. Es wird daher dringend empfohlen bei einer Kopplung mit mehr als 3 gleichzeitig gekoppelten Antrieben, diese auf mehrere Antriebsgeräte zu verteilen.

19.4 Momentenausgleichsregler

Der Momentenausgleichsregler (PI-Regler) berechnet aus der Momentendifferenz zwischen Master- und Slaveachse einen lastseitigen Zusatzdrehzahlsollwert. Der Zusatzdrehzahlsollwert kann über folgendes Maschinendatum unterschiedlich aufgeschaltet werden:

MD37254 \$MA_MS_TORQUE_CTRL_MODE[<Slaveachse>] = <Wert>

<Wert>	Aufschaltung des Zusatzdrehzahlsollwertes auf:
0	Master- und Slaveachse (Standardwert)
1	Slaveachse
2	Masterachse
3	nicht aufschalten

Hinweis

Mehrere Slaveachsen

Bei einem Master-Slave-Verbund mit einer Master- und mehreren Slaveachsen kann die in MD37254 standardmäßig eingestellte Aufschaltung des Zusatzdrehzahlsollwertes auf Master- und Slaveachsen zu Instabilitäten führen. Das Aufschalten des Zusatzdrehzahlsollwertes sollte bei mehreren Slaveachsen nur in den Slaveachsen vorgenommen werden:

MD37254 \$MA_MS_TORQUE_CTRL_MODE = 1

SINAMICS S120: Stromsollwertfilter

Im SINAMICS S120 können bis zu vier Stromsollwertfilter (p1656 - p1674, Funktionsplan [5710]) aktiviert werden. Es wird empfohlen, in einem Master-Slave-Verbund für die Master- und alle Slaveachsen, zu einem Zeitpunkt nur Filter mit gleich Daten zu aktivieren.

Normierung

Die Normierung der Maschinendaten für den Verstärkungsfaktor (P-Anteil) (MD37256 \$MA_MS_TORQUE_CTRL_P_GAIN) und der Drehzahlsollwert-Begrenzung (MD37260 \$MA_MS_MAX_CTRL_VELO) wird über folgendes Maschinendatum vorgegeben:

MD37253 \$MA_MS_FUNCTION_MASK[<Slaveachse>], Bit 0 = <Wert>

<Wert>	Beschreibung
0	MD37256 und MD37260 werden intern mit folgendem Faktor multipliziert: 1 / Interpolatortakt ¹⁾
1	MD37256 und MD37260 werden unverändert übernommen

1) Der Interpolatortakt wird angezeigt über MD10071 \$MN_IPO_CYCLE_TIME

Hinweis

Es wird empfohlen, MD37256 und MD37260 unverändert zu belassen:

MD37253 \$MA_MS_FUNCTION_MASK[<Slaveachse>], Bit 0 = 1

Verstärkungsfaktor (P-Anteil)

Der Verstärkungsfaktor des Momentenausgleichsreglers wird in folgendem Maschinendatum als Prozentwert des Verhältnisses der maximalen lastseitigen Achsgeschwindigkeit der Slaveachse (MD32000 \$MA_MAX_AX_VELO) zu ihrem Nennmoment (SINAMICS S120: p2003) eingestellt:

MD37256 \$MA_MS_TORQUE_CTRL_P_GAIN[<Slaveachse>]

Hinweis: Normierung über MD37253 \$MA_MS_FUNCTION_MASK[<Slaveachse>], Bit 0

Nachstellzeit (I-Anteil)

Die Nachstellzeit des Momentenausgleichsreglers wird in folgendem Maschinendatum eingestellt:

MD37258 \$MA_MS_TORQUE_CTRL_I_TIME[<Slaveachse>]

Standardmäßig ist der I-Anteil ausgeschaltet.

Drehzahlsollwert-Begrenzung

Der vom Momentenausgleichsregler berechnete Drehzahlsollwert kann über das nachfolgende Maschinendatum begrenzt werden. Die Angabe erfolgt als Prozentwert bezogen auf die maximale lastseitige Achsgeschwindigkeit der Slaveachse (MD32000 \$MA_MAX_AX_VELO).

MD37260 \$MA_MS_MAX_CTRL_VELO[<Slaveachse>]

Hinweis: Normierung über MD37253 \$MA_MS_FUNCTION_MASK[<Slaveachse>], Bit 0

Deaktivierung des Momentenausgleichsreglers

Bei folgenden Einstellungen ist der Momentenausgleichsregler inaktiv:

- MD37254 \$MA_MS_TORQUE_CTRL_MODE[<Slaveachse>] = 3
- MD37256 \$MA_MS_TORQUE_CTRL_P_GAIN[<Slaveachse>] = 0

Momentengewichtung

Über die Momentengewichtung kann der prozentuale Beitrag der Slaveachse zum Gesamtmoment eingestellt werden.

MD37268 \$MA_MS_TORQUE_WEIGHT_SLAVE[<Slaveachse>]

Durch die Momentengewichtung ist eine unterschiedliche Momentenaufteilung zwischen Master- und Slaveachse bei Motoren mit unterschiedlichen Nennmomenten realisierbar. Bei Motoren von Master- und Slaveachse mit gleichen Nennmomenten ist eine Momentenaufteilung von jeweils 50 % sinnvoll (Standardeinstellung).

Der Momentenbeitrag der Masterachse ergibt sich zu:

Momentenbeitrag der Masterachse = 100 % - MD37268

Hinweis

Mechanische Kopplung

Bei Verwendung des Momentenausgleichsreglers ist eine mechanische Kopplung zwischen Master- und Slaveachse zwingend erforderlich. Anderenfalls kann es zu unkontrollierten Beschleunigungsvorgängen der beteiligten Antriebe kommen.

Ein- / Ausschalten über NC/PLC-Nahtstelle

Das Einschalten des Momentenausgleichsreglers über die NC/PLC-Nahtstelle muss explizit freigegeben werden über:

MD37255 \$MA_MS_TORQUE_CTRL_ACTIVATION[<Slaveachse>] = 1

Einschalten

Der Momentenausgleichsregler kann achsspezifisch eingeschaltet werden über:

DB31, ... DBX24.4

Lesen des aktuellen Status

Der aktuelle Aktivierungsstatus des Momentenausgleichsreglers kann gelesen werden über:

DB31, ... DBX96.4

19.5 Verspannmoment

Das Verspannmoment ist ein Zusatzmoment, das auf den aktiven Momentenausgleichsregler aufgeschaltet wird. Dadurch wird ein mechanisches Verspannen zwischen Achsen innerhalb eines Master-Slave-Verbundes möglich. Das Verspannen ist nicht nur zwischen der Master- und einer Slaveachse möglich, sondern auch zwischen zwei Slaveachsen, indem eine der Slaveachsen zur Bezugsachse für den Momentenausgleichsregler deklariert wird.

Einstellung

Das Verspannmoment wird in Prozent des Nennmoments der Slaveachse eingegeben und ist sofort wirksam:

MD37264 \$MA_MS_TENSION_TORQUE[<Slaveachse>] = <Verspannmoment>

Wie aus der Struktur des Momentenausgleichsreglers ersichtlich (Kapitel "Momentenausgleichsregler (Seite 749)"), wird das Verspannmoment über ein PT1-Filter aufgeschaltet. Die Filterzeitkonstanten wird über folgendes Maschinendatum eingestellt:

MD37266 \$MA_MS_TENSION_TORQ_FILTER_TIME[<Slaveachse>] = <Zeitkonstante>

Ein Wert für die Filterzeitkonstanten größer 0 aktiviert den Filter.

ACHTUNG

Kein Achsausgleich

Das Aufschalten eines Verspannmoments ohne mechanische Kopplung zwischen Master- und Slaveachse führt zu Achsbewegungen.

Hinweis

Mindest-Verspannmoment

Das Verspannmoment ist so hoch zu wählen, dass das resultierende Moment auch in Beschleunigungsvorgängen die minimale erforderliche Verspannung nicht unterschreitet.

Verminderung der Motorerwärmung

Zur Verminderung der Motorerwärmung, kann das Verspannmoment für die Zeit des Stillstandes reduziert werden.

Beispiel 1 (840D sl): Statische Kopplung und paarweises Verspannen

Eine Master-Slave-Anwendung wird so parametrisiert, dass einer Masterachse drei Slaveachsen zugeordnet sind und jeweils ein Achspaar mit einem Verspannmoment beaufschlagt wird. Dadurch muss die Masterachse das Verspannmoment nicht alleine gegen alle Slaveachsen aufbringen.

Annahme bezüglich Maschinenachsen:

- 1. bis 4. Maschinenachse: AX1, AX2, AX3, AX4

Statische Kopplung für alle Slaveachsen

- MD37262 \$MA_MS_COUPLING_ALWAYS_ACTIVE[AX2] = 1
- MD37262 \$MA_MS_COUPLING_ALWAYS_ACTIVE[AX3] = 1
- MD37262 \$MA_MS_COUPLING_ALWAYS_ACTIVE[AX4] = 1

Achse	Bezugsachse der Drehzahlsollwertkopplung MD37250 = Wert		Bezugsachse der Momentenausgleichsreg. MD37252 = Wert		Aufschaltung des Momentenausgleichsreg. MD37254 = Wert	
	Wert	Beschreibung	Wert	Beschreibung	Wert	Beschreibung
AX1	0	keine Bezugsachse	0	keine Bezugsachse	0	Master und Slave
AX2	1	Masterachse, AX1	1	Masterachse, AX1	0	Master und Slave
AX3	1	Masterachse, AX1	0	keine Bezugsachse	3	keine Aufschaltung
AX4	1	Masterachse, AX1	3	2. Slaveachse, AX3	0	Master und Slave

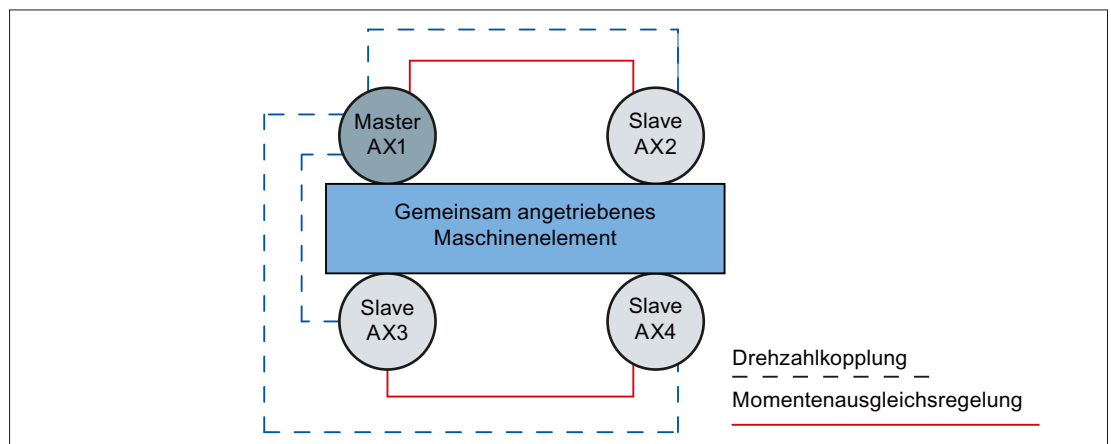


Bild 19-3 Beispiel 1: paarweise Momentenausgleichsregelung

Beispiel 2 (840D sl): Dynamische Kopplung mit 1x4 und 2x2 Achsen und paarweises Verspannen

Annahme bezüglich Maschinenachsen:

- 1. bis 4. Maschinenachse: AX1, AX2, AX3, AX4

Dynamische Kopplung für alle Slaveachsen:

- MD37262 \$MA_MS_COUPLING_ALWAYS_ACTIVE[AX2] = 0
- MD37262 \$MA_MS_COUPLING_ALWAYS_ACTIVE[AX3] = 0
- MD37262 \$MA_MS_COUPLING_ALWAYS_ACTIVE[AX4] = 0

Bei dynamischer Kopplung ist standardmäßig die bei der Definition der Kopplung (MASLDEF) angegebene Masterachse Bezugsachse für Drehzahlsollwertkopplung und Momentenausgleichsregelung.

19.5 Verspannmoment

Die Definition der Bezugsachse AX3 für die Momentenausgleichsregelung der 3. Slaveachse AX4 ist für den Anwendungsfall "1x4 Achsen" erforderlich, bei dem das Maschinendatum gesetzt wird: MD37253 \$MA_MS_FUNCTION_MASK[AX4], Bit 1 = 1 (siehe Teileprogramm)

Achse	Bezugsachse der Drehzahl- wertkopplung MD37250 = Wert		Bezugsachse der Momentenaus- gleichsreg. MD37252 = Wert		Aufschaltung des Momentenaus- gleichsreg. MD37254 = Wert	
	Wert	Beschreibung	Wert	Beschreibung	Wert	Beschreibung
AX1	0	keine Bezugsachse	0	keine Bezugsachse	0	Master und Slave
AX2	0	keine Bezugsachse	0	keine Bezugsachse	0	Master und Slave
AX3	0	keine Bezugsachse	0	keine Bezugsachse	3	keine Aufschaltung
AX4	0	keine Bezugsachse	3	Maschinenachse AX3	0	Master und Slave

Programmcode	Kommentar
PROC MASL_SWITCH IPRTLOCK DISPLOF	
IF MASL_REQUEST==4	; "1x4 Achsen" aktivieren
MASLOF(AX2, AX4)	; Kopplung ausschalten
MASLDEL(AX2, AX4)	; Kopplung löschen
\$MA_MS_FUNCTION_MASK[AX4]=	; MD37253, Bit1 = 1:
\$MA_MS_FUNCTION_MASK[AX4] B_OR 'B10'	; Bezugsachse der Momentenausgleichsreg.
	; von AX4 ist AX3 (MD37252)
NEWCONF	; Maschinendatenänderung aktivieren
STOPRE	
MASLDEF(AX2, AX3, AX4, AX1)	; Zuordnung für "1x4 Achsen" definieren
MASLON(AX2, AX3, AX4)	; Kopplungen einschalten
MASL_ACTIVE=4 MASL_REQUEST=0	; Rückmeldung: "1x4 Achsen" eingeschaltet
ENDIF	
IF MASL_REQUEST==2	; "2x2 Achsen" aktivieren
MASLOF(AX2, AX3, AX4)	; Kopplung ausschalten
MASLDEL(AX2, AX3, AX4)	; Kopplung löschen
\$MA_MS_FUNCTION_MASK[AX4]=	; MD37253, Bit1 = 0:
\$MA_MS_FUNCTION_MASK[AX4] B_AND	; Bezugsachse der Momentenausgleichsreg.
'HFFFD'	; von AX4 ist die mit MASLDEF definierte
	; Masterachse
NEWCONF	
STOPRE	
MASLDEF(AX2, AX1)	; Zuordnung für 1. "2x2 Achsen" definieren
MASLDEF(AX4, AX3)	; Zuordnung für 2. "2x2 Achsen" definieren
MASLON(AX2, AX4)	; Kopplungen einschalten
MASL_ACTIVE=2 MASL_REQUEST=0	; Rückmeldung: "2x2 Achsen" eingeschaltet
ENDIF	
RET	

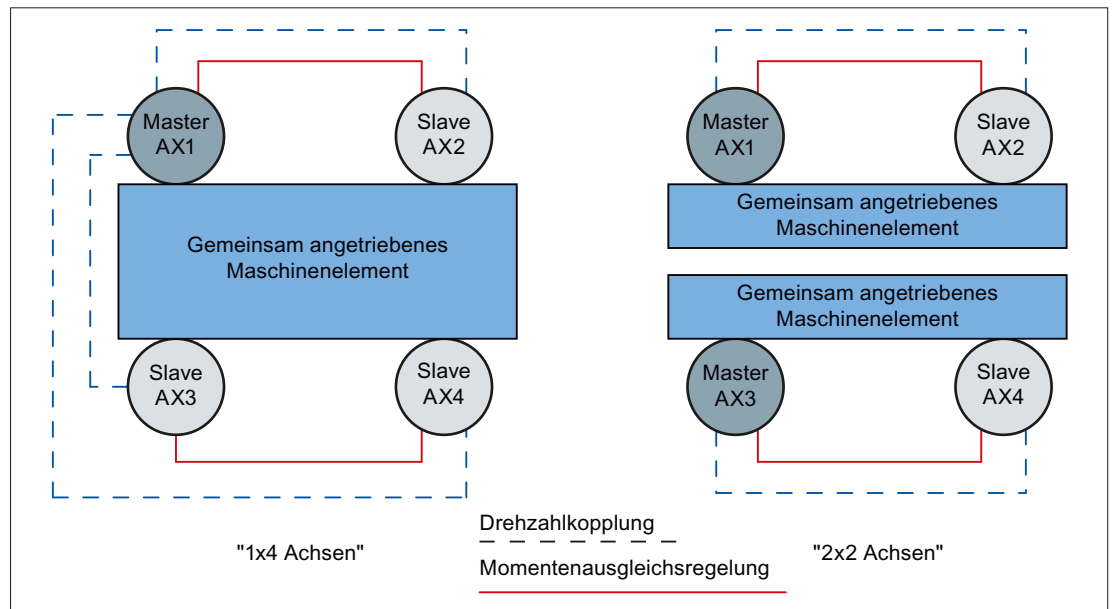


Bild 19-4 Beispiel 2: wechselseitige Kopplung mit 1x4 und 2x2 Achsen

19.6 Ein-/Ausschalten einer Kopplung

Voreinstellung

Über folgendes Maschinendatum wird festgelegt, ob die Kopplung nach dem Hochlauf der Steuerung permanent eingeschaltet wird (statisch) oder dynamisch ein-/ausgeschaltet und umkonfiguriert werden kann:

MD37262 \$MA_MS_COUPLING_ALWAYS_ACTIVE[<Slaveachse>] = <Einschaltmode>

Statisches Einschalten einer Kopplung

MD37262 \$MA_MS_COUPLING_ALWAYS_ACTIVE[<Slaveachse>] = 1

Die Kopplung wird nach dem Hochlauf der Steuerung statisch eingeschaltet.

Durch Schreiben des Maschinendatums im Teileprogramm oder Synchronaktion mit anschließendem Warmstart `NEWCONF`, kann auch eine statische Kopplung zu einem späteren Zeitpunkt aus- und eingeschaltet werden.

Beispiel: Aus- und Einschalten der statischen Kopplung für Slaveachse AX2

Programmcode	Kommentar
N100 \$MA_MS_COUPLING_ALWAYS_ACTIVE[AX2] = 0	; Kopplungsart: statisch -> dynamisch
N110 NEWCONF	; Maschinendatum aktivieren.
N120 MASLOF(AX2)	; Kopplung ausschalten.
....	
N200 PRESETON(AX2, ...)	; z.B.: Istwertsetzen der Slaveachse.
....	

Programmcode	Kommentar
N300 \$MA_MS_COUPLING_ALWAYS_ACTIVE[AX2] = 1	; Kopplungsart: dynamisch -> statisch, Kopplung einschalten.

Hinweis

Eine statisch eingeschaltete Kopplung kann durch die Master-Slave-spezifischen NC/PLC-Nahtstellensignale und/oder Programmbefehle weder ein-/ausgeschaltet noch umkonfiguriert werden.

Dynamisches Ein-/Ausschalten einer Kopplung

MD37262 \$MA_MS_COUPLING_ALWAYS_ACTIVE[<Slaveachse>] = 0

Die Kopplung kann dynamisch ein- und ausgeschaltet und umkonfiguriert werden.

Das Ein-/Ausschalten der Kopplung kann erfolgen durch:

- Maschinendatum MD37262
Schreiben des Maschinendatums zum Einschalten (= 1) bzw. Ausschalten (= 0) im Teileprogramm oder Synchronaktion. Die Änderung wird sofort aktiv.
- NC/PLC-Nahtstelle
DB31, ... DBX24.7 = 1 bzw. 0 (Master-Slave Ein bzw. Aus)
- Programmbefehle (nur 840D sl)
 - Einschalten: MASLON (<Slaveachse_1>, <Slaveachse_2>, ...)
 - Ausschalten: MASLOF (<Slaveachse_1>, <Slaveachse_2>, ...)
 - Ausschalten mit Abbremsen der Slavespindeln:
MASLOFS (<Slavespindel_1>, < Slavespindel _2>, ...)
 - Ausschalten mit Löschen der Zuordnung:
MASLDEL (<Slaveachse_1>, <Slaveachse_2>, ...)

Hinweis**Ändern der statischen Zuordnung**

Die Einstellung:

MD37262 \$MA_MS_COUPLING_ALWAYS_ACTIVE[<Slaveachse>] = 0

erlaubt ein dynamisches Ändern der Kopplung. Damit ist es möglich durch MASLDEF eine neue Zuordnung zu definieren. Die in den Maschinendaten parametrisierte Zuordnung (siehe Statische Zuordnung, Kapitel "Konfiguration einer Kopplung (Seite 747)") wird erst wieder nach einem erneuten Hochlauf der Steuerung (NC-Reset) wirksam.

Kopplungszustand**Systemvariable**

Der aktuelle Kopplungszustand einer Slaveachse kann im Teileprogramm und Synchronaktion über folgende Systemvariable gelesen werden:

\$AA_MASL_STAT[<Slaveachse>]

Wert	Beschreibung
0	1) Die Kopplung der Slaveachse ist nicht aktiv. 2) Die angegebene Achse ist keine Slaveachse
> 0	Die Kopplung ist aktiv. <Wert> == Maschinenachsnummer der Masterachse

NC/PLC-Nahtstellensignal

Der aktuelle Kopplungszustand einer Slaveachse kann über folgendes achsspezifische NC/PLC-Nahtstellensignal gelesen werden:

DB31, ... DBX96.7 (Master-Slave-Kopplung aktiv)

Sollzustand

Zeitlich aufeinander folgende Anforderungen zum Ein-/Ausschalten einer Master-Slave-Kopplung (NC/PLC-Nahtstellensignal und/oder Programmbefehle) überschreiben sich gegenseitig. Der wirksame Sollzustand ergibt sich aus der zeitlich letzten Anforderung (siehe Kapitel "Ein-/Ausschaltverhalten (Seite 757)").

19.7 Ein-/Ausschaltverhalten

Ein-/Ausschalten im Stillstand (Achse)

Eine Anforderung zum Ein-/Ausschalten der Kopplung wird erst dann wirksam, wenn Master- und Slaveachse im Stillstand sind:

DB31, ... DBX61.4 == 1 (Achse/Spindel steht)

Die gekoppelten Achsen müssen dabei in Regelung sein.

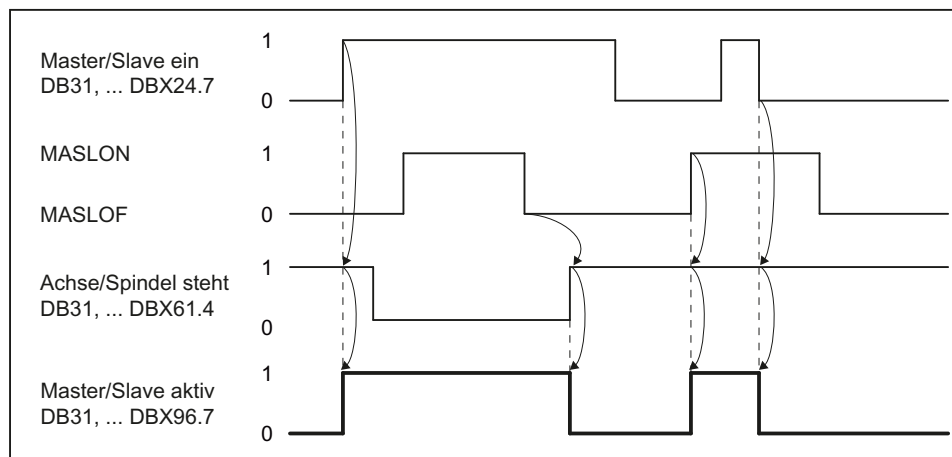


Bild 19-5 Aktivierungsvorgaben

Beim Einschalten der Kopplung mit dem Programmbefehl `MASLON`, wird mit dem Satzwechsel so lange gewartet, bis die Kopplung geschlossen wurde. An der Bedienoberfläche wird so lange die Meldung "Master-Slave-Umschaltung aktiv" angezeigt.

Ein-/Ausschalten in der Bewegung (Spindel)

Hinweis

Ein-/Ausschalten in der Bewegung

Während der Bewegung kann nur bei **Spindeln** im **Drehzahlsteuerbetrieb** die Kopplung ein- bzw. ausgeschaltet werden.

Einschalten

Beim Einschalten während der Bewegung, teilt sich der Koppelvorgang bei unterschiedlichen Drehzahlen in zwei Phasen.

- Phase 1
 - Das Einschalten der Kopplung muss im PLC-Anwenderprogramm angefordert werden mit:
DB31, ... DBX24.7 = 1 (Master/Slave Ein)
 - Die Slavespindel beschleunigt oder bremst rampenförmig auf die Solldrehzahl der Masterspindel.
 - Beim Erreichen der Solldrehzahl wird die Kopplung eingeschaltet und von der NC das NC/PLC-Nahtstellensignal gesetzt:
DB31, ... DBX96.7 == 1 (Kopplung aktiv)
 - Wird während des Einschaltvorgangs die Masterspindel beschleunigt, so verlängert sich Phase 1 entsprechend der Dynamikdifferenz von Master- und Slavespindel.
- Phase 2
 - Aus der Ist-Differenzdrehzahl zwischen Master- und Slavespindel(n) werden die folgenden Synchronlaufsignale erzeugt:
 - DB31, ... DBX96.3 (Drehzahltoleranz grob)
 - DB31, ... DBX96.2 (Drehzahltoleranz fein)
Die zugehörigen Grenzwerte werden über folgende Maschinendaten eingestellt:
 - MD37270 \$MA_MS_VELO_TOL_COARSE ("Toleranz grob")

- MD37272 \$MA_MS_VELO_TOL_FINE ("Toleranz fein").

Hinweis

Mit dem Signal "Drehzahltoleranz grob" kann eine PLC-seitige Überwachung realisiert werden, die einen gekoppelten Master-Slave-Verbund auf den Verlust der Drehzahl synchronität überprüft.

Aus dem Signal "Drehzahltoleranz fein" kann direkt der Zeitpunkt zum mechanischen Schließen der Kopplung und zum Einschalten des Momentenausgleichsreglers abgeleitet werden.

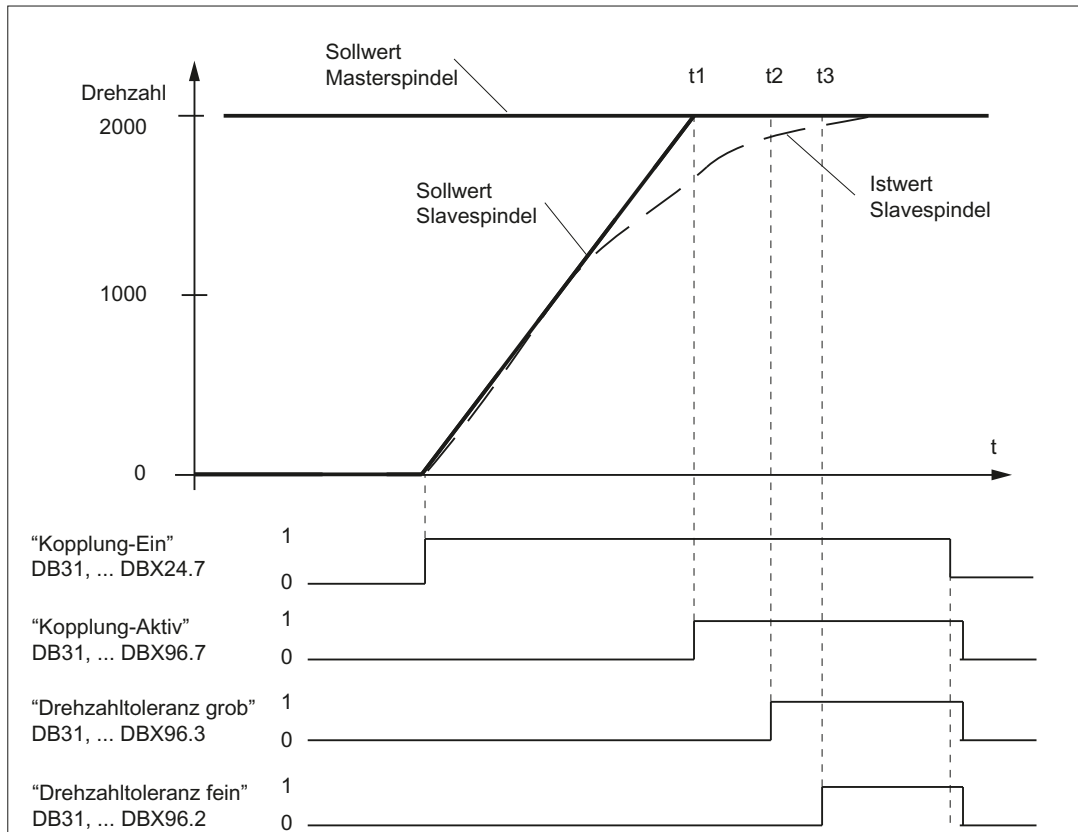


Bild 19-6 Koppelvorgang bei unterschiedlichen Drehzahlen

Ausschalten ohne Bremsen

Wird die Kopplung mit dem Programmbefehl MASLOF ausgeschaltet, wird bei Spindeln im Drehzahlsteuerbetrieb die Kopplung sofort ausgeschaltet. Die Slavespindeln behalten ihre zum Ausschaltzeitpunkt aktuellen Drehzahlen bis zur einer erneuten Drehzahlprogrammierung bei.

Ausschalten mit Bremsen

Wird die Kopplung mit dem Programmbefehl `MASLOFS` ausgeschaltet wird bei Spindeln im Drehzahlsteuerbetrieb die Kopplung sofort ausgeschaltet und die Slavespindeln abgebremst.

Hinweis

Bei `MASLON` und `MASLOF` entfällt der implizite Vorlaufstopp. Bedingt durch den fehlenden Vorlaufstopp liefern die \$P-Systemvariablen der Slavespindeln bis zum Zeitpunkt erneuter Programmierung keine aktualisierten Werte.

Koppelverhalten

Das Verhalten bezüglich der Programmbefehle `MASLON`, `MASLOF`, `MASLOFS`, `MASLDEL` (nur 840D sl) und des NC/PLC-Nahtstellensignals: DB31, ... DBX24.7 (Master/Slave Ein) wird bei Spindeln im Drehzahlsteuerbetrieb über folgendes Maschinendatum eingestellt:

MD37263 \$MA_MS_SPIND_COUPLING_MODE[<Slavespindel>] = <Wert>

<Wert>	Beschreibung
0	Das Koppeln und Trennen erfolgt ausschließlich im Stillstand. Der aktuelle Kopplungszustand bleibt bis zum tatsächlichen Stillstand aller beteiligten Spindeln erhalten. Die <code>MASLOFS</code> Anweisung ist der <code>MASLOF</code> gleichwertig, ein automatisches Abbremsen der Slavespindel entfällt.
1	Koppeln und Trennen erfolgt unmittelbar und damit auch in der Bewegung. Beim Koppeln werden die Slavespindeln automatisch auf die aktuelle Drehzahl der Master- spindel beschleunigt. Beim Trennen behalten die zu diesem Zeitpunkt drehenden Slavespindeln ihre Drehzahlen bis zur erneuten Drehzahlprogrammierung bei. Dagegen bremst eine mit <code>MASLOFS</code> getrennte Slavespindel automatisch ab.

19.8 Randbedingungen**19.8.1 Funktionale Randbedingungen****Allgemein**

- Master- und Slaveachse müssen auf derselben NCU liegen.
- Eine Kopplung wird unabhängig vom Kanalzustand im nächsten Achsstillstand ein- bzw. ausgeschaltet.
- Slaveachsen können bei geschlossener Kopplung nur über die Masterachse verfahren werden.
- Eine gekoppelte Slaveachse kann nicht über Achscontainer weitergeschaltet werden.
- Ein angeforderter Achstausch von gekoppelten Slaveachsen wird nicht durchgeführt.

- Beim Einschalten der Kopplung über die Slaveachse, wird die Masterachse, falls sie Kanalachse im gleichen Kanal ist, automatisch abgebremst: ⇒ Asymmetrisches Verhalten beim Ein- und Ausschalten der Kopplung:
 - Einschalten: automatisches Abbremsen der Masterachse
 - Ausschalten: **kein** automatisches Abbremsen der Masterachse
- Die Sollposition einer gekoppelten Slaveachse entspricht ihrer aktuellen Istposition.
- Bei Achsen und Spindeln im Positionierbetrieb wird die Kopplung nur im Stillstand ein- und ausgeschaltet.
- Vor einem Getriebewechsel oder einer Stern-Dreieck-Umschaltung, muss eine Master-Slave-Kopplung ausgeschaltet werden.

Axiale Überwachungen

- Bis auf die Drehzahlsollwert- und Istgeschwindigkeitsüberwachung sind axiale Überwachungen wie die Kontur- und die Stillstandsüberwachung in der Slaveachse wegen des fehlenden Lagereglers inaktiv. Die Lageregelkreisparameter wie Verstärkungsfaktor, Vorsteuerung, Symmetrierung können damit in der Master- und Slaveachse unterschiedlich eingestellt sein, ohne dass die Überwachungen ansprechen.
- Um im Fehlerfall das gleiche Bremsverhalten für alle Achsen eines Master-Slave-Verbundes zu erreichen, werden bei aktiver Kopplung alle Achsen des Master-Slave-Verbundes mit der gleichen Alarmreaktion beaufschlagt.
- Die Parametrierung und Auswertung der Stillstandserkennung (z. B. Bremsrampe) erfolgt axial für jede Achse im Master-Slave-Verbund. Dies kann zu einer asynchronen Wegnahme der Reglerfreigabe bei den gekoppelten Achsen führen.
- Beim Aufheben von Fehlerzuständen erfolgt für die Slaveachsen kein Repositionieren auf die Unterbrechungsstelle.

Virtuelle Achse

Master- und/oder Slaveachsen dürfen keine "virtuelle" Achsen sein:

MD30132 \$MA_IS_VIRTUAL_AX (Achse ist virtuelle Achse) = 0

Modulo Rundachsen

- Für Slaveachsen wird bei eingeschalteter Kopplung der Istwert in der Serviceanzeige nicht modulo 360° angezeigt. Unabhängig von der Einstellung im Maschinendatum:
MD30310 \$MA_ROT_IS_MODULO[<Slaveachse>]

Spindeln

- Wird eine Master-Slave-Kopplung mit Spindeln aktiviert, wird die Slavespindel im Drehzahlsteuerbetrieb betrieben. In der Serviceanzeige wird der Istwert der Slavespindel nicht modulo 360° angezeigt. Im Automatikgrundbild wird der Istwert modulo 360° angezeigt.
- Beschleunigen von Spindeln an der Stromgrenze bietet im gekoppelten Zustand möglicherweise keine Stellreserve für den Momentenausgleichsregler mehr, um die gewünschte Momentenaufteilung zwischen Master und Slave einzuhalten.
- Die maximale Futterdrehzahl der Masterspindel im folgenden Maschinendatum ist kleiner bzw. gleich der von den Slavespindeln zu projektieren:
MD35100 \$MA_SPIND_VELO_LIMIT[<Masterspindel>]
- Die axiale Geschwindigkeitsüberwachung sollte der Futterdrehzahl angepasst werden:
MD36200 \$MA_AX_VELO_LIMIT[<Masterspindel>]

19.8.2 Axiale NC/PLC-Nahtstellensignale

- In der Bremsenansteuerungslogik darf bei Slaveachsen das NC/PLC-Nahtstellensignal: DB3x.DBX61.5 (Lageregler aktiv) nicht mehr ausgewertet werden. Das Signal wird bei eingeschalteter Kopplung nicht mehr gesetzt. Stattdessen ist folgendes NC/PLC-Nahtstellensignal zu verwenden: DB31, ... DBX96.7 (Master-Slave-Kopplung aktiv).
- Bei angeforderter und bei eingeschalteter Kopplung werden für die Slaveachse folgende NC/PLC-Nahtstellensignale direkt von der Masterachse übernommen:
DB31, ... DBX2.1 (Reglerfreigabe)
DB31, ... DBX21.7 (Impulsfreigabe)
- Die Wegnahme der Reglerfreigabe für die Masterachse:
DB31, ... DBX2.1 (Reglerfreigabe) = 0
führt innerhalb der projektierten Zeit von:
MD36610 \$MA_AX_EMERGENCY_STOP_TIME
auch zum interpolatorischen Abbremsen der Slaveachse. Die entsprechenden Drehzahl- und Stromreglerfreigaben werden erst nach Ablauf der projektierten Zeit weggenommen:
MD36620 \$MA_SERVO_DISABLE_DELAY_TIME
- Für ein einheitliches Bremsverhalten in allen gekoppelten Achsen wird empfohlen, folgende Parameter gleich einzustellen:
 - MD36620 \$MA_SERVO_DISABLE_DELAY_TIME
 - p9560 (Impulslöschung Abschaltgeschwindigkeit)
 - p1228 (Impulslöschung Verzögerungszeit)

- Ist für die Master- oder Slaveachse eines der folgenden Antriebsstatussignale nicht gesetzt:
DB31, ... DBX61.7 (Stromregler aktiv) == 0 ODER
DB31, ... DBX61.6 (Drehzahlregler aktiv) == 0
wird im Stillstand der Slaveachse das Statussignal zurückgesetzt:
DB31, ... DBX96.7 (Master/Slave aktiv) = 0
Sobald Master- und Slaveachse wieder in Regelung sind:
DB31, ... DBX61.7 (Stromregler aktiv) == 1 UND
DB31, ... DBX61.6 (Drehzahlregler aktiv) == 1
wird für die Slaveachse das Statussignal wieder gesetzt:
DB31, ... DBX96.7 (Master/Slave aktiv) = 1
- Der Momentenausgleichsregler kann über die NC/PLC-Nahtstelle eingeschaltet werden über:
DB31, ... DBX24.4 = 1 (Momentenausgleichsregler Ein)
Der Status des Momentenausgleichsreglers wird angezeigt in:
DB31, ... DBX96.4 (Master/Slave Ausgleichregler aktiv)

Hinweis

Bei eingeschalteter Kopplung wird die Slaveachse nur noch drehzahl geregelt. Das NC/PLC-Nahtstellensignal: DB31, ... DBX61.5 (Lageregler aktiv) wird nicht mehr gesetzt.

19.8.3 Zusammenspiel mit anderen Funktionen

Funktionsgenerator

Für die Vermessung des Drehzahlregelkreises bei eingeschalteter Kopplung wird empfohlen, die Momentengewichtung der Slaveachse (MD37268 \$MA_MS_TORQUE_WEIGHT_SLAVE) auf einen kleinen Wert einzustellen. Eine mechanisch mitgeschleppte Slaveachse wird dabei nicht vom Momentenausgleichsregler am Verfahren gehindert.

Referenzpunktfahren

Bei eingeschalteter Kopplung ist nur die Masterachse referenzierbar. Das Referenzieren von Slaveachsen wird unterdrückt. Die Slaveachse muss dafür nicht explizit aus der Referenzpflicht genommen werden. Der Referenzierstatus von gekoppelten Slaveachsen bleibt unverändert. Die Slaveachsposition ist im Regelfall ungleich der Masterachsposition. Diese Positionsdifferenz ist nicht von Bedeutung. Bei getrennter Kopplung kann jede der Achsen wie gewohnt einzeln referenziert werden.

Positionsbezogene Kompensationen

Positionsbezogene Kompensationen der Slaveachse wie Spindelsteigungsfehler-, Lose-, Temperatur- und Durchhangkompensation werden aufgrund des nicht aktiven Lagereglers der Slaveachse nicht wirksam.

Eine korrekte Verrechnung der Losekompensation setzt voraus, dass die Lose der Slaveachse im gekoppelten Zustand immer durch die Bewegung der Masterachse überfahren wird. Ein Aufheben der Kopplung innerhalb der Lose führt zu einem falschen Istwert der Slaveachse.

Dynamische Steifigkeits-Regelung (DSC)

Die Funktion "Dynamische Steifigkeits-Regelung (DSC)" muss für alle Achsen eines Master-Slave-Verbands gleichermaßen aktiv bzw. nicht aktiv sein.

MD32640 \$MA_STIFFNESS_CONTROL_ENABLE

Drehzahl-/Momentenvorsteuerung (FFW)

Die Funktion "Drehzahl-/Momentenvorsteuerung (FFW)" muss in der Slaveachse nicht explizit aktiviert werden. Es werden die entsprechenden Einstellungen der Masterachse verwendet. Der Drehzahlvorsteuerwert der Masterachse ist bereits im Drehzahl Sollwert der Slaveachse eingerechnet. Bei aktiver Momentenvorsteuerung wird zusätzlich der lastseitige Momentenvorsteuerwert der Masterachse im Slaveantrieb wirksam.

Bei eingeschalteter Kopplung verändern sich die mechanischen Verhältnisse. Achsspezifische Einstellungen müssen entsprechend angepasst werden. Alle gekoppelten Antriebe sollten die gleiche Drehzahlreglerdynamik aufweisen.

Gantry

Wird zur Leistungsverstärkung auf jeder Seite des Gantryverbandes jeweils eine Master-Slave-Beziehung definiert, so dürfen nur die Leit- bzw. die Folgeachse als eine Masterachse betrieben werden.

Fahren auf Festanschlag (FXS)

Die Funktion "Fahren auf Festanschlag (FXS)" ist bei eingeschalteter Kopplung nur in der Masterachse programmierbar und wirkt sich auf die Master- und Slaveachse unterschiedlich aus:

- Der programmierte Wert wird in Prozent des Antriebsnennmoments der Masterachse ausgedrückt. Das Erreichen des Festanschlags wird von der Masterachse erkannt.
- Der programmierte Wert wird ebenfalls in der Slaveachse wirksam, bezieht sich aber auf das eigene Antriebsnennmoment der Slaveachse.

Bei unterschiedlichen Nennmomenten der Master- und der Slaveachse kann eine Angleichung über das folgende Maschinendatum der Slaveachse vorgenommen werden:

MD37014 \$MA_FIXED_STOP_TORQUE_FACTOR

Die Vorgabe vom Faktor < 1 reduziert entsprechend das programmierte Klemmmoment in der Slaveachse.

Folgende Randbedingungen sind zu beachten:

- Eine Momentenaufteilung zwischen der Master- und der Slaveachse ist während der Klemmung nicht möglich, weil der Momentenausgleichsregler hierbei abgeschaltet wird.
- Zustandsänderungen der Master-Slave-Kopplung sind während Fahren auf Festanschlag unwirksam. Erst nach Abschluss der Funktion wird die Vorgabe eines neuen Zustands übernommen.

Safety Integrated (neu 840D sl)

Da die Slaveachse über den Drehzahlollwert der Masterachse verfahren wird, ist die achsspezifische Sollwertbegrenzung MD36933 \$MA_SAFE_DES_VELO_LIMIT in den gekoppelten Slaveachsen unwirksam. Sämtliche Safety-Überwachungen bleiben dagegen in den Slaveachsen uneingeschränkt wirksam.

Getriebestufenwechsel bei eingeschalteter Master-Slave-Kopplung

Ein automatischer Getriebestufenwechsel in einer gekoppelten Slavespindel ist nicht möglich und kann nur indirekt mit Hilfe der Masterspindel realisiert werden. Der Zeitpunkt für den Getriebestufenwechsel leitet sich dann von der Masterspindel ab. Die Pendelbewegung der gekoppelten Slavespindel wird implizit durch die Pendelbewegung der Masterspindel erzeugt.

Im Gegensatz zur Masterspindel muss in der gekoppelten Slavespindel explizit der dazugehörige Parametersatz angewählt werden. Um die Parametersatzvorgabe zu ermöglichen, muss das folgende Maschinendatum eingestellt werden:

MD35590 \$MA_PARAMSET_CHANGE_ENABLE = 2

Hinweis

Weitere Informationen zum Getriebestufenwechsel und zum Parametersatz beim Wechsel in den Spindelbetrieb siehe:

Literatur:

Funktionshandbuch Grundfunktionen; Spindeln (S1)

Achscontainer

Ist eine gekoppelte Slaveachse als Container-Achse projektiert, wird bei einer Achscontainer-Drehung der Alarm "4025 Achscontainer %3 schalten nicht erlaubt: Master-Slave aktiv Kanal %1 Achse %2 " angezeigt.

Achscontainer können bei wechselndem Master verwendet werden, um jeweils die nach einer Achscontainer-Drehung relevante Spindel zur Masterspindel zu machen. Sowohl Master- als auch Slavespindel können Container-Spindel sein.

Damit nach der Achscontainer-Drehung die Kopplung mit jeweils einer anderen Spindel geschlossen werden kann, muss vor der Drehung die alte Kopplung getrennt, die Projektierung gelöscht und nach der Drehung die neue Kopplung projektiert und geschlossen werden.

Beispiel für eine zyklische Kopplungssequenz (Lage = 3 / Container = CT1)

Programmcode	Kommentar
MASLDEF(AUX,SPI(3))	; S3 Master für AUX
MASLON(AUX)	; Kopplung ein für AUX
M3=3 S3=4000	; Bearbeitung ...
MASLDEL(AUX)	; Projektierung löschen und trennen der Kopplung
AXCTSWE(CT1)	; Achscontainer-Drehung

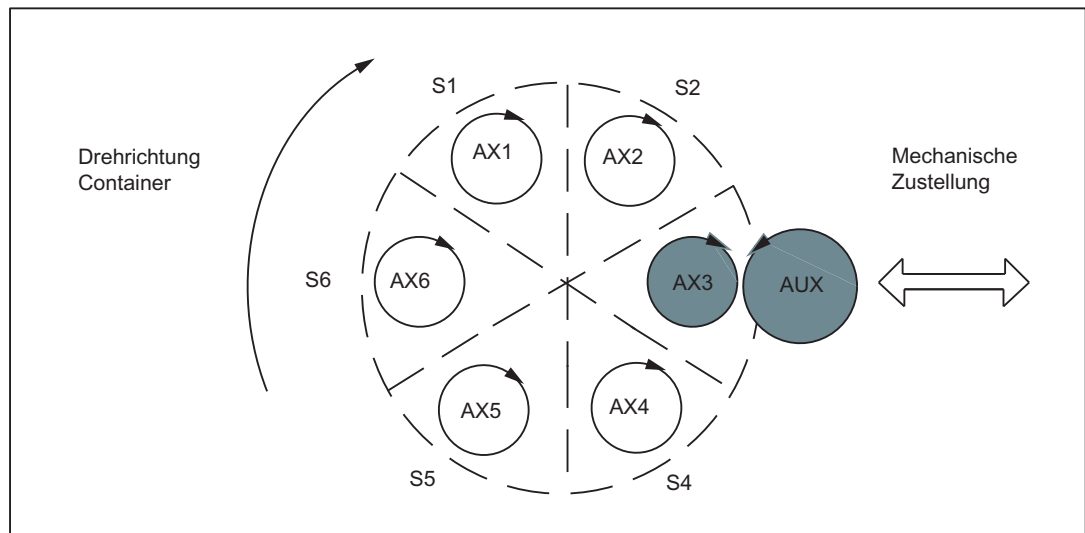


Bild 19-7 Kopplung zwischen Containerspindel S3 und Hilfsmotor AUX (vor der Drehung)

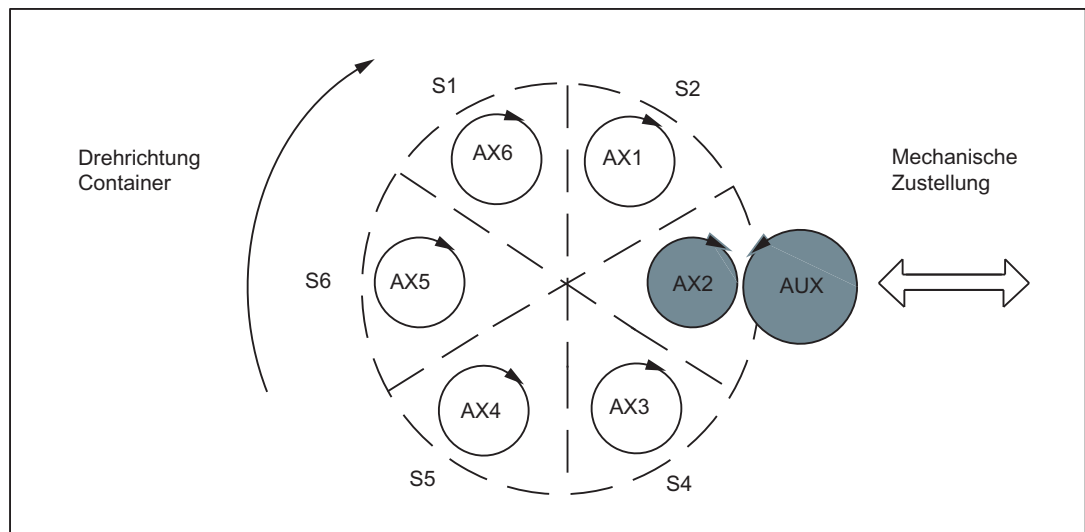


Bild 19-8 Kopplung zwischen Containerspindel S3 und Hilfsmotor AUX (nach der Drehung)

Hardware- und Software-Endschalter

Wird von einer Slaveachse der Soft- oder Hardware-Endschalter überfahren, wird der Master-Slave-Verband über die Masterachse angehalten. Im Alarm wird die betroffene Slaveachse angezeigt. Der Endschalter wird dabei von der Slaveachse um den Betrag des Bremsweges der Masterachse überfahren.

Das Freifahren der Slaveachse vom Endschalter muss über das Verfahren der Masterachse erfolgen. Ein Ausschalten der Kopplung (MASLOF) und Verfahren der Slaveachse ist nicht

möglich. Die Kopplung kann erst wieder ausgeschaltet werden, wenn die Fehlerursache beseitigt ist.

ACHTUNG

Power Off/On oder Warmstart (Reset (po)) nach Überfahren des Soft- oder Hardware-Endschalters und Ausschalten der Kopplung mit MASLOF

Wird von einer Slaveachse der Soft- oder Hardware-Endschalter überfahren und während die Slaveachse noch hinter dem Endschalter steht, versucht die Kopplung mit MASLOF auszuschalten, wird die Meldung "Warten auf gekoppelte Slaveachse" angezeigt. Die Kopplung kann dann nur noch mittels Power Off/On oder Warmstart (Reset (po)) ausgeschaltet werden.

Satzsuchlauf

Statische Kopplung

Die Funktion "Satzsuchlauf mit Berechnung" (SERUPRO) kann in Verbindung mit einer statischen Master-Slave-Kopplung uneingeschränkt genutzt werden.

Dynamische Kopplung (nur 840D sl)

Bei dynamischer Kopplung sind bezüglich der Programmbefehle MASLON und MASLOF folgende Einschränkungen zu beachten:

- Die gekoppelten Achsen müssen auf derselben NCU liegen.
- Die gekoppelten Achsen müssen sich zum Zeitpunkt des Satzsuchlaufs im gleichen Kanal befinden.
- Nach dem Satzsuchlauf muss über das System-ASUP "PROGEVENT.SPF" der Kopplungszustand, die zugehörigen Achspositionen und Drehzahlen nachträglich vom Anwender beeinflusst werden. Dafür stehen Systemvariablen zur Verfügung.
 - \$P_PROG_EVENT: Die Variable gibt Auskunft über das Ereignis, welches das Unterprogramm aktiviert hat. Der Wert 5 steht für den Satzsuchlauf.
 - \$P_SEARCH_MASLC[Slaveachsname]: Die Variable steht für die Änderung des Kopplungszustands während des Satzsuchlaufs.
 - \$P_SEARCH_MASLD[Slaveachsname]: Die Variable gibt den im Satzsuchlauf ermittelten Positionsversatz zwischen der Slave- und der Masterachse zum Schließzeitpunkt der Kopplung wieder.
 - \$AA_MASL_STAT[Slaveachsname]: Die Variable gibt den aktuellen Kopplungszustand wieder.

- Das System-ASUP "PROGEVENT.SPF" muss unter folgendem Pfad abgelegt werden: /
_N_CMA_DIR/_N_PROG_EVENT_SPF
- Damit PROGEVENT.SPF gestartet wird, sind folgende Maschinendaten zu parametrieren.
NC-spezifische Maschinendaten:
 - MD11450 \$MN_SEARCH_RUN_MODE = 'H02'
 - MD11602 \$MN_ASUP_START_MASK = 'H01'
 - MD11604 \$MN_ASUP_START_PRIO_LEVEL = 100
 Kanal-spezifische Maschinendaten:
 - MD20105 \$MC_PROG_EVENT_IGN_REFP_LOCK = 'H3F'

Tabelle 19-1 Beispiel 1: PROGEVENT.SPF: Beispiel 1

Programmcode	Kommentar
N10 IF \$P_PROG_EVENT==5	; Satzsuchlauf aktiv
N20 IF ((\$P_SEARCH_MASLC[Y]<>0) AND (\$AA_MASL_STAT[Y]<>0))	; Im Satzsuchlauf hat sich der Kopp- lungszustand verändert UND aktueller Zustand ist "gekoppelt"
N30 MASLOF(Y)	; Kopplung trennen
N40 SUPA Y=\$AA_IM[X]-\$P_SEARCH_MASLD[Y]	; Positionsversatz über die Slaveachse herausfahren
N50 MASLON(Y)	; Kopplung schließen
N60 ENDIF	
N70 ENDIF	
N80 REPOSA	

Tabelle 19-2 Beispiel 2: PROGEVENT.SPF

Programmcode	Kommentar
N10 IF \$P_PROG_EVENT==5	; Satzsuchlauf aktiv
N20 IF ((\$P_SEARCH_MASLC[SPI(2)]<>0) AND (\$AA_MASL_STAT[SPI(2)]=0))	; Im Satzsuchlauf hat sich der Kopplungs- zustand der zweiten Spindel verändert UND aktueller Zustand ist "getrennt"
N30 M2=\$P_SEARCH_SDIR[2]	; Drehrichtung aktualis.
N40 S2=\$P_SEARCH_S[2]	; Drehzahl aktualisieren
N50 ENDIF	
N60 ENDIF	
N70 REPOSA	

Weitere Anwendungsbeispiele (siehe Kapitel "Beispiele (Seite 770)").

Hinweis

Es wird empfohlen bei eingeschalteter Kopplung für einen Satzsuchlauf ausschließlich den Suchlauftyp 5, "Satzsuchlauf über Programmtest" (SERUPRO), zu verwenden.

Ausführliche Informationen zu ereignisgesteuerten Programmaufrufen und "Satzsuchlauf über Programmtest" (SERUPRO) sind enthalten in:

Literatur:

Funktionshandbuch Grundfunktionen;

- Hilfsfunktionsausgaben an PLC (H2)
 - BAG, Kanal, Programmbetrieb, Reset-Verhalten (K1)
 - SpindelIn (S1)
-

19.9 Beispiele

19.9.1 Master-Slave-Kopplung zwischen AX1=Master und AX2=Slave

Konfiguration

Master-Slave-Kopplung zwischen AX1=Master und AX2=Slave.

1. Maschinenachsnnummer der Masterachse bei Drehzahlsollwertkopplung
MD37250 \$MA_MS_ASSIGN_MASTER_SPEED_CMD[AX2] = 1
2. Masterachse bei Momentenaufteilung gleich der Masterachse bei Drehzahlsollwertkopplung
MD37252 \$MA_MS_ASSIGN_MASTER_TORQUE_CTRL[AX2] = 0
3. Permanente Kopplung
MD37262 \$MA_MS_COUPLING_ALWAYS_ACTIVE[AX2] = 1
4. Momentenaufschaltung erfolgt sowohl in der Master- wie in der Slaveachse
MD37254 \$MA_MS_TORQUE_CTRL_MODE[AX2] = 0
5. Momentenaufteilung zwischen der Master- und der Slaveachse jeweils 50% zu 50%
MD37268 \$MA_MS_TORQUE_WEIGHT_SLAVE[AX2] = 50
6. Parameter des Momentenausgleichsreglers
MD37256 \$MA_MS_TORQUE_CTRL_P_GAIN[AX2] = 0.5
MD37258 \$MA_MS_TORQUE_CTRL_I_TIME[AX2] = 5.0

19.9.2 Kopplung schließen über PLC

Diese Anwendung ermöglicht, eine Master-Slave-Kopplung zwischen den Maschinenachsen AX1=Masterachse und Ax2=Slaveachse im Betrieb schließen und trennen zu können.

Vorbedingungen

- Eine projektierte Masterachse
MD37250 \$MA_MS_ASSIGN_MASTER_SPEED_CMD ≠ 0
- Aktivierung von Master-Slave-Kopplung über
MD37262 \$MA_MS_COUPLING_ALWAYS_ACTIVE=0
- Die Kopplung ist offen.

Typischer Ablauf

Aktion	Auswirkung/Bemerkung
• Kopplungsposition anfahren	Jede Achse fährt auf die Kopplungsposition.
• Kopplung mechanisch schließen	Beide Achsen werden mechanisch miteinander verbunden.
• Anforderung zum Schließen der Kopplung	Das PLC-Nahtstellensignal "Master/Slave Ein" DB32, ... DBX24.7 wird gesetzt.
• Kopplungszustand rücklesen	Im Achsstillstand wird von der eingekoppelten Slaveachse das PLC-Nahtstellensignal "Master/Slave aktiv" DB32, ... DBX96.7 gesetzt und das "Lageregler aktiv" DB32, ... DBX61.5 gelöscht. Rückmeldung abwarten
• Master-Slave-Verbund verfahren	Die Masterachse wird verfahren.

19.9.3 Kopplung schließen/trennen über Teileprogramm bei SINUMERIK 840D sl

Diese Anwendung ermöglicht, eine Master-Slave-Kopplung zwischen den Maschinenachsen AX1=Masterachse und AX2=Slaveachse über Teileprogramm zu schließen und zu trennen.

Vorbedingungen

- Eine projektierte Masterachse MD37250 0 0.
- Keine statische Master-Slave-Kopplung:
MD37262 \$MA_MS_COUPLING_ALWAYS_ACTIVE= 0
- Die Kopplung ist offen.

Teileprogramm

Programmcode	Kommentar
N10 G0 AX1=0 AX2=0	; Achsspezifisch die Kopplungsposition anfahren.
N20 MASLON (AX2)	; (Achsen mechanisch verbinden)
	; Kopplung schließen.
N30 AX1=100	; Master-Slave-Verbund über Masterachse verfahren.
N40 MASLOF (AX2)	; Kopplung öffnen.
	; (Achsen mechanisch trennen)

Programmcode	Kommentar
N50 AX1=200 AX2=200	; Die Achsen getrennt verfahren.
N60 M30	

19.9.4 Mechanische Bremse öffnen

Diese Anwendung ermöglicht, eine Bremsensteuerung für die Master-Slave-gekoppelten Maschinenachsen AX1=Masterachse und AX2=Slaveachse zu realisieren.

Vorbedingungen

- Master-Slave-Kopplung projiziert
- Achsen stehen still.
- Reglerfreigaben fehlen.

Typischer Ablauf

Aktion	Auswirkung/Bemerkung
<ul style="list-style-type: none"> • Anforderung zum Schließen der Kopplung 	Gesetzt wird das PLC-Nahtstellensignal: DB31, ... DBX24.7 (Master/Slave Ein)
<ul style="list-style-type: none"> • Reglerfreigabe setzen 	Das PLC-Nahtstellensignal "Reglerfreigabe" DB31, ... DBX2.1 wird für beide Achsen gesetzt.
<ul style="list-style-type: none"> • Rückmeldungen auswerten 	PLC-Nahtstellensignale der Masterachse mit UND verknüpfen: DB31, ... DBX61.7 (Stromregler aktiv) DB31, ... DBX61.6 (Drehzahlregler aktiv) DB31, ... DBX61.5 (Lageregler aktiv) PLC-Nahtstellensignale der Slaveachse mit UND verknüpfen: DB31, ... DBX61.7 (Stromregler aktiv) DB31, ... DBX61.6 (Drehzahlregler aktiv) DB31, ... DBX96.7 (Master/Slave aktiv)
<ul style="list-style-type: none"> • Bremsen öffnen 	Wenn das Ergebnis der UND-Verknüpfungen in der Master- und der Slaveachse $\neq 0$ ist, darf die Bremse gelöst werden.

19.10 Datenlisten

19.10.1 Maschinendaten

19.10.1.1 Achs-/Spindel-spezifische Maschinendaten

Nummer	Bezeichner: \$MA_	Beschreibung
37250	MS_ASSIGN_MASTER_SPEED_CMD	Masterachse bei Drehzahlsollwertkopplung
37252	MS_ASSIGN_MASTER_TORQUE_CTR	Masterachse für Momentenaufteilung
37254	MS_TORQUE_CTRL_MODE	Verschaltung Momentenausgleichsregler
37255	MS_TORQUE_CTRL_ACTIVATION	Aktivierung Momentenausgleichsregler
37256	MS_TORQUE_CTRL_P_GAIN	Verstärkungsfaktor des Momentenausgleichsreglers
37258	MS_TORQUE_CTRL_I_TIME	Nachstellzeit des Momentenausgleichsreglers
37260	MS_MAX_CTRL_VELO	Begrenzung Momentenausgleichsregler
37262	MS_COUPLING_ALWAYS_ACTIVE	Dauerhafte Master-Slave-Kopplung
37263	MS_SPIND_COUPLING_MODE	Koppelverhalten einer Spindel
37264	MS_TENSION_TORQUE	Master-Slave-Verspannmoment
37268	MS_TORQUE_WEIGHT_SLAVE	Momentengewichtung der Slave-Achse
37270	MS_VELO_TOL_COARSE	Master-Slave-Geschwindigkeitstoleranz "grob"
37272	MS_VELO_TOL_FINE	Master-Slave-Geschwindigkeitstoleranz "fein"
37274	MS_MOTION_DIR_REVERSE	Master-Slave-Verfahrriichtung invertieren

19.10.2 Systemvariablen

Nach dem Satzsuchlauf können über das System-ASUP (asynchrones Unterprogramm) "PROGEVENT.SPF" der Kopplungszustand und die zugehörigen Achspositionen nachträglich beeinflusst werden. Dafür stehen die Systemvariablen \$P_SEARCH_MASL_C, \$P_SEARCH_MASL_D und \$AA_MASL_STAT zur Verfügung, mit denen der Positionsversatz zwischen den gekoppelten Achsen und der Kopplungszustand verändert werden kann:

Bezeichner	Bedeutung
\$P_SEARCH_MASLC[Slaveachsname]	Die Variable registriert eine Änderung des Kopplungszustands während des Satzsuchlaufes SERUPRO.
\$P_SEARCH_MASLD[Slaveachsname]	Die Variable gibt den Positionsversatz zwischen der Slave- und der Master-Achse zum Schließzeitpunkt der Kopplung an.
\$AA_MASL_STAT[Slaveachsname]	Die Variable gibt den aktuellen Kopplungszustand aus. Wert ≠ 0: "Master-Slave-Kopplung aktiv" In diesem Fall enthält er die aktuelle Maschinenachsnummer der Masterachse, bei aktivem NCU-Link (mehrere Bedientafelfronten und NCUs) zusätzlich die NCU-Nr (an der Hunderter-Stelle. Beispiel: 201 für Achse1 auf NCU2.

19.10.3 Signale**19.10.3.1 Signale an Achse/Spindel**

Signalname	SINUMERIK 840D sl	SINUMERIK 828D
Anforderung Master-Slave Momentenausgleichsregler Ein==1 / Aus==0	DB31,DBX24.4	DB380x.DBX5000.4
Anforderung Master-Slave Kopplung Ein==1 / Aus==0	DB31,DBX24.7	DB380x.DBX5000.7

19.10.3.2 Signale von Achse/Spindel

Signalname	SINUMERIK 840D sl	SINUMERIK 828D
Zustand Master-Slave Differenzdrehzahl "Fein"	DB31, ... DBX96.2	DB390x.DBX5000.2
Zustand Master-Slave Differenzdrehzahl "Grob"	DB31, ... DBX96.3	DB390x.DBX5000.3
Zustand Master-Slave Momentenausgleichsregler Ein==1 / Aus==0	DB31, ... DBX96.4	DB390x.DBX5000.4
Zustand Master-Slave Kopplung Ein==1 / Aus==0	DB31, ... DBX96.7	DB390x.DBX5000.7

20.1 Kurzbeschreibung

Das Transformationspaket Handling ist für den Einsatz bei **Handhabungsmaschinen** und **Robotern** konzipiert. Es handelt sich dabei um eine Art Baukastensystem, bei dem der Kunde die Möglichkeit hat, die Transformation für seine Maschine über Maschinendaten zu konfigurieren, sofern die Kinematik im Transformationspaket Handling enthalten ist.

Kapitelaufbau

Das Kapitel "Transformationspaket Handling" behandelt folgende Themen:

- Kapitel "Kinematische Transformation (Seite 776)" beschreibt das Umfeld der kinematischen Transformation.
- Kapitel "Begriffsbestimmungen (Seite 776)" erläutert grundlegende Begriffe.
- Kapitel "Konfiguration der kinematischen Transformation (Seite 779)" erläutert die zur Projektierung notwendigen Maschinendaten.
- Das Kapitel "Kinematikbeschreibungen (Seite 794)" stellt die gängigsten 2- bis 5-Achs Kinematiken, die mit dem Transformationspaket Handling konfiguriert werden können, anhand von Projektierungsbeispielen dar.
- Kapitel "Werkzeugorientierung (Seite 816)" bis Kapitel "Werkzeugprogrammierung (Seite 823)" behandelt die Programmierung. Hierbei wird auf die Programmierung der Orientierung, der Angabe der Werkzeugparameter und den Aufruf der Transformation eingegangen.

Abkürzungen

FL	Flansch-Koordinatensystem
HP	Handpunkt-Koordinatensystem
IRO	internes Roboter-Koordinatensystem
p_3, q_3, r_3	Koordinaten der letzten Grundachse
RO	Roboter- bzw. Fußpunkt-Koordinatensystem (= Basis-Koordinatensystem)
WS	Werkstück-Koordinatensystem
WZ	Werkzeug-Koordinatensystem
x_3, y_3, z_3	Koordinaten der ersten Handachse

20.2 Kinematische Transformation

Aufgabe der Transformation

Aufgabe der Transformation ist es, Bewegungen der Werkzeugspitze, die in einem kartesischen Koordinatensystem programmiert sind, in die Maschinenachsenpositionen zu transformieren.

Einsatzgebiet

Das hier beschriebene Transformationspaket Handling ist darauf ausgelegt, eine möglichst große Zahl von Kinematiken allein durch Maschinendatenparametrierung abzudecken. Es können zurzeit Kinematiken mit 2 bis maximal 6 Achsen, die in die Transformation eingehen, betrieben werden. Dies entspricht bis zu fünf räumlichen Freiheitsgraden. Hierbei sind maximal 3 Freiheitsgrade für die Translation und 3 Freiheitsgrade für die Orientierung möglich. Damit kann bei einer 5-Achsen Maschine ein Werkzeug (Fräser, Laserstrahl) in jedem Punkt des Bearbeitungsraumes beliebig zum Werkstück orientiert werden. Das Werkstück wird immer im rechtwinkligen Werkstückkoordinatensystem programmiert, eventuell programmierte oder eingestellte Frames drehen und verschieben dieses System auf das Basissystem. Die kinematische Transformation setzt diese Informationen dann in Bewegungsvorgaben für die realen Maschinenachsen um. Die kinematische Transformation benötigt Angaben über den Aufbau (Kinematik) der Maschine, die in Maschinendaten hinterlegt sind.

Kinematikklassen

Das Transformationspaket Handling gliedert sich in zwei Kinematikklassen, die ausgewählt werden über das Maschinendatum:

MD62600 \$MC_TRAFO6_KINCLASS (Kinematikkategorie)

- STANDARD: Diese Klasse beinhaltet die gängigsten Kinematiken.
- SPECIAL: Sonderkinematiken

20.3 Begriffsbestimmungen

20.3.1 Einheiten und Richtungen

Längen und Winkel

In den Transformationsmaschinendaten werden sämtliche Längen in Millimeter bzw. Inch und sämtliche Winkel, soweit nicht anders vermerkt, in Grad im Intervall [-180°, 180°] angegeben.

Drehrichtung

Bei Winkeln geben Pfeile in den Zeichnungen immer die mathematisch positive Drehrichtung an.

20.3.2 Positions- und Orientierungsbeschreibung mit Hilfe von Frames

Um eine Abgrenzung zum Begriff Frame, wie er in der NC-Sprache definiert ist, vorzunehmen, wird im Folgenden erläutert, welche Bedeutung der Begriff Frame im Bezug auf das Transformationspaket Handling hat.

Frame

Über einen Frame lässt sich ein Koordinatensystem in ein anderes überführen. Hierbei unterscheidet man zwischen Translation und Rotation. Eine Translation bewirkt eine Verschiebung, eine Rotation eine Verdrehung des Koordinatensystems bezüglich des Ausgangssystems. Als Bezeichner für eine Translation werden die Koordinaten X, Y und Z verwendet. Sie sind so festgelegt, dass das Koordinatensystem ein Rechts-System ergibt.

Translation

Eine Translation ist immer bezüglich der Koordinatenrichtungen des Ausgangssystems angegeben. Die Richtungen sind den Maschinendaten folgendermaßen zugeordnet:

- X- Richtung: ..._POS[0]
- Y- Richtung: ..._POS[1]
- Z- Richtung: ..._POS[2]

Rotation

Eine Rotation wird in RPY-Winkel A, B und C (R=Roll, P=Pitch, Y=Yaw) angegeben. Die positive Drehrichtung wird durch die Rechte-Hand-Regel festgelegt, d. h. zeigt der Daumen der rechten Hand in Richtung der Drehachse, geben die restlichen Finger die positive Winkelrichtung an. Die Winkel A und C sind im Intervall [-180; +180] und B im Intervall [-90; +90] definiert.

Die RPY-Winkel sind folgendermaßen festgelegt:

- Winkel A: 1. Drehung um die Z-Achse des Ausgangssystems
- Winkel B: 2. Drehung um die gedrehte Y-Achse
- Winkel C: 3. Drehung um die 2-fach gedrehte X-Achse

Die RPY-Winkel sind den Maschinendaten folgendermaßen zugeordnet:

- Winkel A: ..._RPY[0]
- Winkel B: ..._RPY[1]
- Winkel C: ..._RPY[2]

Das nachfolgenden Bild zeigt ein Beispiel für eine Drehung um RPY-Winkel. Das Ausgangskordinatensystem X1, Y1, Z1 wird dabei in das Koordinatensystem X4, Y4, Z4 überführt

1. Drehung um den Winkel A um die Z1-Achse
2. Drehung um den Winkel B um die Y2-Achse
3. Drehung um den Winkel C um die X3-Achse.

20.3 Begriffsbestimmungen

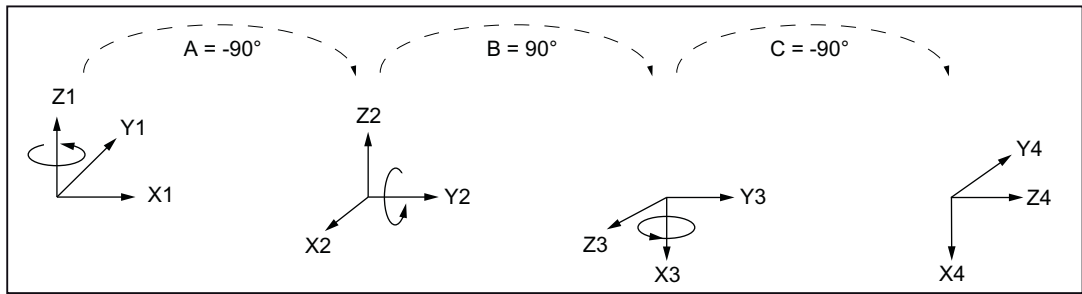


Bild 20-1 Beispiel für Drehung um die RPY-Winkel

20.3.3 Gelenkdefinition

Bedeutung

Ein Schiebegelenk wird durch eine translatorische, ein Drehgelenk durch einer rotatorische Achse realisiert.

Die Grundachskennungen bestimmen sich aus der Anordnung und Reihenfolge der einzelnen Gelenke. Diese werden mit Buchstabenkennungen (S, C, R, N) angeben.

Kennung	Bezeichnung	Symbole	
S	Schiebegelenk		
C	Schiebegelenk Drehgelenk		
R	Drehgelenk Drehgelenk Drehgelenk ⊥ Schiebegelenk	 	
N	Drehgelenk ⊥ Drehgelenk		
FL	Flansch zur Befestigung des Werkzeugs		<ul style="list-style-type: none"> → positive Achsrichtung positive Achsrichtung: <ul style="list-style-type: none"> ⊗ - senkrecht in die Zeichenebene hinein ⊙ - senkrecht aus der Zeichenebene heraus ↻ positive Drehrichtung Achsrichtungen parallel zueinander ⊥ Achsrichtungen senkrecht zueinander
WZ	Werkzeug (Tool)		

Bild 20-2 Gelenkbezeichnungen

20.4 Konfiguration der kinematischen Transformation

Bedeutung

Damit die kinematische Transformation die programmierten Werte in Achsbewegungen umrechnen kann, sind einige Informationen über die mechanische Ausführung der Maschine notwendig, die in Maschinendaten abgelegt werden:

- Achszuordnungen
- Geometrie-Informationen

20.4.1 Allgemeine Maschinendaten

MD24100 \$MC_TRAFO_TYPE_1 (Definition der Transformation 1 im Kanal)

Hier ist der Wert 4100 für das Transformationspaket Handling einzutragen.

MD24110 \$MC_TRAFO_AXES_IN_1 (Achszuordnung für Transformation)

Die Achszuordnung am Eingang der Transformation legt fest, welche Achse von der Transformation intern auf eine Kanalachse abgebildet wird. Sie wird im Maschinendatum festgelegt:

MD24110 \$MC_TRAFO_AXES_IN_1

Für das Transformationspaket Handling ist die Reihenfolge der Achsen fest vorgegeben. In die ersten 3 Komponenten (Index 0...2) werden die Grundachsen eingetragen. Die oberen 3 Komponenten (Index 3...5) werden den Handachsen zugeordnet.

- MD24110 \$MC_TRAFO_AXES_IN_1[0] = 1 ; 1. Grundachse
- MD24110 \$MC_TRAFO_AXES_IN_1[1] = 2 ; 2. Grundachse
- MD24110 \$MC_TRAFO_AXES_IN_1[2] = 3 ; 3. Grundachse
- MD24110 \$MC_TRAFO_AXES_IN_1[3] = 4 ; 1. Handachse
- MD24110 \$MC_TRAFO_AXES_IN_1[4] = 5 ; 2. Handachse
- MD24110 \$MC_TRAFO_AXES_IN_1[5] = 6 ; 3. Handachse

MD24120 \$MC_TRAFO_GEOAX_ASSIGN_TAB_1 (Zuordnung Geometrieachse zu Kanalachse für Transformation 1)

Wie viele translatorische Freiheitsgrade für die Transformation vorhanden sind wird eingetragen über das Maschinendatum:

MD24120 \$MC_TRAFO_GEOAX_ASSIGN_TAB_1

Normalerweise entsprechen die 3 Geometrieachsen den kartesischen Achsrichtungen X, Y und Z.

Hierbei sind die ersten drei Kanalachsnummern aus MD24110 \$MC_TRAFO_AXES_IN_1 in MD24120 \$MC_TRAFO_GEO_AX_ASSIGN_TAB_1 zu übernehmen.

- MD24120 \$MC_TRAFO_GEO_AX_ASSIGN_TAB_1[0] = 1
- MD24120 \$MC_TRAFO_GEO_AX_ASSIGN_TAB_1[1] = 2
- MD24120 \$MC_TRAFO_GEO_AX_ASSIGN_TAB_1[2] = 3

20.4.2 Parametrierung über Geometriedaten

Baukastenprinzip

Die Parametrierung der Maschinengeometrie erfolgt nach einer Art Baukastenprinzip. Hierbei wird die Maschine sukzessive von ihrem Fußpunkt bis zur Werkzeugspitze über Geometrie-Parameter projiziert, so dass sich eine geschlossene kinematische Kette bildet. Hierbei werden Frames zur Beschreibung der Geometrie verwendet. Im Hochlauf der Steuerung werden die Konfigurationsmaschinendaten überprüft und gegebenenfalls Alarme angezeigt. Als Alarmreaktion werden alle Achsen der BAG nachgeführt. Die Alarme können nur mittels POWER ON gelöscht werden.

Wie aus dem nachfolgenden Bild "Kinematische Kette am Beispiel eines Roboters" ersichtlich ist, erfolgt über die kinematische Transformation eine Umrechnung des Arbeitspunktes des Werkzeuges (Werkzeugkoordinatensystem: X_{WZ} , Y_{WZ} , Z_{WZ}), der bezüglich des Basiskoordinatensystems (BKS = Roboter-Koordinatensystem: X_{RO} , Y_{RO} , Z_{RO}) angegeben ist, in Maschinenachswerte (MKS-Positionen: A1, A2, A3, ...). Der Arbeitspunkt (X_{WZ} , Y_{WZ} , Z_{WZ}) wird im Teileprogramm bezüglich dem zu bearbeitenden Werkstück (Werkstückkoordinatensystem WKS: X_{WS} , Y_{WS} , Z_{WS}) angegeben. Durch die programmierbaren Frames kann das Werkstückkoordinatensystem (WKS) bezüglich dem Basiskoordinatensystem (BKS) verschoben werden.

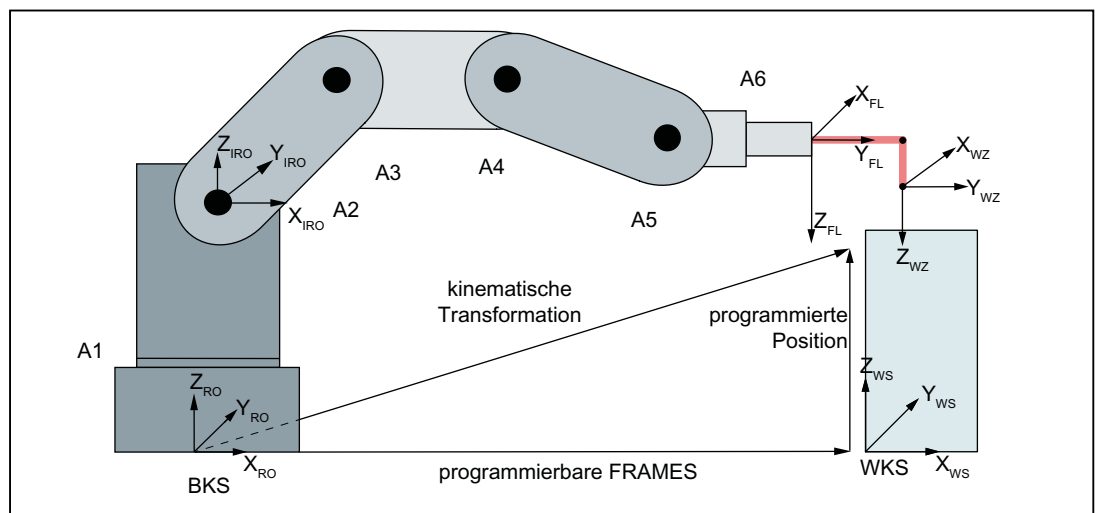


Bild 20-3 Kinematische Kette am Beispiel eines Roboters

Hinweis

Weitergehende Erläuterungen zu Koordinatensystemen entnehmen Sie bitte:

Literatur:

Funktionshandbuch Grundfunktionen;

Kapitel "K2, Achsen, Koordinatensysteme, Frames" > "Koordinatensysteme"

Folgende Maschinendaten sind zur Projektierung der kinematischen Transformation vorhanden:

Frame zwischen Fußpunkt- und internem Koordinatensystem

Das Frame T_IRO_RO verbindet den Fußpunkt der Maschine (BKS = RO) mit dem ersten von der Transformation bestimmten internen Koordinatensystem (IRO).

- MD62613 \$MC_TRAFO6_TIRORO_RPY (Frame zwischen Fußpunkt- und internem Koordinatensystem (Rotationsanteil), n = 0...2)
- MD62612 \$MC_TRAFO6_TIRORO_POS (Frame zwischen Fußpunkt- und internem Koordinatensystem (Positionsanteil), n = 0...2)

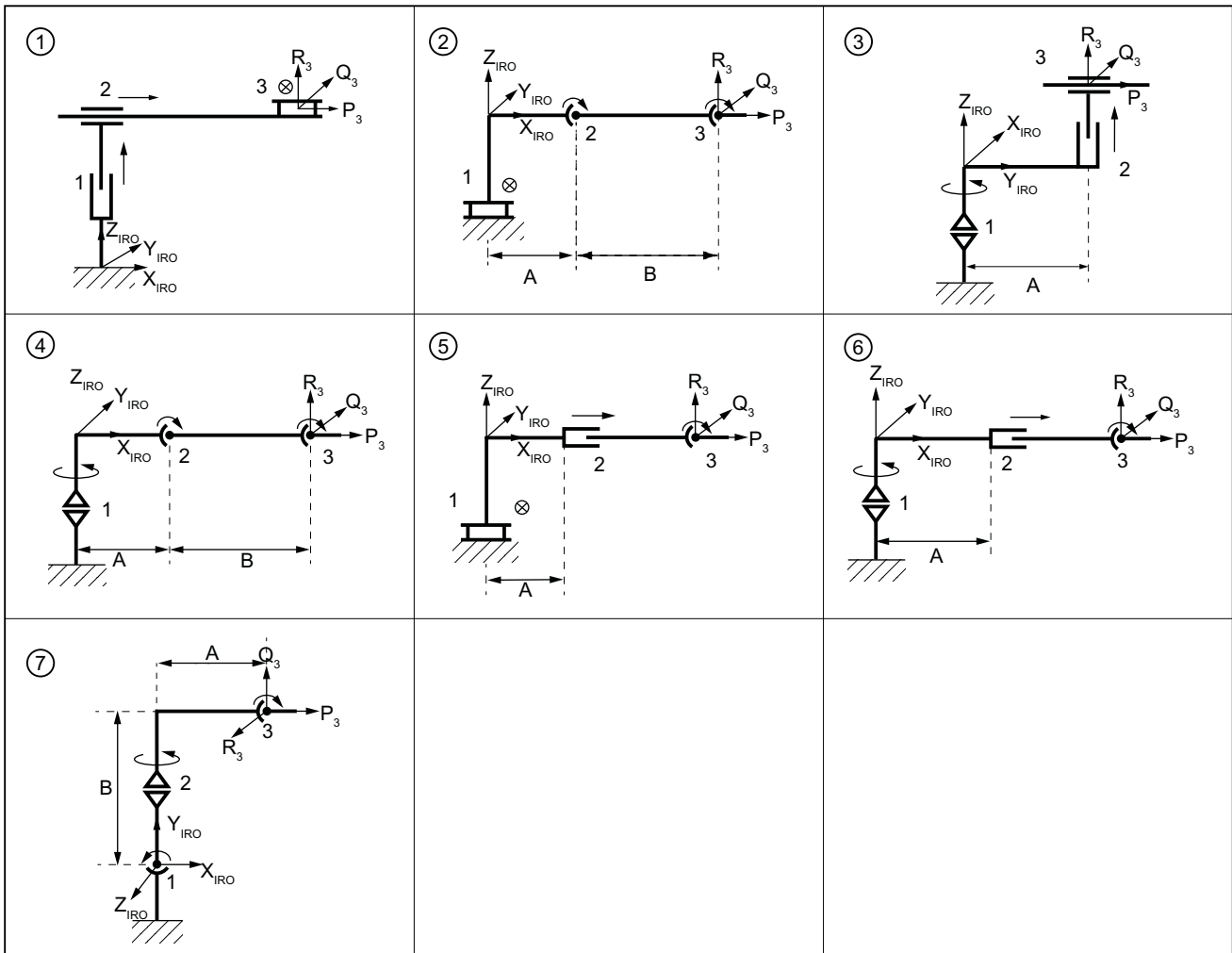
Grundachsenanordnung

Der Typ der Grundachsenanordnung wird spezifiziert mit dem Maschinendatum:

- MD62603 \$MC_TRAFO6_MAIN_AXES (Grundachsenkennung)

Als Grundachsen werden in der Regel die ersten drei Achsen bezeichnet, die in die Transformation eingehen. Sie müssen immer parallel oder senkrecht zueinander stehen.

Die im folgende Bild dargestellten Grundachskinematiken sind im Transformationspaket Handling enthalten. Jede Grundachsenanordnungen wird mit einer Kennung versehen (siehe Kapitel "Gelenkdefinition (Seite 778)").



- ① SS: MD62603 = 1, Portal (3 Linearachsen, rechtwinklig)
- ② CC: MD62603 = 2, Scara (1 Linearachse, 2 Rundachsen (parallel))
- ③ CS: MD62603 = 6, Scara (2 Linearachsen, 1 Rundachse (Drehachse))
- ④ NR: MD62603 = 3, Gelenkarm (3 Rundachsen (2 Achsen parallel))
- ⑤ SC: MD62603 = 4, Scara (2 Linearachsen, 1 Rundachse (Schwenkachse))
- ⑥ RR: MD62603 = 5, Gelenkarm (1 Linearachse, 2 Rundachsen (senkrecht))
- ⑦ NN: MD62603 = 7, Gelenkarm (3 Rundachsen)

Bild 20-4 Übersicht Grundachskonfiguration

Grundachslängen A und B

Die Grundachslängen A und B werden angegeben mit dem Maschinendatum:

- MD62607 \$MC_TRAFO6_MAIN_LENGTH_AB (Grundachslängen A und B, n = 0...1)

Diese sind, wie aus Bild "Übersicht Grundachskonfigurationen" ersichtlich ist, für jeden Grundachstyp speziell festgelegt.

Lage der 4. Achse

Ob die 4. Achse parallel/antiparallel oder senkrecht zur letzten rotatorischen Grundachse montiert ist kennzeichnet das Maschinendatum:

- MD62606 \$MC_TRAFO6_A4PAR (Achse 4 parallel/antiparallel zu letzter Grundachse)

Beschreibung der Hand

Frame zur Anbringung der Hand

Das Frame T_X3_P3 verbindet das letzte Koordinatensystem der Grundachsen mit dem ersten Koordinatensystem der Hand.

- MD62608 \$MC_TRAFO6_TX3P3_POS (Anbringung der Hand (Positionsanteil), n = 0...2)
- MD62609 \$MC_TRAFO6_TX3P3_RPY (Anbringung der Hand (Rotationsanteil), n = 0...2)

Frame zwischen Handpunkt- und Flanschkoordinatensystem

Das Frame T_FL_WP verbindet das letzte Koordinatensystem der Hand mit dem Flanschkoordinatensystem.

- MD62610 \$MC_TRAFO6_TFLWP_POS (Frame zwischen Handpunkt- und Flanschkoordinatensystem (Positionsanteil), n = 0...2)
- MD62611 \$MC_TRAFO6_TFLWP_RPY (Frame zwischen Handpunkt- und Flanschkoordinatensystem (Rotationsanteil), n = 0...2)

Diese Daten werden in den folgenden Kapiteln näher erläutert.

Beschreibung des Handtyps

Mit diesen Parametern wird der Typ der Hand beschrieben.

- MD62604 \$MC_TRAFO6_WRIST_AXES (Handachsenkennung)
Als Handachsen bezeichnet man in der Regel die Achsen vier bis sechs.

Prinzipiell enthaltene Handachsen

Als Hand werden in der Regel alle Achsen ab der vierten Achse bezeichnet. Das Transformationspaket Handling kennt nur Hände mit rotatorischen Achsen. Für dreiachsige Hände wird die Handachsenkennung eingetragen über das Maschinendatum:

- MD62604 \$MC_TRAFO6_WRIST_AXES (Handachsenkennung)

Für Hände mit weniger als drei Achsen muss die Kennung für Winkelschräghand oder Zentralhand eingetragen werden ins Maschinendatum:

- MD62604 \$MC_TRAFO6_WRIST_AXES (Handachsenkennung)

Zur Zeit werden nur die Handachstypen Zentralhand und Winkelschräghand unterstützt.

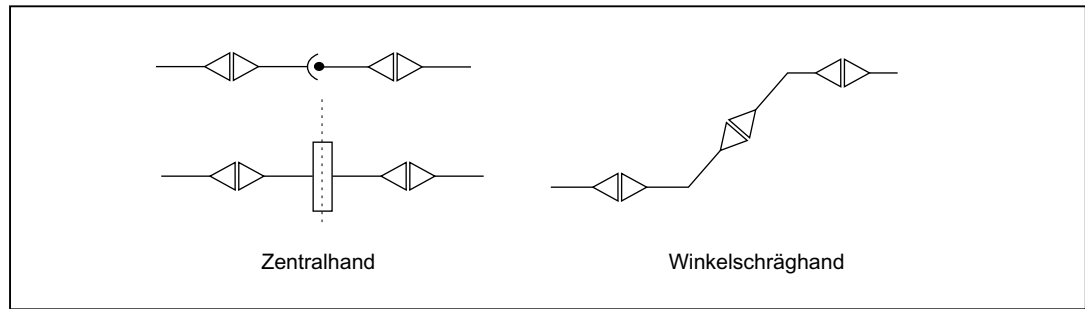


Bild 20-5 Übersicht Handachskonfiguration

Parametrierung der Handachsen

Mit den nachfolgenden Maschinendaten wird mittels einer speziellen Art von Frames die Geometrie der Hand bzw. die Lage der Koordinatensysteme in der Hand zueinander beschrieben.

- MD62614 \$MC_TRAFO6_DHPAR4_5A (Parameter A zur Projektierung der Hand, $n = 0 \dots 1$)
- MD62615 \$MC_TRAFO6_DHPAR4_5D (Parameter D zur Projektierung der Hand, $n = 0 \dots 1$)
- MD62616 \$MC_TRAFO6_DHPAR4_5ALPHA. (Parameter ALPHA zur Projektierung der Hand, $n = 0 \dots 1$)

Hierbei entsprechen die Maschinendaten bestimmten Komponenten eines Frames (siehe Kapitel "Positions- und Orientierungsbeschreibung mit Hilfe von Frames (Seite 777)"):

- X-Komponente: MD62614 \$MC_TRAFO6_DHPAR4_5A \triangleq _POS[0] (x-Komponente)
- Z-Komponente: MD62615 \$MC_TRAFO6_DHPAR4_5D \triangleq _POS[2] (z-Komponente)
- C-Winkel: MD62616 \$MC_TRAFO6_DHPAR4_5ALPHA \triangleq _RPY[2] (C-Winkel)

Die anderen Komponenten des Frames sind Null.

Zentralhand (ZEH)

Bei einer Zentralhand schneiden sich alle Handachsen in einem Punkt. Alle Parameter sind wie aus Tabelle "Projektierungsdaten Zentralhand" ersichtlich einzutragen.

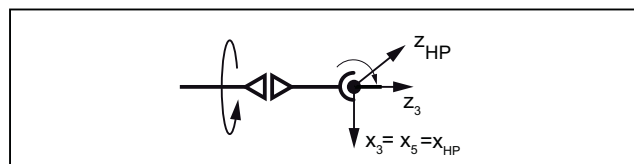


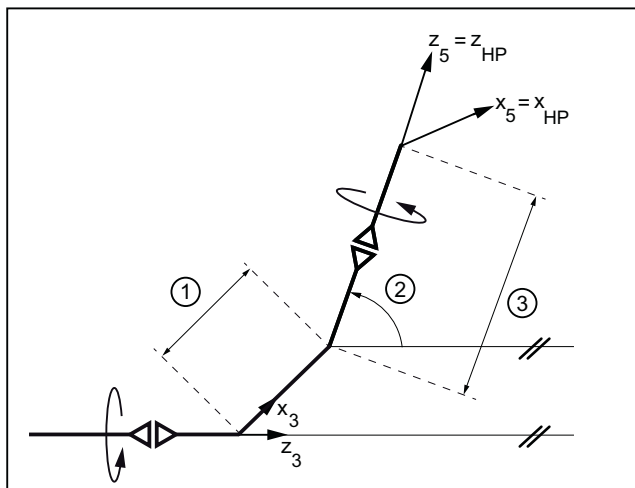
Bild 20-6 Zentralhand (ZEH) 5-Achser

Tabelle 20-1 Projektierungsdaten Zentralhand

Maschinendatum	Wert
MD62604 \$MC_TRAFO6_WRIST_AXES	2
MD62614 \$MC_TRAFO6_DHPAR4_5A	[0.0, 0.0]
MD62615 \$MC_TRAFO6_DHPAR4_5D	[0.0, 0.0]
MD62616 \$MC_TRAFO6_DHPAR4_5ALPHA	[-90.0, 90.0]

Winkelschräghand (WSH)

Die Winkelschräghand unterscheidet sich zur Zentralhand dadurch, dass sich die Achsen nicht schneiden und auch nicht senkrecht zueinander. Für diese Hand stehen die Parameter a_4 , d_5 , und α_4 wie aus Tabelle "Projektierungsdaten Zentralhand" ersichtlich zur Verfügung.



- ① MD62614 \$MC_TRAFO6_DHPAR4_5A[0]
- ② MD62616 \$MC_TRAFO6_DHPAR4_5ALPHA[0]
- ③ MD62615 \$MC_TRAFO6_DHPAR4_5D[1]

Bild 20-7 Winkelschräghand (WSH) 5-Achser

Tabelle 20-2 Projektierungsdaten Winkelschräghand (5-Achser)

Maschinendatum	Wert
MD62604 \$MC_TRAFO6_WRIST_AXES	6
MD62614 \$MC_TRAFO6_DHPAR4_5A	[a4, 0.0]
MD62615 \$MC_TRAFO6_DHPAR4_5D	[0.0, d5]
MD62616 \$MC_TRAFO6_DHPAR4_5ALPHA	[α4, 0.0]

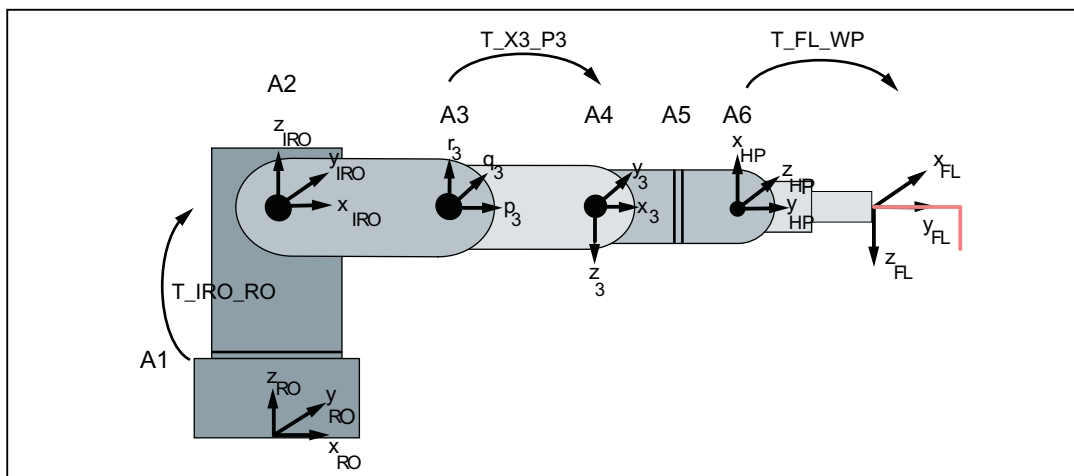


Bild 20-8 Verbindungsframes

Frame: T_IRO_RO

Das Frame T_IRO_RO verbindet das vom Anwender definierte Fußpunktkoordinatensystem (RO) mit dem internen Roboterkoordinatensystem (IRO). Das interne Roboterkoordinatensystem ist für jeden Grundachstyp über das Transformationspaket Handling fest vorgegeben und in den Kinematikbildern für die Grundachsenanordnungen eingezeichnet. Das Fußpunktkoordinatensystem liegt im kartesischen Nullpunkt der Maschine. Es entspricht dem Basiskoordinatensystem. Wenn keine FRAMES programmiert sind, entspricht das Basiskoordinatensystem dem Werkstückkoordinatensystem.

Hinweis

Weitergehende Erläuterungen zu FRAMES entnehmen Sie bitte:

Literatur:

Programmierhandbuch Grundlagen

Das Frame T_IRO_RO unterliegt keinerlei Einschränkungen für 5-achsige Kinematiken.

Bei 4-Achsern gelten folgende Einschränkungen:

- Die erste rotatorische Achse muss immer parallel/antiparallel zu einer der Koordinatenachsen des Fußpunktkoordinatensystems (RO) sein.
- Bei Grundachsen vom Typ SS gibt es keine weiteren Einschränkungen.
- Bei Grundachsen vom Typ CC, CS oder SC gibt es keine weiteren Einschränkungen, wenn die Achse 4 parallel zur letzten rotatorischen Grundachse ist.
- Bei allen anderen Grundachsen und bei Grundachsen vom Typ CC, CS oder SC, wenn die Achse 4 senkrecht zur letzten rotatorischen Grundachse ist, muss die Z-Achse von RO parallel zur Z-Achse von IRO sein.

Frame: T_X3_P3

Das Frame T_X3_P3 beschreibt die Anbringung der Hand an die Grundachsen. Mit dem Frame T_X3_P3 wird das Koordinatensystem der letzten Grundachse (p3_q3_r3-Koordinatensystem) mit dem in die erste Handachse gelegten Koordinatensystem (x3_y3_z3-Koordinatensystem) verbunden. Das p3_q3_r3-Koordinatensystem ist in den Kinematikbildern für die Grundachsenanordnungen eingezeichnet.

Die z3-Achse liegt immer auf der 4. Achse.

Entsprechend der Anzahl der Achsen, die in die Transformation eingehen, unterliegt der Frame T_X3_P3 gewissen grund- und handachsenspezifischen Einschränkungen:

- Für 5-Achser ist das Frame T_X3_P3 in folgenden Fällen frei wählbar:
 - Wenn die Grundachsen vom Typ SS sind.
 - Wenn die Grundachsen vom Typ CC, CS oder SC sind, so muss entweder eine Zentralhand (ZEH) vorliegen, oder die 4. Achse muss parallel zur letzten rotatorischen Grundachse sein.
 - Wenn die Grundachsen vom Typ NR oder RR sind, so muss entweder eine Zentralhand (ZEH) vorliegen, oder die 4. Achse muss parallel zur letzten rotatorischen Grundachse sein und X-Flansch muss die 5. Achse schneiden.
 - Sind die Grundachsen vom Typ NN, so muss eine Zentralhand vorliegen.
- Bei 4-Achsern ist zu beachten, dass die z3-Achse immer parallel/antiparallel oder senkrecht zur letzten Grundachse ist.

Frame: T_FL_WP

Mit dem Frame T_FL_WP wird der Flansch mit dem letzten internen, über das Transformationspaket Handling fest vorgegebenen Koordinatensystem (Handpunktkoordinatensystem) verbunden.

Dieser Frame unterliegt für Kinematiken mit weniger als 6 Achsen gewissen Einschränkungen, die bei den jeweiligen Kinematiken erläutert werden.

Sonstige Projektierungsdaten

Anzahl der transformierten Achsen

Wie viele Achsen in die Transformation eingehen wird festgelegt mit dem Maschinendatum:

- MD62605 \$MC_TRAFO6_NUM_AXES (Anzahl der transformierten Achsen)

Die Anzahl der transformierten Achsen kann zurzeit zwischen 2 und 6 Achsen betragen.

Änderung der Achsreihenfolge

Umordnung von Achsen: MD62620

Hinweis

Bei bestimmten Kinematiken sind Vertauschungen von Achsen möglich, ohne dass sich ein anderes kinematisches Verhalten ergibt. Um diese Kinematiken ineinander überzuführen, gibt es das Maschinendatum:

MD62620 \$MC_TRAFO6_AXIS_SEQ (Umordnung von Achsen)

Dabei sind die Achsen an der Maschine mit 1 bis 6 durchnummeriert und müssen in der internen Reihenfolge eingetragen werden in die:

MD62620 \$MC_TRAFO6_AXIS_SEQ[0] ...[4]

Alle weiteren achsspezifischen Maschinendaten beziehen sich auf die an der Maschine vorhandene Achsreihenfolge.

Tabelle 20-3 Änderung der Achsreihenfolge

Grundachskinematik	Vertauschungsmöglichkeiten
SS, CC	beliebig
SC	1 und 2
CS	2 und 3

Beispiel 1

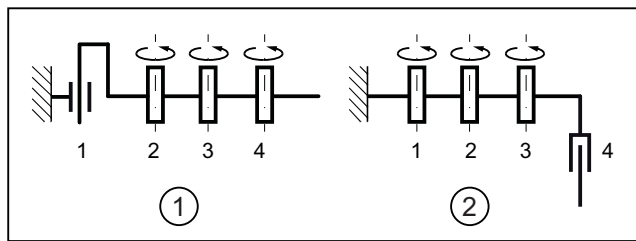
Es seien zwei Kinematiken wie in Bild "Umordnen von Achsen 1" zu sehen gegeben. Kinematik 1 ist direkt im Transformationspaket Handling enthalten. Sie entspricht einer CC-Kinematik mit einer Handachse parallel zur letzten rotatorischen Grundachse.

Kinematik 2 ist äquivalent zu Kinematik 1, da es für die resultierende Bewegung des Roboters unerheblich ist, ob die translatorische Achse die Achse 1 oder die Achse 4 ist. In diesem Fall müssen die Daten für Kinematik 2 eingegeben werden im Maschinendatum:

- MD62620 \$MC_TRAFO6_AXIS_SEQ (Umordnung von Achsen)

Die Eingabe erfolgt folgendermaßen:

- MD62620 \$MC_TRAFO6_AXIS_SEQ[0] = 4
- MD62620 \$MC_TRAFO6_AXIS_SEQ[1] = 1
- MD62620 \$MC_TRAFO6_AXIS_SEQ[2] = 2
- MD62620 \$MC_TRAFO6_AXIS_SEQ[3] = 3



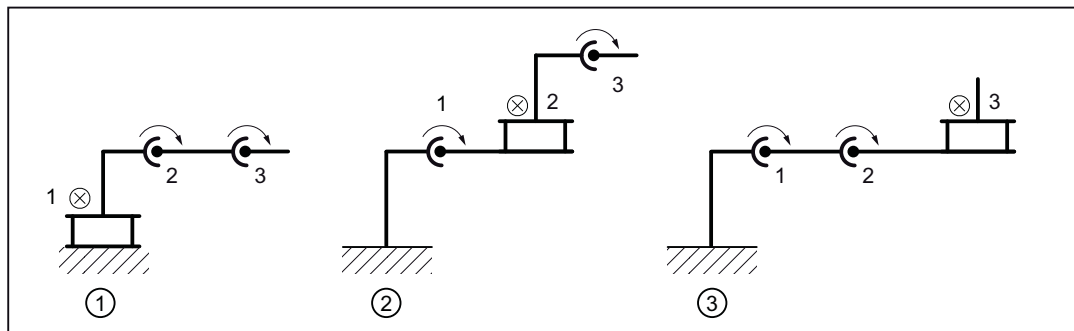
① Kinematik 1

② Kinematik 2

Bild 20-9 Umordnen von Achsen 1

Beispiel 2

Bei einer SCARA-Kinematik nach Bild "Umordnen von Achsen 2" können die Achsen beliebig vertauscht werden. Kinematik 1 ist direkt im Transformationspaket Handling enthalten. Sie entspricht einer CC-Kinematik. Es ist für den Fall der Achsvertauschung nicht relevant wie viele Handachsen beteiligt sind.



① Kinematik 1: Achsfolge 1 2 3

② Kinematik 2: Achsfolge 2 1 3

③ Kinematik 3: Achsfolge 2 3 1

Bild 20-10 Umordnen von Achsen 2

Änderungen der Achsrichtungen

Über das Transformationspaket Handling ist für jede Achse eine feste Dreh- oder Verschieberichtung vorgegeben. Diese Richtung stimmt nicht unbedingt mit der entsprechenden Richtung an der Maschine überein. Um sie anzugleichen, muss im folgenden Maschinendatum für die jeweilige Achse eine **-1** eingegeben werden, wenn die Richtung umgedreht werden soll, sonst eine **+1**:

- MD62618 \$MC_TRAFO6_AXES_DIR[] (Anpassung der physikalischen und mathematischen Drehrichtung [Achse-Nr.]: 0...5)

Anpassung der Nullpunkte der Achsen

Verschiebung des mathematischen zum mechanischen Nullpunkt

Die mathematischen Nullpunkte der Achsen sind über das Transformationspaket Handling fest vorgegeben. Die mathematische Nullstellung stimmt aber nicht immer mit der mechanischen Nullstellung (Justagestellung) der Achsen überein. Um die Nullstellungen einander anzupassen, muss im folgenden Maschinendatum für jede Achse die Abweichung zwischen der mathematischen Nullstellung und dem Justagepunkt eingetragen werden:

- MD62617 \$MC_TRAFO6_MAMES[] (Verschiebung mathematischer zu mechanischer Nullpunkt [Achse-Nr.]: 0...5)

Hierbei muss die Abweichung ausgehend von der mechanischen Nullstellung bezogen auf die mathematisch positive Drehrichtung der Achse eingetragen werden.

Beispiel

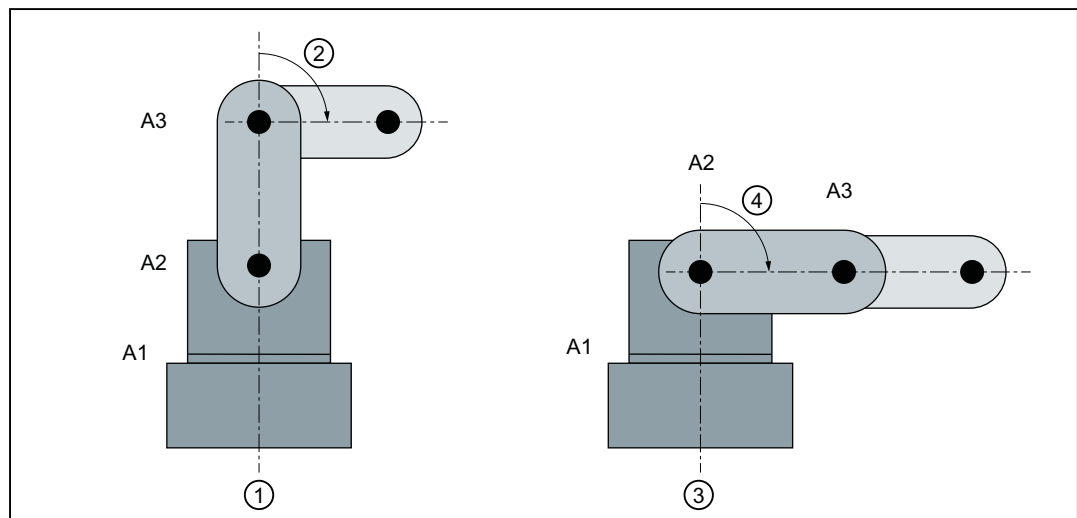
Im nachfolgenden Bild ist eine Gelenkarmkinematik dargestellt.

Achse 2 (A2) hat in der mathematischen Nullstellung ② den Wert 90°. Dieser Wert ist für Achse 2 (Index 1) in das Maschinendatum einzutragen:

- MD62617 \$MC_TRAFO6_MAMES[1] = 90

Achse 3 (A3) wird relativ zur in der kinematischen Kette vorhergehenden Achse 2 (A2) gezählt und hat in der mechanischen Nullstellung ① den Wert: -90°:

- MD62617 \$MC_TRAFO6_MAMES[2] = -90



- ① mechanische Nullstellung (Justagestellung):
- ② MD62617 \$MC_TRAFO6_MAMES[2] = -90
- ③ mathematische Nullstellung
- ④ MD62617 \$MC_TRAFO6_MAMES[1] = 90

Bild 20-11 Anpassung an mechanische Nullstellung

Achstyp für die Transformation

Um welchen Achstyp es sich handelt, wird angegeben mit dem Maschinendatum:

- MD62601 \$MC_TRAFO6_AXES_TYPE (Achstyp für Transformation [Achse-Nr.]: 0...5)

Die Transformation unterscheidet nach folgenden Achstypen:

- Linearachse: MD62601 = 1
- Rundachse: MD62601 = 3

Geschwindigkeiten und Beschleunigungen für Dynamik-begrenzende Override-Regelung

VORSICHT

Unnötig stark begrenzende Defaultwerte

Seit dem Softwarestand **RCTRA 07.05.00** berücksichtigt die RCTRA Transformation für die Begrenzung der kartesischen Geschwindigkeiten und Beschleunigungen die Maschinendaten **MD62629** bis **MD62632**.

Um unnötige Begrenzungen der Geschwindigkeiten und Beschleunigungen zu vermeiden, müssen diese Maschinendaten an die Fähigkeiten der Maschine angepasst werden. Da diese Daten bisher nicht verwendet wurden, sind dort häufig noch die unnötig stark begrenzenden Defaultwerte eingestellt.

Für das Verfahren der Achsen mit aktiver RCTRA-Transformation und Dynamik-begrenzender Override-Regelung werden eigene Ersatzgrenzwerte für Geschwindigkeiten und Beschleunigungen für die kartesischen Bewegungskomponenten eingeführt.

Der Override-Regler innerhalb der Transformation ist immer dann aktiv, wenn Bewegungen interpoliert werden, deren Dynamikgrenzen nicht oder nicht vollständig durch die Bewegungsplanung im Look-Ahead der Steuerung abgesichert werden.

In den folgenden Fällen ist der Geschwindigkeitsregler aktiv:

- Beim Verfahren in der Betriebsart JOG mit Transformation
- Bei Bewegungsüberlagerungen in Echtzeit, z. B. über die Systemvariable \$AA_OFF[X]
- Bei Geschwindigkeitsbegrenzungen durch Safety-Integrated (SG-Stufe aktiv)

Kartesische Geschwindigkeit

Die Geschwindigkeiten für die einzelnen translatorischen Bewegungsrichtungen beim Verfahren mit Override-Regler werden vorgegeben werden mit dem Maschinendatum:

- MD62629 \$MC_TRAFO6_VELCP[i] (kartesische Geschwindigkeit [Nr.]: 0...2)
 - Index i = 0: X-Komponente des Basis-Systems
 - Index i = 1: Y-Komponente des Basis-Systems
 - Index i = 2: Z-Komponente des Basis-Systems

Kartesische Beschleunigungen

Die Beschleunigungen für die einzelnen translatorischen Bewegungsrichtungen beim Verfahren mit Override-Regler werden vorgegeben werden mit dem Maschinendatum:

- MD62630 \$MC_TRAFO6_ACCCP[i] (kartesische Beschleunigungen [Nr.]: 0...2)
 - Index i = 0: X-Komponente des Basis-Systems
 - Index i = 1: Y-Komponente des Basis-Systems
 - Index i = 2: Z-Komponente des Basis-Systems

Orientierungswinkel-Geschwindigkeiten

Die Geschwindigkeiten für die einzelnen Orientierungsrichtungen beim Verfahren mit Override-Regler werden vorgegeben werden mit dem Maschinendatum:

- MD62631 \$MC_TRAFO6_VELORI[i] (Orientierungswinkel-Geschwindigkeiten [Nr.]: 0...2)
 - Index i = 0: A-Winkel
 - Index i = 1: B-Winkel
 - Index i = 2: C-Winkel

Orientierungswinkel-Beschleunigungen

Die Beschleunigungen für die einzelnen Orientierungsrichtungen beim Verfahren mit Override-Regler werden vorgegeben werden mit dem Maschinendatum:

- MD62632 \$MC_TRAFO6_ACCORI[i] (Orientierungswinkel-Beschleunigungen [Nr.]: 0...2)
 - Index i = 0: A-Winkel
 - Index i = 1: B-Winkel
 - Index i = 2: C-Winkel

Reduzierfaktor für den Geschwindigkeitsregler

Der Reduzierfaktor für den Geschwindigkeitsregler in der Betriebsart JOG wird vorgegeben mit dem Maschinendatum:

- MD62633 \$MC_ROBX_DYN_LIM_REDUCE (Reduzierfaktor für Geschwindigkeitsregler)

Zeitkonstante für den Geschwindigkeitsregler

Die Zeitkonstante für den PT1-Filter des Geschwindigkeitsreglers wird vorgegeben mit dem Maschinendatum:

- MD62634 \$MC_ROBX_VEL_FILTER_TIME (Zeitkonstante für Geschwindigkeitsregler)

20.5 Kinematikbeschreibungen

Die folgenden Kinematikbeschreibungen für 2- bis 5-Achs-Kinematiken beschreiben zuerst das allgemeine Vorgehen bei der Projektierung und erläutern dann anhand eines Projektierungsbeispiels für jeden Kinematiktyp, wie die Maschinendaten projektiert werden müssen. In diesen Beispielen sind nicht alle möglichen Längen und Versätze eingezeichnet. Die Richtungsangaben beziehen sich auf die für die Transformation positiven Verfahr- und Drehrichtungen. Die Achsstellungen entsprechen der Nullstellung der Achsen für die jeweilige Transformation.

20.5.1 3-Achs-Kinematiken

3-Achs Kinematiken haben normalerweise drei translatorische Freiheitsgrade. Sie haben keinen rotatorischen Freiheitsgrad für die Orientierung. Daher haben 3-Achs Kinematiken nur Grundachsen.

Projektierung

Folgendes Vorgehen ist zur Projektierung einer 3-Achs Kinematik notwendig:

1. Kinematikklasse "Standard" eintragen in Maschinendatum:
MD62600 \$MC_TRAFO6_KINCLASS (Kinematikklasse)
2. Anzahl der Achsen für die Transformation im Maschinendatum:
MD62605 \$MC_TRAFO6_NUM_AXES = 3 (Anzahl der transformierten Achsen)
3. Vergleich der Grundachsen mit dem im Transformationspaket Handling enthaltenen Grundachsen. → Eintragen der Grundachskennung in Maschinendatum:
MD62603 \$MC_TRAFO6_MAIN_AXES (Grundachsenkennung)
4. Wenn eine zur normalen Achsreihenfolge unterschiedliche Reihenfolge vorliegt, muss sie korrigiert werden im Maschinendatum:
MD62620 \$MC_TRAFO6_AXIS_SEQ (Umordnung von Achsen)
5. Als Kennung für die Handachsen muss das folgende Maschinendatum gleich 1 gesetzt werden (keine Hand):
MD62604 \$MC_TRAFO6_WRIST_AXES = 1 (Handachsenkennung)
6. Eintragung der Achstypen für die Transformation im Maschinendatum:
MD62601 \$MC_TRAFO6_AXES_TYPE (Achstyp für Transformation)
7. Vergleich der Drehrichtungen der Achsen mit den durch das Transformationspaket Handling vorgegebenen Drehrichtungen und Korrektur in Maschinendatum:
MD62618 \$MC_TRAFO6_AXES_DIR (Anpassung der physikalischen und mathematischen Drehrichtung)
8. Eintragen der mechanischen Nullpunktverschiebung in Maschinendatum:
MD62617 \$MC_TRAFO6_MAMES (Verschiebung mathematischer zu mechanischer Nullpunkt)
9. Eintragen der Grundachslängen in Maschinendatum:
MD62607 \$MC_TRAFO6_MAIN_LENGTH_AB (Grundachslängen A und B)

10. Bestimmung des Frames T_IRO_RO und Eintragung der Verschiebung in Maschinendatum:
 Maschinendatum:
 MD62612 \$MC_TRAFO6_TIRORO_POS (Frame zwischen Fußpunkt und internem System (Positionsanteil))
 Eintragen der Verdrehung in Maschinendatum:
 MD62613 \$MC_TRAFO6_TIRORO_RPY (Frame zwischen Fußpunkt und internem System (Rotationsanteil))
11. Bestimmung des Flanschkoordinatensystems. Hierzu ist das p3_q3_r3-Koordinatensystem als Ausgangssystem zu sehen. Die Verschiebung wird eingetragen in Maschinendatum:
 MD62610 \$MC_TRAFO6_TFLWP_POS (Frame zwischen Handpunkt und Flansch (Positionsanteil))
 Die Verdrehung wird eingetragen in Maschinendatum:
 MD62611 \$MC_TRAFO6_TFLWP_RPY (Frame zwischen Handpunkt und Flansch (Rotationsanteil))

SCARA-Kinematiken

SCARA-Kinematiken zeichnen sich dadurch aus, dass sie sowohl translatorische als auch rotatorische Achsen besitzen. Je nachdem, wie die Grundachsen zueinander stehen, werden die Kinematiken unterschiedlich bezeichnet, z.B.: CC-, SC-, CS-Kinematik (siehe Kapitel "Gelenkdefinition (Seite 778)").

3-Achser CC-Kinematik

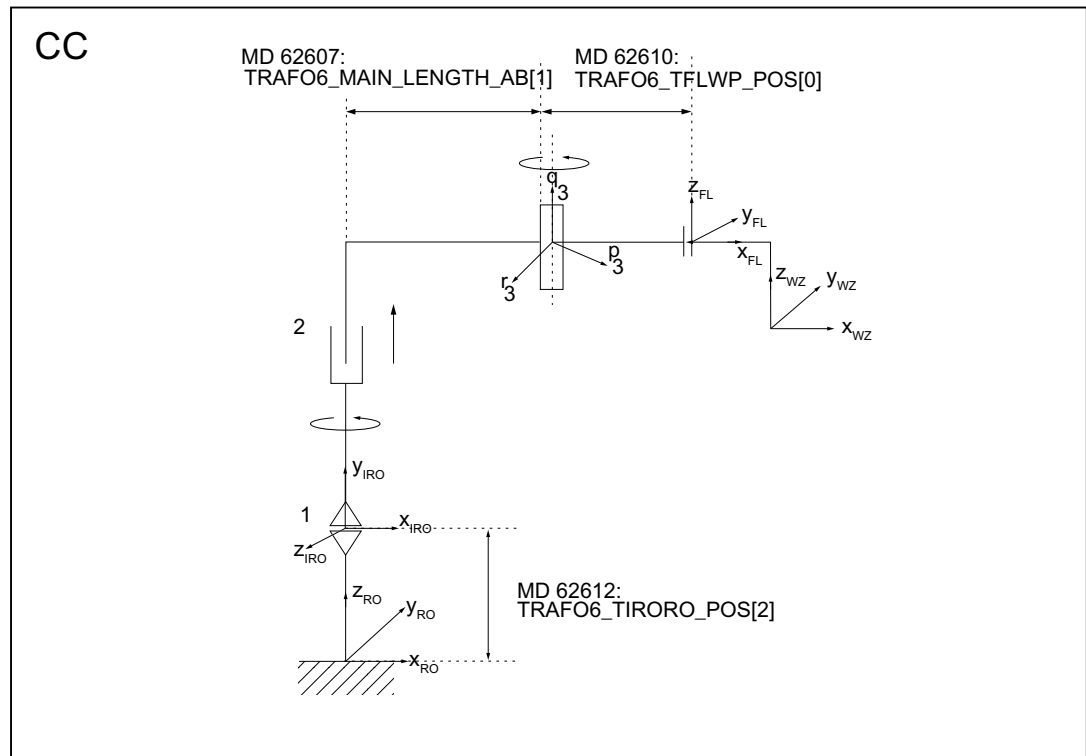


Bild 20-12 3-Achser CC-Kinematik

Tabelle 20-4 Projektierungsdaten 3-Achser CC-Kinematik

Maschinendatum	Wert
MD62600 \$MC_TRAFO6_KINCLASS	1
MD62605 \$MC_TRAFO6_NUM_AXES	3
MD62603 \$MC_TRAFO6_MAIN_AXES	2
MD62604 \$MC_TRAFO6_WRIST_AXES	1
MD62601 \$MC_TRAFO6_AXES_TYPE	[3, 1, 3, ...]
MD62620 \$MC_TRAFO6_AXIS_SEQ	[2, 1, 3, 4, 5, 6]
MD62618 \$MC_TRAFO6_AXES_DIR	[1, 1, 1, 1, 1, 1]
MD62617 \$MC_TRAFO6_MAMES	[0.0, 0.0, 0.0, 0.0, 0.0, 0.0]
MD62607 \$MC_TRAFO6_MAIN_LENGTH_AB	[0.0, 300.0]
MD62612 \$MC_TRAFO6_TIRORO_POS	[0.0, 0.0, 500.0]
MD62613 \$MC_TRAFO6_TIRORO_RPY	[0.0, 0.0, 90.0]
MD62608 \$MC_TRAFO6_TX3P3_POS	[0.0, 0.0, 0.0]
MD62609 \$MC_TRAFO6_TX3P3_RPY	[0.0, 0.0, 0.0]
MD62610 \$MC_TRAFO6_TFLWP_POS	[200.0, 0.0, 0.0]
MD62611 \$MC_TRAFO6_TFLWP_RPY	[0.0, 0.0, -90.0]

3-Achser SC-Kinematik

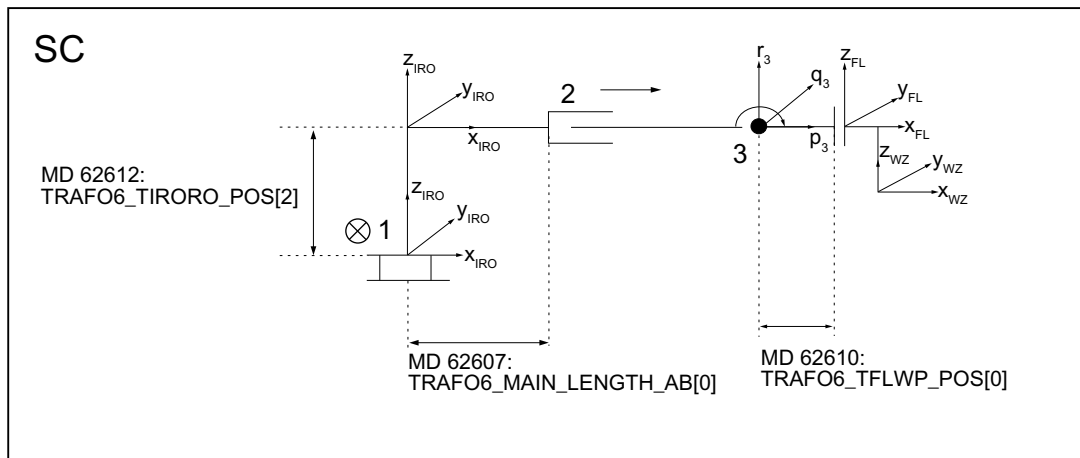


Bild 20-13 3-Achser SC-Kinematik

Tabelle 20-5 Projektierungsdaten 3-Achser SC-Kinematik

Maschinendatum	Wert
MD62600 \$MC_TRAFO6_KINCLASS	1
MD62605 \$MC_TRAFO6_NUM_AXES	3
MD62603 \$MC_TRAFO6_MAIN_AXES	4
MD62604 \$MC_TRAFO6_WRIST_AXES	1
MD62601 \$MC_TRAFO6_AXES_TYPE	[1, 1, 3, ...]
MD62620 \$MC_TRAFO6_AXIS_SEQ	[1, 2, 3, 4, 5, 6]
MD62618 \$MC_TRAFO6_AXES_DIR	[1, 1, 1, 1, 1, 1]

Maschinendatum	Wert
MD62617 \$MC_TRAFO6_MAMES	[0.0, 0.0, 0.0, 0.0, 0.0, 0.0]
MD62607 \$MC_TRAFO6_MAIN_LENGTH_AB	[500.0, 0.0]
MD62612 \$MC_TRAFO6_TIRORO_POS	[0.0, 0.0, 500.0]
MD62613 \$MC_TRAFO6_TIRORO_RPY	[0.0, 0.0, 0.0]
MD62608 \$MC_TRAFO6_TX3P3_POS	[0.0, 0.0, 0.0]
MD62609 \$MC_TRAFO6_TX3P3_RPY	[0.0, 0.0, 0.0]
MD62610 \$MC_TRAFO6_TFLWP_POS	[300.0, 0.0, 0.0]
MD62611 \$MC_TRAFO6_TFLWP_RPY	[0.0, 0.0, 0.0]

3-Achser CS-Kinematik

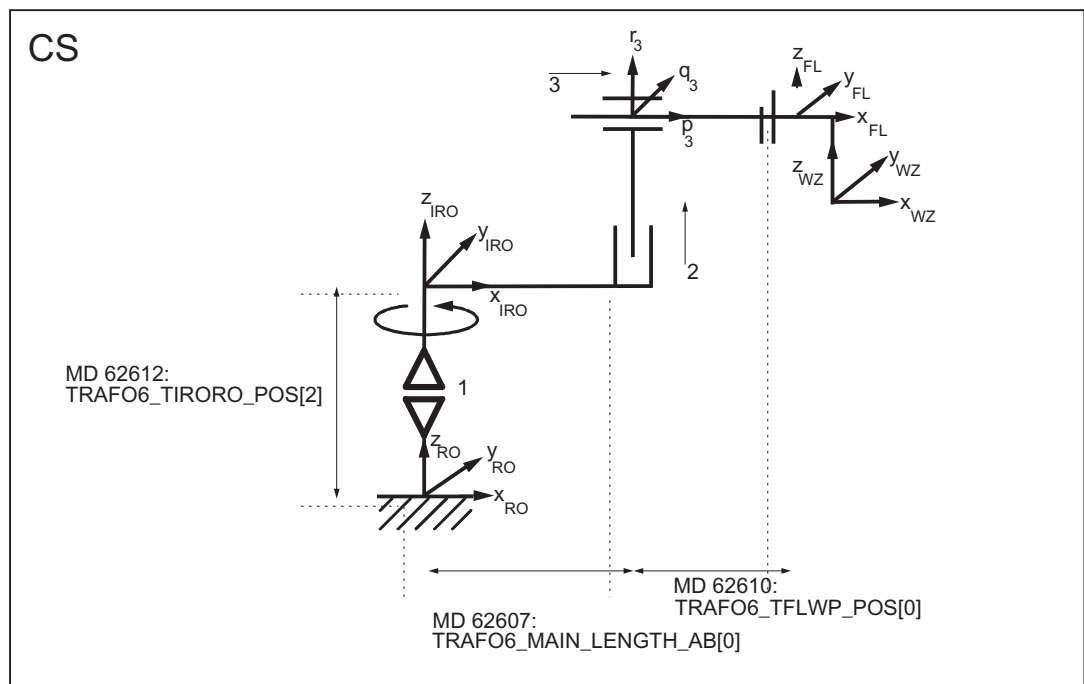


Bild 20-14 3-Achs CS-Kinematik

Tabelle 20-6 Projektierungsdaten 3-Achser CS-Kinematik

Maschinendatum	Wert
MD62600 \$MC_TRAFO6_KINCLASS	1
MD62605 \$MC_TRAFO6_NUM_AXES	3
MD62603 \$MC_TRAFO6_MAIN_AXES	6
MD62604 \$MC_TRAFO6_WRIST_AXES	1
MD62601 \$MC_TRAFO6_AXES_TYPE	[3, 1, 1, ...]
MD62620 \$MC_TRAFO6_AXIS_SEQ	[1, 2, 3, 4, 5, 6]
MD62618 \$MC_TRAFO6_AXES_DIR	[1, 1, 1, 1, 1, 1]
MD62617 \$MC_TRAFO6_MAMES	[0.0, 0.0, 0.0, 0.0, 0.0, 0.0]
MD62607 \$MC_TRAFO6_MAIN_LENGTH_AB	[500.0, 0.0]

Maschinendatum	Wert
MD62612 \$MC_TRAFO6_TIRORO_POS	[0.0, 0.0, 500.0]
MD62613 \$MC_TRAFO6_TIRORO_RPY	[0.0, 0.0, 0.0]
MD62608 \$MC_TRAFO6_TX3P3_POS	[0.0, 0.0, 0.0]
MD62609 \$MC_TRAFO6_TX3P3_RPY	[0.0, 0.0, 0.0]
MD62610 \$MC_TRAFO6_TFLWP_POS	[300.0, 0.0, 0.0]
MD62611 \$MC_TRAFO6_TFLWP_RPY	[0.0, 0.0, 0.0]

Gelenkarm-Kinematiken

3-Achser NR-Kinematik

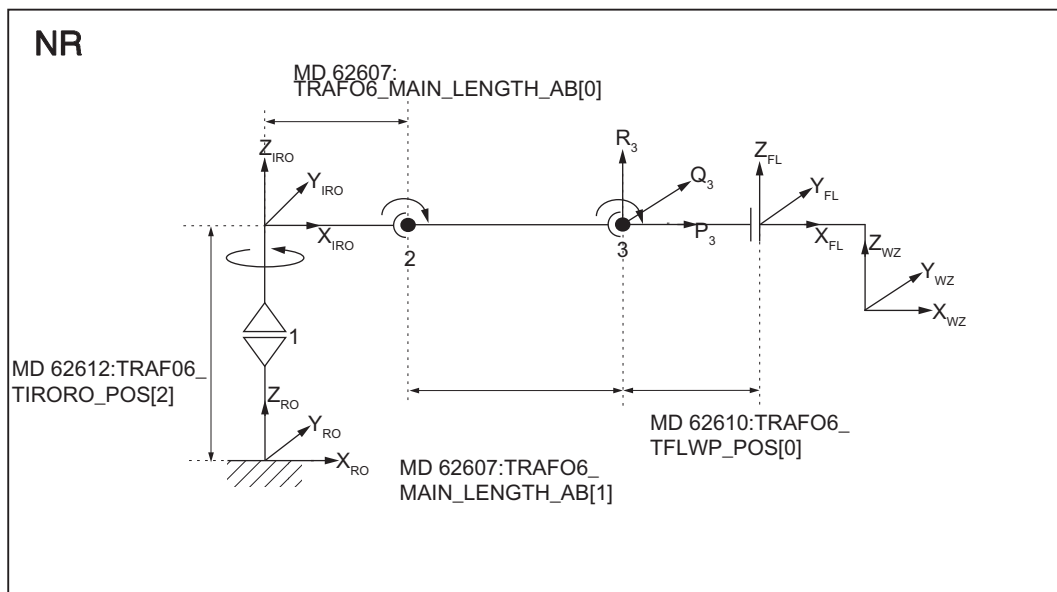


Bild 20-15 3-Achser NR-Kinematik

Tabelle 20-7 Projektierungsdaten 3-Achser NR-Kinematik

Maschinendatum	Wert
MD62600 \$MC_TRAFO6_KINCLASS	1
MD62605 \$MC_TRAFO6_NUM_AXES	3
MD62603 \$MC_TRAFO6_MAIN_AXES	3
MD62604 \$MC_TRAFO6_WRIST_AXES	1
MD62601 \$MC_TRAFO6_AXES_TYPE	[3, 3, 3, ...]
MD62620 \$MC_TRAFO6_AXIS_SEQ	[1, 2, 3, 4, 5, 6]
MD62618 \$MC_TRAFO6_AXES_DIR	[1, 1, 1, 1, 1, 1]
MD62617 \$MC_TRAFO6_MAMES	[0.0, 0.0, 0.0, 0.0, 0.0, 0.0]
MD62607 \$MC_TRAFO6_MAIN_LENGTH_AB	[300.0, 500.0]
MD62612 \$MC_TRAFO6_TIRORO_POS	[0.0, 0.0, 500.0]
MD62613 \$MC_TRAFO6_TIRORO_RPY	[0.0, 0.0, 0.0]
MD62608 \$MC_TRAFO6_TX3P3_POS	[0.0, 0.0, 0.0]

Maschinendatum	Wert
MD62609 \$MC_TRAFO6_TX3P3_RPY	[0.0, 0.0, 0.0]
MD62610 \$MC_TRAFO6_TFLWP_POS	[300.0, 0.0, 0.0]
MD62611 \$MC_TRAFO6_TFLWP_RPY	[0.0, 0.0, 0.0]

3-Achser RR-Kinematik

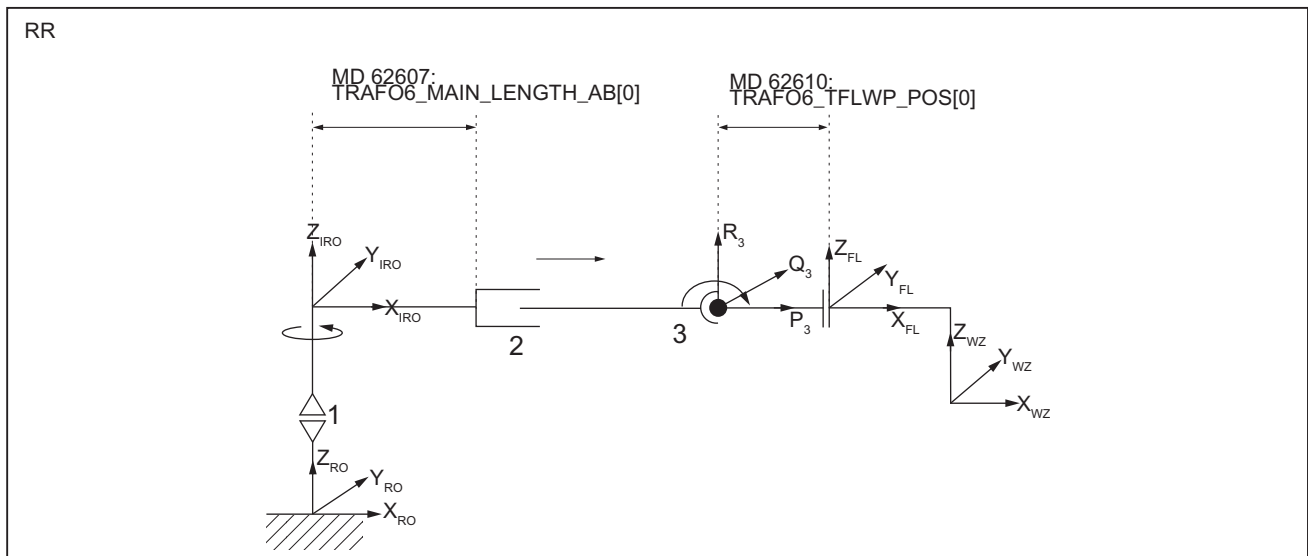


Bild 20-16 3-Achser RR-Kinematik

Tabelle 20-8 Projektierungsdaten 3-Achser RR-Kinematik

Maschinendatum	Wert
MD62600 \$MC_TRAFO6_KINCLASS	1
MD62605 \$MC_TRAFO6_NUM_AXES	3
MD62603 \$MC_TRAFO6_MAIN_AXES	5
MD62604 \$MC_TRAFO6_WRIST_AXES	1
MD62601 \$MC_TRAFO6_AXES_TYPE	[3, 1, 3, ...]
MD62620 \$MC_TRAFO6_AXIS_SEQ	[1, 2, 3, 4, 5, 6]
MD62618 \$MC_TRAFO6_AXES_DIR	[1, 1, 1, 1, 1, 1]
MD62617 \$MC_TRAFO6_MAMES	[0.0, 0.0, 0.0, 0.0, 0.0, 0.0]
MD62607 \$MC_TRAFO6_MAIN_LENGTH_AB	[300.0, 0.0]
MD62612 \$MC_TRAFO6_TIRORO_POS	[0.0, 0.0, 300.0]
MD62613 \$MC_TRAFO6_TIRORO_RPY	[0.0, 0.0, 0.0]
MD62608 \$MC_TRAFO6_TX3P3_POS	[0.0, 0.0, 0.0]
MD62609 \$MC_TRAFO6_TX3P3_RPY	[0.0, 0.0, 0.0]
MD62610 \$MC_TRAFO6_TFLWP_POS	[200.0, 0.0, 0.0]
MD62611 \$MC_TRAFO6_TFLWP_RPY	[0.0, 0.0, 0.0]

3-Achser NN-Kinematik

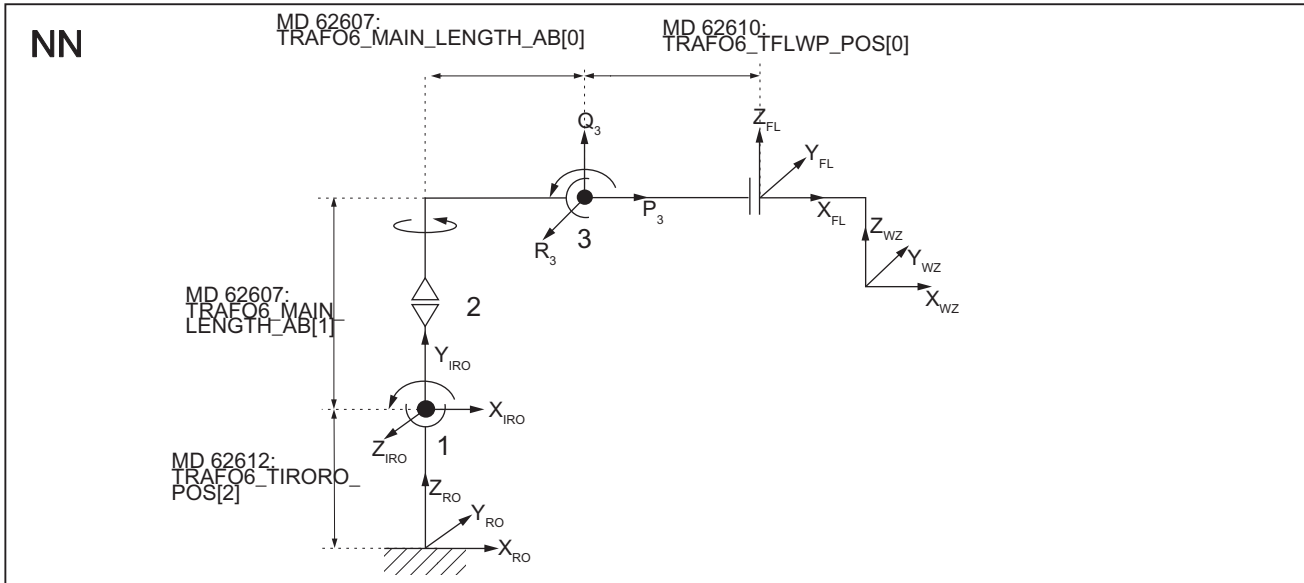


Bild 20-17 3-Achser NN-Kinematik

Tabelle 20-9 Projektierungsdaten 3-Achser NN-Kinematik

Maschinendatum	Wert
MD62600 \$MC_TRAFO6_KINCLASS	1
MD62605 \$MC_TRAFO6_NUM_AXES	3
MD62603 \$MC_TRAFO6_MAIN_AXES	7
MD62604 \$MC_TRAFO6_WRIST_AXES	1
MD62601 \$MC_TRAFO6_AXES_TYPE	[3, 3, 3, ...]
MD62620 \$MC_TRAFO6_AXIS_SEQ	[1, 2, 3, 4, 5, 6]
MD62618 \$MC_TRAFO6_AXES_DIR	[1, 1, 1, 1, 1, 1]
MD62617 \$MC_TRAFO6_MAMES	[0.0, 0.0, 0.0, 0.0, 0.0, 0.0]
MD62607 \$MC_TRAFO6_MAIN_LENGTH_AB	[300.0, 500.0]
MD62612 \$MC_TRAFO6_TIRORO_POS	[0.0, 0.0, 300.0]
MD62613 \$MC_TRAFO6_TIRORO_RPY	[0.0, 0.0, 90.0]
MD62608 \$MC_TRAFO6_TX3P3_POS	[0.0, 0.0, 0.0]
MD62609 \$MC_TRAFO6_TX3P3_RPYFehler! Textmarke nicht definiert.	[0.0, 0.0, 0.0]
MD62610 \$MC_TRAFO6_TFLWP_POS	[400.0, 0.0, 0.0]
MD62611 \$MC_TRAFO6_TFLWP_RPY	[0.0, 0.0, -90.0]

20.5.2 4-Achs-Kinematiken

4-Achs Kinematiken haben normalerweise 3 translatorische Freiheitsgrade und einen rotatorischen Freiheitsgrad für die Orientierung.

Einschränkungen

4-Achs Kinematiken haben folgende Einschränkungen:

Das Frame T_FL_WP ist der folgenden Bedingung unterworfen:

- MD62611 \$MC_TRAFO6_TFLWP_RPY = [0.0, 90.0, 0.0] (Frame zwischen Handpunkt und Flansch (Rotationsanteil))
- X-Flansch und X-Werkzeug müssen parallel zur 4.Achse sein.
- Zwei aufeinander folgende Grundachsen müssen parallel oder orthogonal sein.
- Die 4. Achse darf nur parallel oder orthogonal an die letzte Grundachse montiert werden.

Projektierung

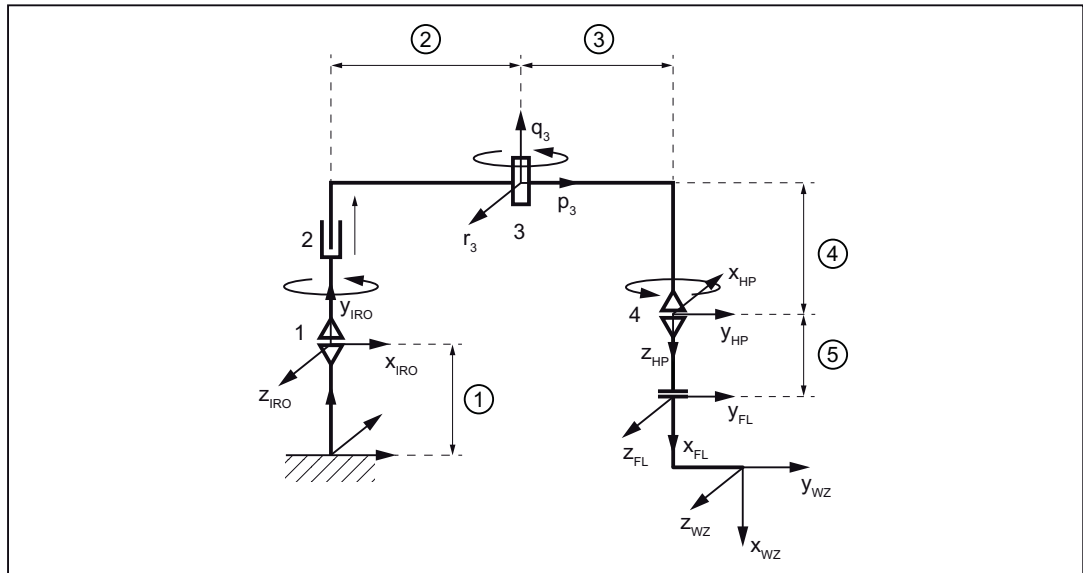
Folgendes Vorgehen ist zur Projektierung einer 4-Achs Kinematik notwendig:

1. Kinematikkategorie "Standard" eintragen in das Maschinendatum:
MD62600 \$MC_TRAFO6_KINCLASS (Kinematikkategorie)
2. Anzahl der Achsen für die Transformation eintragen in das Maschinendatum:
MD62605 \$MC_TRAFO6_NUM_AXES=4 (Anzahl der transformierten Achsen)
3. Vergleich der Grundachsen mit dem im Transformationspaket Handling enthaltenen Grundachsen.
 - Eintragen der Grundachsenkennung in Maschinendatum:
MD62603 \$MC_TRAFO6_MAIN_AXES (Grundachsenkennung)
4. Wenn eine zur normalen Achsreihenfolge unterschiedliche Reihenfolge vorliegt, muss sie korrigiert werden im Maschinendatum:
MD62620 \$MC_TRAFO6_AXIS_SEQ (Umordnung von Achsen)
5. Als Kennung für die Handachsen muss das folgende Maschinendatum gesetzt werden (keine Hand):
MD62604 \$MC_TRAFO6_WRIST_AXES = 1 (Handachsenkennung)
6. Ob Achse 4 parallel/antiparallel zur letzten rotatorischen Grundachse verläuft wird eintragen in das Maschinendatum:
MD62606 \$MC_TRAFO6_A4PAR (Achse 4 parallel/antiparallel zu letzter Grundachse)
7. Eintragung der Achstypen für die Transformation in Maschinendatum:
MD62601 \$MC_TRAFO6_AXES_TYPE (Achstyp für Transformation)
8. Vergleich der Drehrichtungen der Achsen mit den durch das Transformationspaket Handling vorgegebenen Drehrichtungen und Korrektur im Maschinendatum:
MD62618 \$MC_TRAFO6_AXES_DIR (Anpassung der physikalischen und mathematischen Drehrichtung)
9. Eintragen der mechanischen Nullpunktverschiebung im Maschinendatum:
MD62617 \$MC_TRAFO6_MAMES (Verschiebung mathematischer zu mechanischer Nullpunkt)
10. Eintragen der Grundachsängen in das Maschinendatum:
MD62607 \$MC_TRAFO6_MAIN_LENGTH_AB (Grundachsängen A und B)

11. Bestimmung des Frames T_IRO_RO und Eintragung der Verschiebung in das Maschinendatum:
MD62612 \$MC_TRAFO6_TIRORO_POS (Frame zwischen Fußpunkt und internem System (Positionsanteil))
Eintragung der Verdrehung in das Maschinendatum:
MD62613 \$MC_TRAFO6_TIRORO_RPY (Frame zwischen Fußpunkt und internem System (Rotationsanteil))
12. Bestimmung des Frames T_X3_P3 zur Anbringung der Hand. Hierzu ist das p3_q3_r3-Koordinatensystem als Ausgangssystem zu sehen. Die Verschiebung wird eingetragen in das Maschinendatum:
MD62608 \$MC_TRAFO6_TX3P3_POS (Anbringung der Hand (Positionsanteil))
Die Verdrehung wird eingetragen in das Maschinendatum:
MD62609 \$MC_TRAFO6_TX3P3_RPY (Anbringung der Hand (Rotationsanteil))
eingetragen.
13. Bestimmung des Flanschkoordinatensystems. Hierzu ist das Handpunkt-Koordinatensystem als Ausgangssystem zu sehen. Die Verschiebung wird eingetragen in das Maschinendatum:
MD62610 \$MC_TRAFO6_TFLWP_POS (Frame zwischen Handpunkt und Flansch (Positionsanteil))
Die Verdrehung wird eingetragen in das Maschinendatum:
MD62611 \$MC_TRAFO6_TFLWP_RPY (Frame zwischen Handpunkt und Flansch (Rotationsanteil))

SCARA-Kinematiken

4-Achser CC-Kinematik



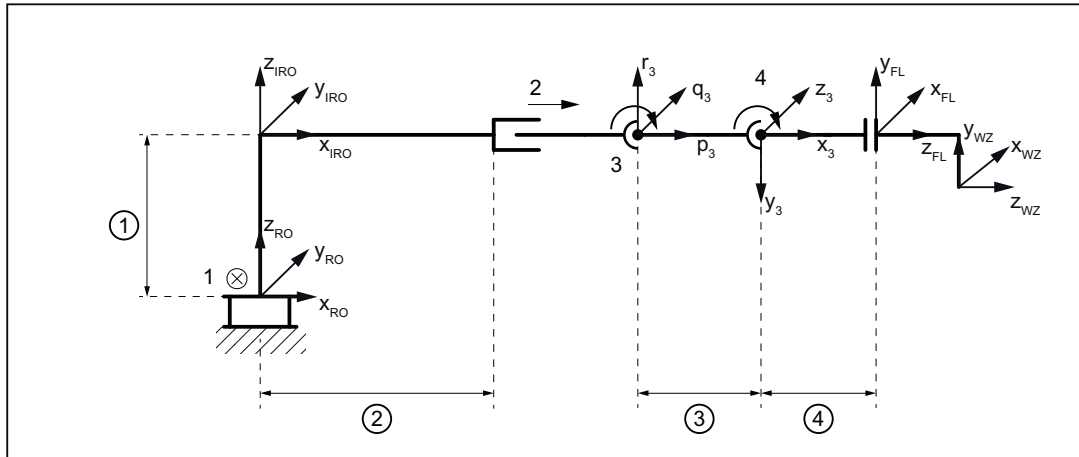
- ① MD62612 \$MC_TRAFO6_TIRORO_POS[2]
- ② MD62607 \$MC_TRAFO6_MAIN_LENGTH_AB[1]
- ③ MD62608 \$MC_TRAFO6_TX3P3_POS[0]
- ④ MD62608 \$MC_TRAFO6_TX3P3_POS[1]
- ⑤ MD62610 \$MC_TRAFO6_TFLWP_POS[0]

Bild 20-18 4-Achser CC-Kinematik

Tabelle 20-10 Projektierungsdaten 4-Achser CC-Kinematik

Maschinendatum	Wert
MD62600 \$MC_TRAFO6_KINCLASS	1
MD62605 \$MC_TRAFO6_NUM_AXES	4
MD62603 \$MC_TRAFO6_MAIN_AXES	2
MD62604 \$MC_TRAFO6_WRIST_AXES	1
MD62606 \$MC_TRAFO6_A4PAR	1
MD62601 \$MC_TRAFO6_AXES_TYPE	[3, 1, 3, 3, ...]
MD62620 \$MC_TRAFO6_AXIS_SEQ	[2, 1, 3, 4, 5, 6]
MD62618 \$MC_TRAFO6_AXES_DIR)	[1, 1, 1, 1, 1, 1]
MD62617 \$MC_TRAFO6_MAMES	[0.0, 0.0, 0.0, 0.0, 0.0, 0.0]
MD62607 \$MC_TRAFO6_MAIN_LENGTH_AB	[0.0, 300.0]
MD62612 \$MC_TRAFO6_TIRORO_POS	[0.0, 0.0, 500.0]
MD62613 \$MC_TRAFO6_TIRORO_RPY	[0.0, 0.0, 90.0]
MD62608 \$MC_TRAFO6_TX3P3_POS	[300.0, 0.0, -200.0]
MD62609 \$MC_TRAFO6_TX3P3_RPY	[-90.0, 90.0, 0.0]
MD62610 \$MC_TRAFO6_TFLWP_POS	[0.0, 0.0, 200.0]
MD62611 \$MC_TRAFO6_TFLWP_RPY	[0.0, -90.0, 0.0]

4-Achser SC-Kinematik



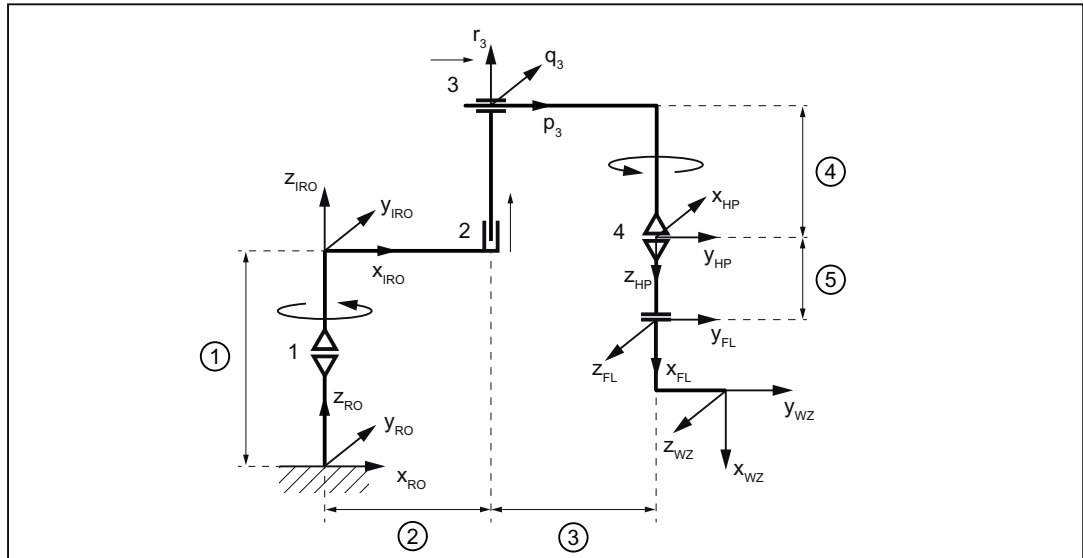
- ① MD62612 \$MC_TRAFO6_TIRORO_POS[2]
- ② MD62607 \$MC_TRAFO6_MAIN_LENGTH_AB[0]
- ③ MD62608 \$MC_TRAFO6_TX3P3_POS[0]
- ④ MD62610 \$MC_TRAFO6_TFLWP_POS[0]

Bild 20-19 4-Achser SC-Kinematik

Tabelle 20-11 Projektierungsdaten 4-Achser SC-Kinematik

Maschinendatum	Wert
MD62600 \$MC_TRAFO6_KINCLASS	1
MD62605 \$MC_TRAFO6_NUM_AXES	4
MD62603 \$MC_TRAFO6_MAIN_AXES	4
MD62604 \$MC_TRAFO6_WRIST_AXES	1
MD62606 \$MC_TRAFO6_A4PAR	1
MD62601 \$MC_TRAFO6_AXES_TYPE	[1, 1, 3, 3, ...]
MD62620 \$MC_TRAFO6_AXIS_SEQ	[1, 2, 3, 4, 5, 6]
MD62618 \$MC_TRAFO6_AXES_DIR	[1, 1, 1, 1, 1, 1]
MD62617 \$MC_TRAFO6_MAMES	[0.0, 0.0, 0.0, 0.0, 0.0, 0.0]
MD62607 \$MC_TRAFO6_MAIN_LENGTH_AB	[300.0, 0.0]
MD62612 \$MC_TRAFO6_TIRORO_POS	[0.0, 0.0, 300.0]
MD62613 \$MC_TRAFO6_TIRORO_RPY	[0.0, 0.0, 0.0]
MD62608 \$MC_TRAFO6_TX3P3_POS	[200.0, 0.0, 0.0]
MD62609 \$MC_TRAFO6_TX3P3_RPY	[0.0, 0.0, -90.0]
MD62610 \$MC_TRAFO6_TFLWP_POS	[200.0, 0.0, 0.0]
MD62611 \$MC_TRAFO6_TFLWP_RPY	[0.0, -90.0, 180.0]

4-Achser CS-Kinematik



- ① MD62612 \$MC_TRAFO6_TIRORO_POS[2]
- ② MD62607 \$MC_TRAFO6_MAIN_LENGTH_AB[0]
- ③ MD62608 \$MC_TRAFO6_TX3P3_POS[0]
- ④ MD62608 \$MC_TRAFO6_TX3P3_POS[2]
- ⑤ MD62610 \$MC_TRAFO6_TFLWP_POS[2]

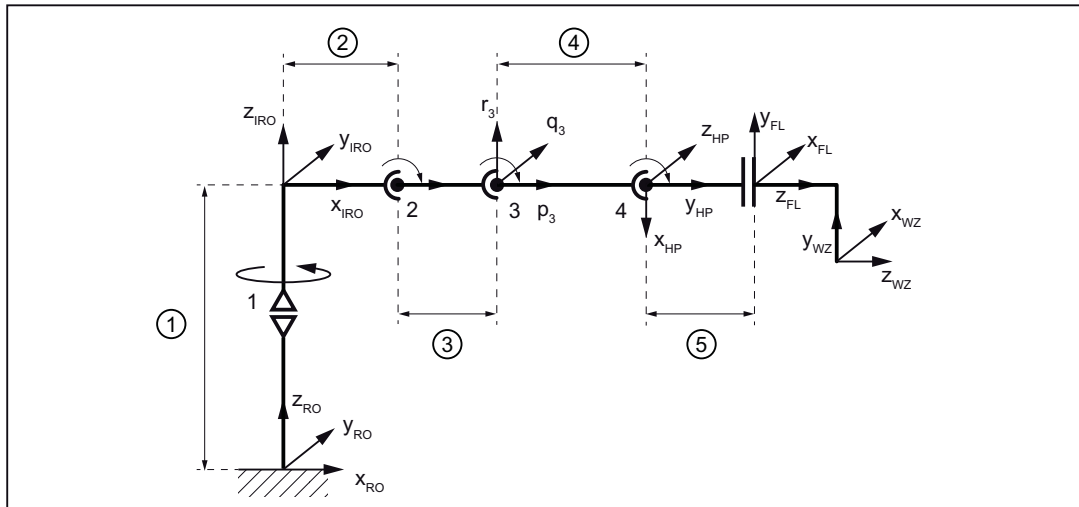
Bild 20-20 4-Achser CS-Kinematik

Tabelle 20-12 Projektierungsdaten 4-Achser CS-Kinematik

Maschinendatum	Wert
MD62600 \$MC_TRAFO6_KINCLASS	1
MD62605 \$MC_TRAFO6_NUM_AXES	4
MD62603 \$MC_TRAFO6_MAIN_AXES	6
MD62604 \$MC_TRAFO6_WRIST_AXES	1
MD62606 \$MC_TRAFO6_A4PAR	1
MD62601 \$MC_TRAFO6_AXES_TYPE	[3, 1, 1, 3, ...]
MD62620 \$MC_TRAFO6_AXIS_SEQ	[1, 2, 3, 4, 5, 6]
MD62618 \$MC_TRAFO6_AXES_DIR	[1, 1, 1, 1, 1, 1]
MD62617 \$MC_TRAFO6_MAMES	[0.0, 0.0, 0.0, 0.0, 0.0, 0.0]
MD62607 \$MC_TRAFO6_MAIN_LENGTH_AB	[400.0, 0.0]
MD62612 \$MC_TRAFO6_TIRORO_POS	[0.0, 0.0, 400.0]
MD62613 \$MC_TRAFO6_TIRORO_RPY	[0.0, 0.0, 0.0]
MD62608 \$MC_TRAFO6_TX3P3_POS	[500.0, 0.0, -200.0]
MD62609 \$MC_TRAFO6_TX3P3_RPY	[90.0, 0.0, 180.0]
MD62610 \$MC_TRAFO6_TFLWP_POS	[0.0, 0.0, 200.0]
MD62611 \$MC_TRAFO6_TFLWP_RPY	[0.0, -90.0, 0.0]

Gelenkarm-Kinematiken

4-Achser NR-Kinematik



- ① MD62612 \$MC_TRAFO6_TIRORO_POS[2]
- ② MD62607 \$MC_TRAFO6_MAIN_LENGTH_AB[0]
- ③ MD62607 \$MC_TRAFO6_MAIN_LENGTH_AB[1]
- ④ MD62608 \$MC_TRAFO6_TX3P3_POS[0]
- ⑤ MD62610 \$MC_TRAFO6_TFLWP_POS[0]

Bild 20-21 4-Achser NR-Kinematik

Tabelle 20-13 Projektierungsdaten 4-Achser NR-Kinematik

Maschinendatum	Wert
MD62600 \$MC_TRAFO6_KINCLASS	1
MD62605 \$MC_TRAFO6_NUM_AXES	4
MD62603 \$MC_TRAFO6_MAIN_AXES	3
MD62604 \$MC_TRAFO6_WRIST_AXES	1
MD62606 \$MC_TRAFO6_A4PAR	1
MD62601 \$MC_TRAFO6_AXES_TYPE	[3, 3, 3, 3, ...]
MD62620 \$MC_TRAFO6_AXIS_SEQ	[1, 2, 3, 4, 5, 6]
MD62618 \$MC_TRAFO6_AXES_DIR	[1, 1, 1, 1, 1, 1]
MD62617 \$MC_TRAFO6_MAMES	[0.0, 0.0, 0.0, 0.0, 0.0, 0.0]
MD62607 \$MC_TRAFO6_MAIN_LENGTH_AB	[300.0, 300.0]
MD62612 \$MC_TRAFO6_TIRORO_POS	[0.0, 0.0, 500.0]
MD62613 \$MC_TRAFO6_TIRORO_RPY	[0.0, 0.0, 0.0]
MD62608 \$MC_TRAFO6_TX3P3_POS	[300.0, 0.0, 0.0]
MD62609 \$MC_TRAFO6_TX3P3_RPY	[0.0, 0.0, -90.0]
MD62610 \$MC_TRAFO6_TFLWP_POS	[200.0, 0.0, 0.0]
MD62611 \$MC_TRAFO6_TFLWP_RPY	[0.0, -90.0, 180.0]

Siehe auch

Gelenkdefinition (Seite 778)

20.5.3 5-Achs-Kinematiken

5-Achs Kinematiken besitzen normalerweise 3 translatorische Freiheitsgrade und 2 weitere für die Orientierung.

Einschränkungen

Für 5-Achs Kinematiken gelten folgende Einschränkungen:

1. Es gibt Einschränkungen für das Flanschkoordinatensystem dahingehend, dass die X-Flansch-Achse die 5. Achse schneiden muss, aber nicht quasi parallel zu ihr sein darf.
2. Bei 5-Achs Gelenkarm-Kinematiken unterliegt das Frame T_FL_WP der Bedingung:
 - MD62610 \$MC_TRAFO6_TFLWP_POS = [0.0, 0.0, Z] (Frame zwischen Handpunkt und Flansch (Positionsanteil))
 - MD62611 \$MC_TRAFO6_TFLWP_RPY = [A, 0.0, 0.0] (Frame zwischen Handpunkt und Flansch (Rotationsanteil))
3. Es gibt Einschränkungen für das Werkzeug bei 5-Achs Gelenkarm-Kinematiken:
 - 4. Achse parallel zur 3. Achse: 2-dimensionales Werkzeug ist möglich [X, 0.0, Z]
 - 4. Achse senkrecht zur 3. Achse: nur 1-dimensionales Werkzeug ist möglich [X, 0.0, 0.0]
4. Es gibt Einschränkungen für das Werkzeug bei 5-Achs Scara-Kinematiken:
 - 4. Achse senkrecht zur 3. Achse: 1-dimensionales Werkzeug ist möglich [X, 0.0, 0.0]
5. Zwei aufeinander folgende Grundachsen müssen parallel oder orthogonal sein.
6. Die 4. Achse darf nur parallel oder orthogonal an die letzte Grundachse montiert werden.

Projektierung

Folgendes Vorgehen ist zur Projektierung einer 5-Achs Kinematik notwendig:

1. Kinematikkategorie "Standard" eintragen in Maschinendatum:
MD62600 \$MC_TRAFO6_KINCLASS (Kinematikkategorie)
2. Anzahl der Achsen für die Transformation in Maschinendatum:
MD62605 \$MC_TRAFO6_NUM_AXES = 5 (Anzahl der transformierten Achsen)
3. Vergleich der Grundachsen mit dem im Transformationspaket Handling enthaltenen Grundachsen.
 - Eintragen der Grundachsenkennung in Maschinendatum:
MD62603 \$MC_TRAFO6_MAIN_AXES (Grundachsenkennung)
4. Wenn eine zur normalen Achsreihenfolge unterschiedliche Reihenfolge vorliegt, muss sie korrigiert werden im Maschinendatum:
MD62620 \$MC_TRAFO6_AXIS_SEQ (Umordnung von Achsen)

5. Bestimmung der Kennung für die Handachsen. Wenn sich Achse 4 und 5 schneiden liegt eine Zentralhand (ZEH) vor. In allen anderen Fällen muss die Kennung für Winkelschräghand (WSH) eingetragen werden in das Maschinendatum:
MD62604 \$MC_TRAFO6_WRIST_AXES (Handachsenkennung)
6. Ob Achse 4 parallel/antiparallel zur letzten rotatorischen Grundachse verläuft muss eingetragen werden in das Maschinendatum:
MD62606 \$MC_TRAFO6_A4PAR (Achse 4 parallel/antiparallel zu letzter Grundachse)
7. Eintragung der Achstypen für die Transformation in Maschinendatum:
MD62601 \$MC_TRAFO6_AXES_TYPE (Achstyp für Transformation)
8. Vergleich der Drehrichtungen der Achsen mit den durch das Transformationspaket Handling vorgegebenen Drehrichtungen und Korrektur in Maschinendatum:
MD62618 \$MC_TRAFO6_AXES_DIR (Anpassung der physikalischen und mathematischen Drehrichtung)
9. Eintragen der mechanischen Nullpunktverschiebung in das Maschinendatum:
MD62617 \$MC_TRAFO6_MAMES (Verschiebung mathematischer zu mechanischer Nullpunkt)
10. Eintragen der Grundachslängen in Maschinendatum:
MD62607 \$MC_TRAFO6_MAIN_LENGTH_AB (Grundachslängen A und B)
11. Bestimmung des Frames T_IRO_RO und Eintragung der Verschiebung in Maschinendatum:
MD62612 \$MC_TRAFO6_TIRORO_POS (Frame zwischen Fußpunkt und internem System (Positionsanteil))
Bestimmung des Frames T_IRO_RO und Eintragung der Verdrehung in Maschinendatum:
MD62613 \$MC_TRAFO6_TIRORO_RPY (Frame zwischen Fußpunkt und internem System (Rotationsanteil))
12. Bestimmung des Frames T_X3_P3 zur Anbringung der Hand. Die Verschiebung wird eingetragen in Maschinendatum:
MD62608 \$MC_TRAFO6_TX3P3_POS (Anbringung der Hand (Positionsanteil))
Die Verdrehung wird eingetragen in das Maschinendatum:
MD62609 \$MC_TRAFO6_TX3P3_RPY (Anbringung der Hand (Rotationsanteil))
13. Bestimmung der Parameter für die Handachsen. Hierbei sind nur die Parameter für Achse 4 einzutragen in die Maschinendaten:
MD62614 \$MC_TRAFO6_DHPAR4_5A[0] (Parameter A zur Projektierung der Hand)
MD62616 \$MC_TRAFO6_DHPAR4_5ALPHA[0] (Parameter ALPHA zur Projektierung der Hand)
Alle anderen Parameter sind 0.0 zu setzen.
14. Bestimmung des Flanschkoordinatensystems. Hierzu ist das Handpunktkoordinatensystem als Ausgangssystem zu sehen. Die Verschiebung wird eingetragen in das Maschinendatum:
MD62610 \$MC_TRAFO6_TFLWP_POS (Frame zwischen Handpunkt und Flansch (Positionsanteil))
Die Verdrehung wird eingetragen in das Maschinendatum:
MD62611 \$MC_TRAFO6_TFLWP_RPY (Frame zwischen Handpunkt und Flansch (Rotationsanteil))

SCARA-Kinematiken

5-Achser CC-Kinematik

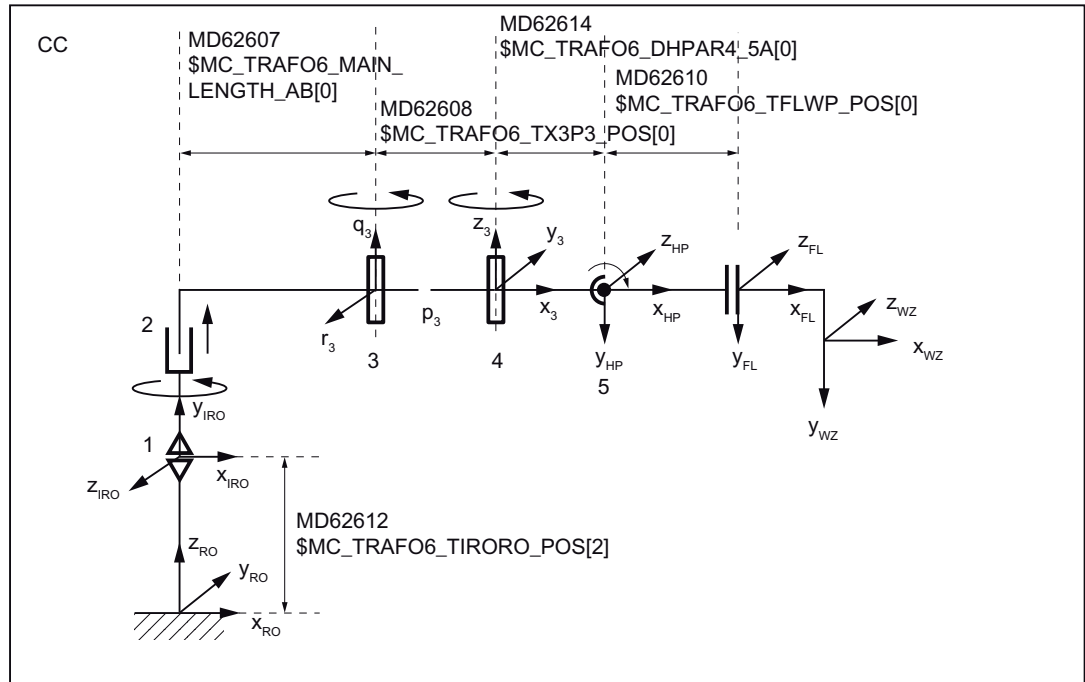


Bild 20-22 5-Achser CC-Kinematik

Tabelle 20-14 Projektierungsdaten 5-Achser CC-Kinematik

Maschinendatum	Wert
MD62600 \$MC_TRAFO6_KINCLASS	1
MD62605 \$MC_TRAFO6_NUM_AXES	5
MD62603 \$MC_TRAFO6_MAIN_AXES	2
MD62604 \$MC_TRAFO6_WRIST_AXES	5
MD62606 \$MC_TRAFO6_A4PAR	1
MD62601 \$MC_TRAFO6_AXES_TYPE	[3, 1, 3, 3, 3, ...]
MD62620 \$MC_TRAFO6_AXIS_SEQ	[2, 1, 3, 4, 5, 6]
MD62618 \$MC_TRAFO6_AXES_DIR	[1, 1, 1, 1, 1, 1]
MD62617 \$MC_TRAFO6_MAMES	[0.0, 0.0, 0.0, 0.0, 0.0, 0.0]
MD62607 \$MC_TRAFO6_MAIN_LENGTH_AB	[0.0, 500.0]
MD62612 \$MC_TRAFO6_TIRORO_POS	[0.0, 0.0, 500.0]
MD62613 \$MC_TRAFO6_TIRORO_RPY	[0.0, 0.0, 90.0]
MD62608 \$MC_TRAFO6_TX3P3_POS	[300.0, 0.0, -200.0]
MD62609 \$MC_TRAFO6_TX3P3_RPY	[0.0, 0.0, -90.0]
MD62610 \$MC_TRAFO6_TFLWP_POS	[200.0, 0.0, 0.0]
MD62611 \$MC_TRAFO6_TFLWP_RPY	[0.0, 0.0, 0.0]
MD62614 \$MC_TRAFO6_DHPAR4_5A	[200.0, 0.0]

Maschinendatum	Wert
MD62615 \$MC_TRAFO6_DHPAR4_5D	[0.0, 0.0]
MD62616 \$MC_TRAFO6_DHPAR4_5ALPHA	[-90.0, 0.0]

5-Achser NR-Kinematik

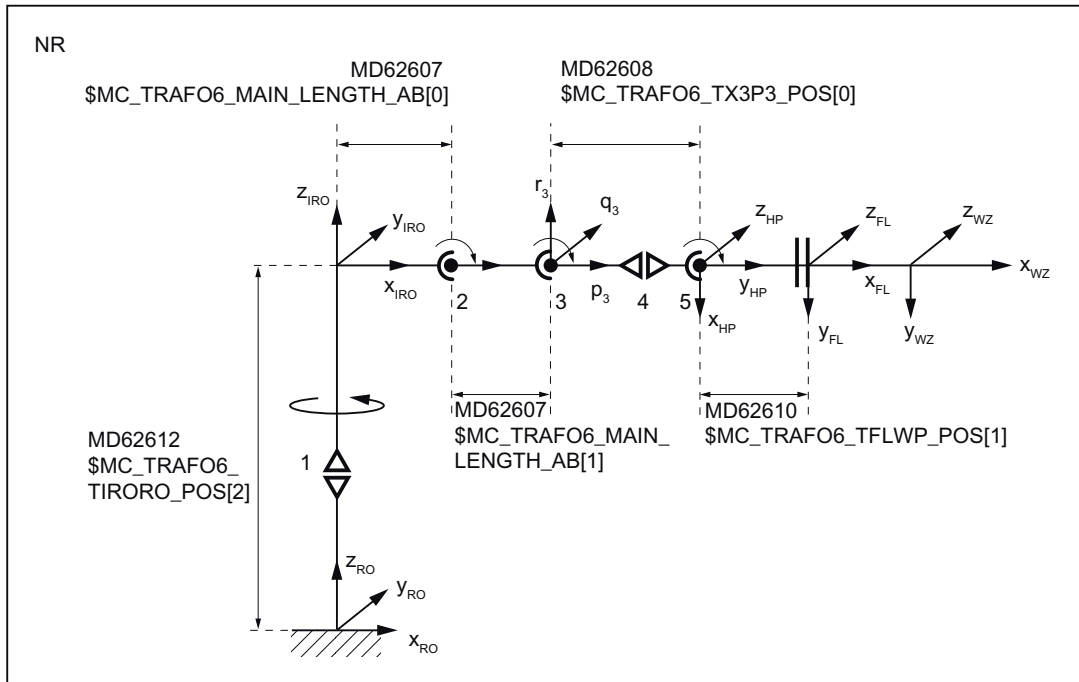


Bild 20-23 5-Achser NR Kinematik

Tabelle 20-15 Projektierungsdaten 5-Achser NR-Kinematik

Maschinendatum	Wert
MD62600 \$MC_TRAFO6_KINCLASS	1
MD62605 \$MC_TRAFO6_NUM_AXES	5
MD62603 \$MC_TRAFO6_MAIN_AXES	3
MD62604 \$MC_TRAFO6_WRIST_AXES	2
MD62606 \$MC_TRAFO6_A4PAR	0
MD62601 \$MC_TRAFO6_AXES_TYPE	[3, 3, 3, 3, 3, ...]
MD62620 \$MC_TRAFO6_AXIS_SEQ	[1, 2, 3, 4, 5, 6]
MD62618 \$MC_TRAFO6_AXES_DIR	[1, 1, 1, 1, 1, 1]
MD62617 \$MC_TRAFO6_MAMES	[0.0, 0.0, 0.0, 0.0, 0.0, 0.0]
MD62607 \$MC_TRAFO6_MAIN_LENGTH_AB	[30.0, 300.0]
MD62612 \$MC_TRAFO6_TIRORO_POS	[0.0, 0.0, 500.0]
MD62613 \$MC_TRAFO6_TIRORO_RPY	[0.0, 0.0, 0.0]
MD62608 \$MC_TRAFO6_TX3P3_POS	[500.0, 0.0, 0.0]
MD62609 \$MC_TRAFO6_TX3P3_RPY	[0.0, 90.0, 0.0]
MD62610 \$MC_TRAFO6_TFLWP_POS	[0.0, -300.0, 0.0]

Maschinendatum	Wert
MD62611 \$MC_TRAFO6_TFLWP_RPY	[-90.0, 0.0, 0.0]
MD62614 \$MC_TRAFO6_DHPAR4_5A	[0.0, 0.0]
MD62615 \$MC_TRAFO6_DHPAR4_5D	[0.0, 0.0]
MD62616 \$MC_TRAFO6_DHPAR4_5ALPHA	[-90.0, 0.0]

Siehe auch

Gelenkdefinition (Seite 778)

20.5.4 6-Achs-Kinematiken

6-Achs Kinematiken besitzen normalerweise 3 translatorische Freiheitsgrade und drei weitere für die Orientierung, bei dem zur Werkzeugrichtung beliebig im Raum, auch das Werkzeug um eine eigene Achse zur Bearbeitungsfläche gedreht oder mit einem Kippwinkel geneigt zugestellt werden kann.

Dieser Kinematiktyp wird von der Software nur für ganz spezielle Sonderlösungen als Zusatzoption zur eigentlichen Option dieses Transformationspaket Handling unterstützt. Für die 6-Achs Kinematik ist noch kein allgemein umfassender und universell einsetzbarer Softwarestand verfügbar.

20.5.5 Sonderkinematiken

MD62602 \$MC_TRAFO6_SPECIAL_KIN (Sonderkinematik-Typ)

Sonderkinematiken sind Kinematiken, die nicht direkt im Baukastensystem des Transformationspaket Handling enthalten sind. Sie sind häufig dadurch gekennzeichnet, dass entweder ein Freiheitsgrad fehlt, oder dass mechanische Kopplungen zwischen den Achsen oder auf das Werkzeug vorhanden sind. Für diese Kinematiken ist das folgende Maschinendatum zu setzen:

MD62600 \$MC_TRAFO6_KINCLASS = 2 (Kinematikklasse)

Um welche Sonderkinematik es sich handelt wird spezifiziert über das Maschinendatum:

MD62602 \$MC_TRAFO6_SPECIAL_KIN (Sonderkinematik-Typ)

2-Achser SC-Sonderkinematik

Diese Sonderkinematik ist dadurch gekennzeichnet, dass das Werkzeug über ein mechanisches Gestänge immer in derselben Orientierung gehalten wird. Sie besitzt 2 kartesische Freiheitsgrade. Diese Kinematik besitzt die Kennung Maschinendatum:

MD62602 \$MC_TARFO6_SPECIAL_KIN = 3 (Sonderkinematik-Typ)

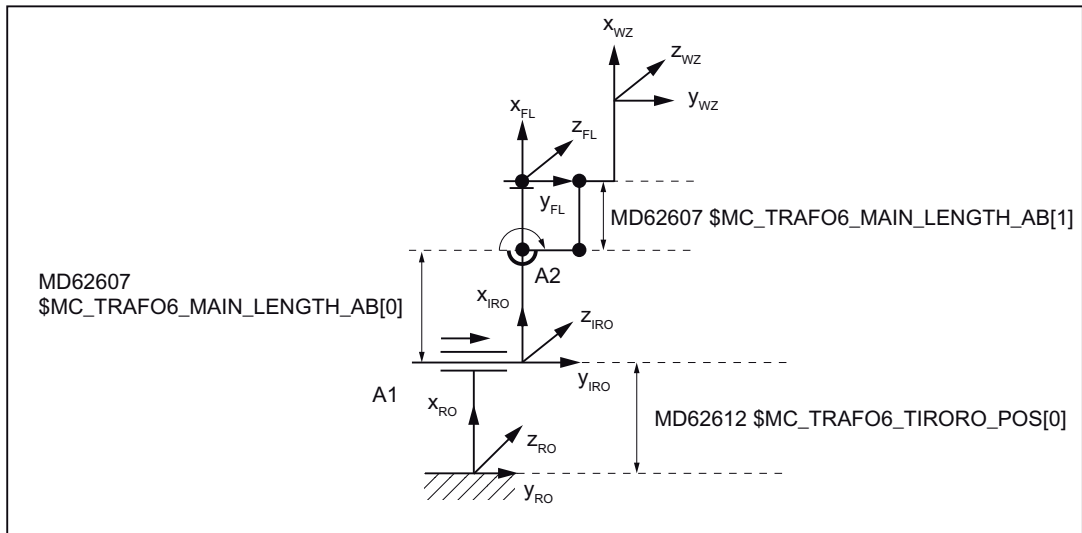


Bild 20-24 2-Achser SC-Sonderkinematik

Tabelle 20-16 Projektierungsdaten Sonderkinematik SC-2-Achser

Maschinendatum	Wert
MD62600 \$MC_TRAFO6_KINCLASS	2
MD62602 \$MC_TRAFO6_SPECIAL_KIN	3
MD62605 \$MC_TRAFO6_NUM_AXES	2
MD62603 \$MC_TRAFO6_MAIN_AXES	2
MD62604 \$MC_TRAFO6_WRIST_AXES	1
MD62601 \$MC_TRAFO6_AXES_TYPE	[1, 3, 3, ...]
MD62620 \$MC_TRAFO6_AXIS_SEQ	[1, 2, 3, 4, 5, 6]
MD62618 \$MC_TRAFO6_AXES_DIR	[1, 1, 1, 1, 1, 1]
MD62617 \$MC_TRAFO6_MAMES	[0.0, 0.0, 0.0, 0.0, 0.0, 0.0]
MD62607 \$MC_TRAFO6_MAIN_LENGTH_AB	[400.0, 500.0]
MD62612 \$MC_TRAFO6_TIRORO_POS	[0.0, 0.0, 300.0]
MD62613 \$MC_TRAFO6_TIRORO_RPY	[0.0, 0.0, 0.0]
MD62608 \$MC_TRAFO6_TX3P3_POS	[0.0, 0.0, 0.0]
MD62609 \$MC_TRAFO6_TX3P3_RPY	[0.0, 0.0, 0.0]
MD62610 \$MC_TRAFO6_TFLWP_POS	[0.0, 0.0, 0.0]
MD62611 \$MC_TRAFO6_TFLWP_RPY	[0.0, 0.0, 0.0]

3-Achser SC- Sonderkinematik

Diese Sonderkinematik besitzt 2 kartesische Freiheitsgrade und einen Freiheitsgrad für die Orientierung. Sie besitzt die Kennung:

MD62602 \$MC_TRAFO6_SPECIAL_KIN = 4 (Sonderkinematik-Typ)

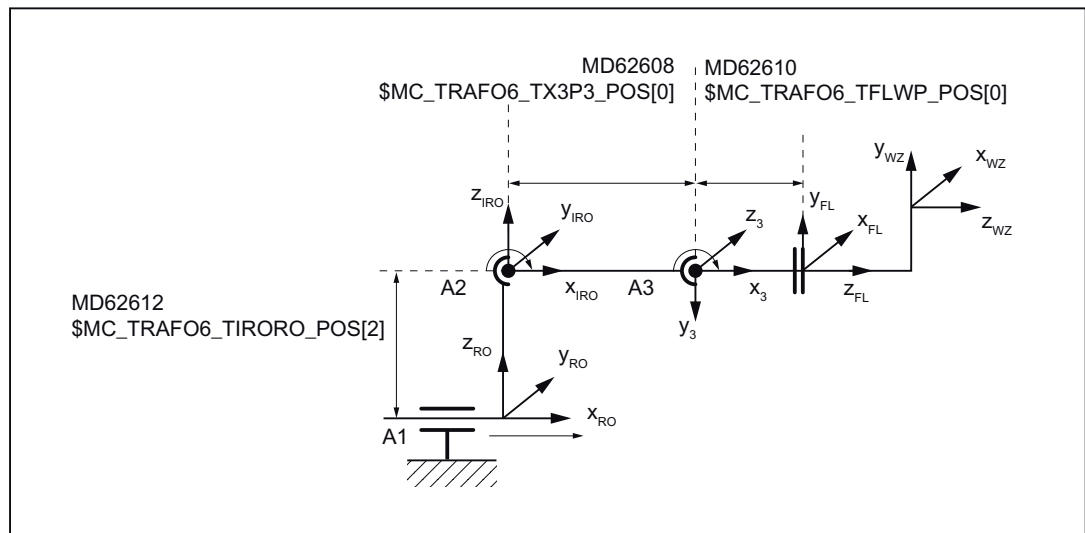


Bild 20-25 3-Achser SC-Sonderkinematik

Tabelle 20-17 Projektierungsdaten Sonderkinematik SC-3-Achser

Maschinendatum	Wert
MD62600 \$MC_TRAFO6_KINCLASS	2
MD62602 \$MC_TRAFO6_SPECIAL_KIN	4
MD62605 \$MC_TRAFO6_NUM_AXES	3
MD62603 \$MC_TRAFO6_MAIN_AXES	2
MD62604 \$MC_TRAFO6_WRIST_AXES	1
MD62601 \$MC_TRAFO6_AXES_TYPE	[1, 3, 3, ...]
MD62620 \$MC_TRAFO6_AXIS_SEQ	[1, 2, 3, 4, 5, 6]
MD62618 \$MC_TRAFO6_AXES_DIR	[1, 1, 1, 1, 1, 1]
MD62617 \$MC_TRAFO6_MAMES	[0.0, 0.0, 0.0, 0.0, 0.0, 0.0]
MD62607 \$MC_TRAFO6_MAIN_LENGTH_AB	[0.0, 0.0]
MD62612 \$MC_TRAFO6_TIRORO_POS	[0.0, 0.0, 400.0]
MD62613 \$MC_TRAFO6_TIRORO_RPY	[0.0, 0.0, 0.0]
MD62608 \$MC_TRAFO6_TX3P3_POS	[400.0, 0.0, 0.0]
MD62609 \$MC_TRAFO6_TX3P3_RPY	[0.0, 0.0, -90.0]
MD62610 \$MC_TRAFO6_TFLWP_POS	[200.0, 0.0, 0.0]
MD62611 \$MC_TRAFO6_TFLWP_RPY	[0.0, -90.0, 180.0]

4-Achser SC-Sonderkinematik

Diese Sonderkinematik ist dadurch gekennzeichnet, dass eine mechanische Kopplung zwischen Achse 1 und Achse 2 vorhanden ist. Hierbei wird die Achse 2 beim Schwenken der Achse 1 immer in konstantem Winkel gehalten. Bei dieser Kinematik werden zusätzlich die Achsen 3 und 4 unabhängig von der Stellung der Achsen 1 und 2 immer senkrecht gehalten. Sie besitzt die Kennung:

MD62602 \$MC_TRAFO6_SPECIAL_KIN = 7 (Sonderkinematik-Typ)

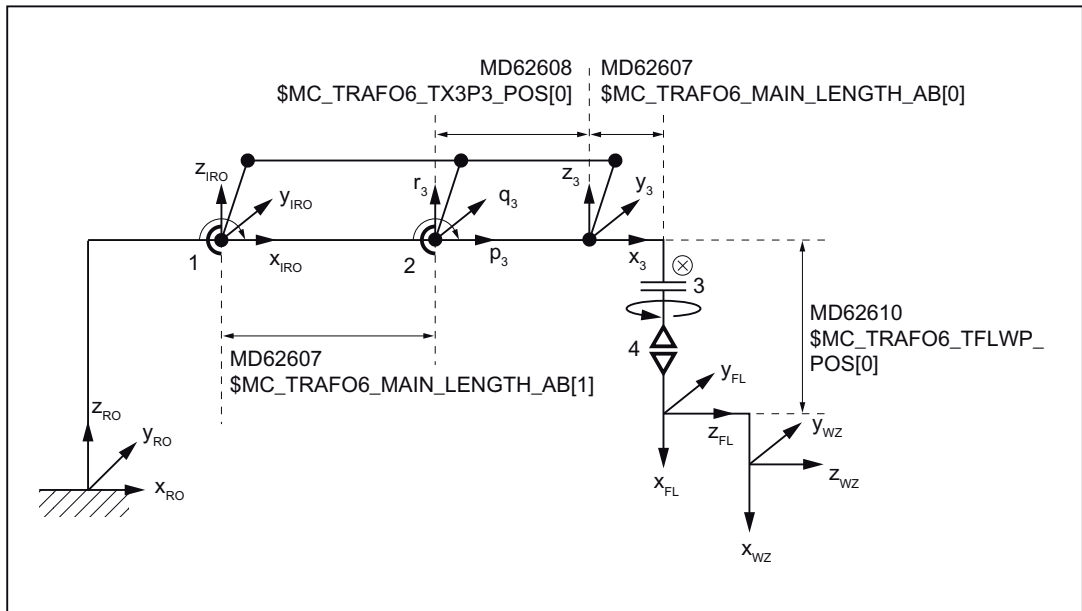


Bild 20-26 4-Achser SC-Sonderkinematik

Tabelle 20-18 Projektierungsdaten Sonderkinematik SC-4-Achser

Maschinendatum	Wert
MD62600 \$MC_TRAFO6_KINCLASS	2
MD62602 \$MC_TRAFO6_SPECIAL_KIN	7
MD62605 \$MC_TRAFO6_NUM_AXES	4
MD62603 \$MC_TRAFO6_MAIN_AXES	2
MD62604 \$MC_TRAFO6_WRIST_AXES	1
MD62601 \$MC_TRAFO6_AXES_TYPE	[3, 3, 1, 3, ...]
MD62620 \$MC_TRAFO6_AXIS_SEQ	[1, 2, 3, 4, 5, 6]
MD62618 \$MC_TRAFO6_AXES_DIR	[1, 1, 1, 1, 1, 1]
MD62617 \$MC_TRAFO6_MAMES	[0.0, 0.0, 0.0, 0.0, 0.0, 0.0]
MD62607 \$MC_TRAFO6_MAIN_LENGTH_AB	[100.0, 400.0]
MD62612 \$MC_TRAFO6_TIRORO_POS	[100.0, 0.0, 1000.0]
MD62613 \$MC_TRAFO6_TIRORO_RPY	[0.0, 0.0, 0.0]
MD62608 \$MC_TRAFO6_TX3P3_POS	[300.0, 0.0, 0.0]
MD62609 \$MC_TRAFO6_TX3P3_RPY	[0.0, 0.0, 0.0]
MD62610 \$MC_TRAFO6_TFLWP_POS	[0.0, 0.0, -600.0]
MD62611 \$MC_TRAFO6_TFLWP_RPY	[0.0, 90.0, 0.0]

2-Achser NR-Sonderkinematik

Diese Sonderkinematik ist dadurch gekennzeichnet, dass eine mechanische Kopplung zwischen Achse 1 und Achse 2 vorhanden ist. Eine weitere Besonderheit stellt das Werkzeug dar. Es behält seine Orientierung im Raum unabhängig von der Stellung der anderen Achsen bei.

Sie besitzt die Kennung:

MD62602 \$MC_TRAFO6_SPECIAL_KIN = 5 (Sonderkinematik-Typ)

Ohne mechanische Kopplung zwischen Achse 1 und 2 besitzt sie folgende Kennung:

MD62602 \$MC_TRAFO6_SPECIAL_KIN = 8 (Sonderkinematik-Typ)

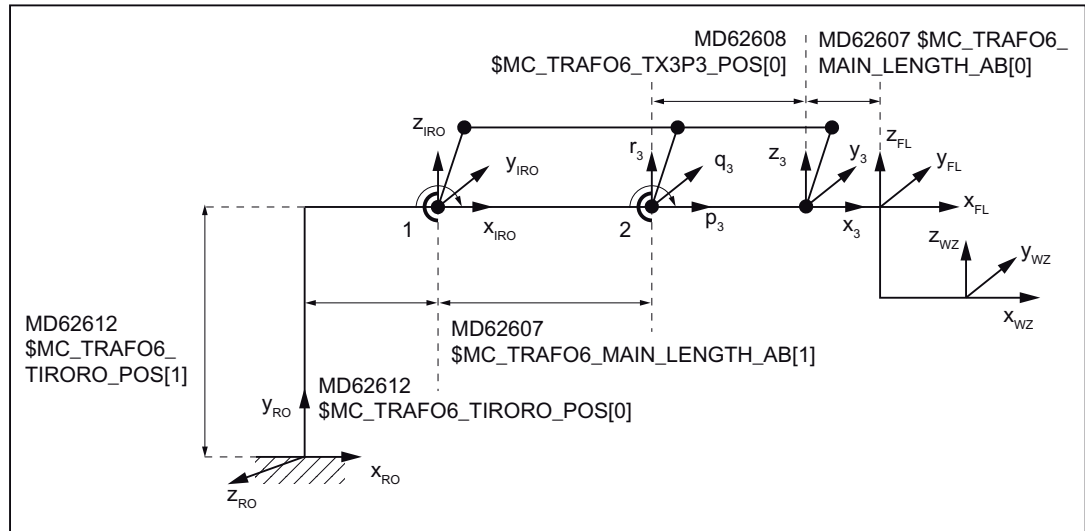
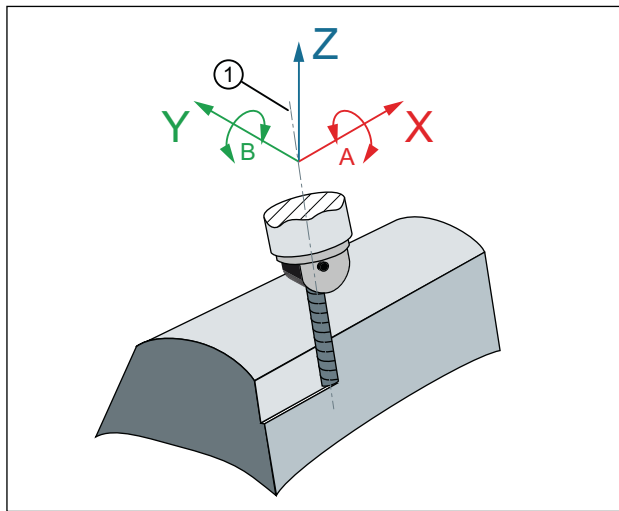


Bild 20-27 2-Achser NR-Sonderkinematik

Tabelle 20-19 Projektierungsdaten Sonderkinematik NR-2-Achser

Maschinendatum	Wert
MD62600 \$MC_TRAFO6_KINCLASS	2
MD62602 \$MC_TRAFO6_SPECIAL_KIN	5 (8)
MD62605 \$MC_TRAFO6_NUM_AXES	2
MD62603 \$MC_TRAFO6_MAIN_AXES	3
MD62604 \$MC_TRAFO6_WRIST_AXES	1
MD62601 \$MC_TRAFO6_AXES_TYPE	[3, 3, ...]
MD62620 \$MC_TRAFO6_AXIS_SEQ	[1, 2, 3, 4, 5, 6]
MD62618 \$MC_TRAFO6_AXES_DIR	[1, 1, 1, 1, 1, 1]
MD62617 \$MC_TRAFO6_MAMES	[0.0, 0.0, 0.0, 0.0, 0.0, 0.0]
MD62607 \$MC_TRAFO6_MAIN_LENGTH_AB	[100.0, 400.0]
MD62612 \$MC_TRAFO6_TIRORO_POS	[100.0, 500.0, 0.0]
MD62613 \$MC_TRAFO6_TIRORO_RPY	[0.0, 0.0, -90.0]
MD62608 \$MC_TRAFO6_TX3P3_POS	[400.0, 0.0, 0.0]
MD62609 \$MC_TRAFO6_TX3P3_RPY	[0.0, 0.0, 0.0]
MD62610 \$MC_TRAFO6_TFLWP_POS	[0.0, 0.0, 0.0]
MD62611 \$MC_TRAFO6_TFLWP_RPY	[0.0, 0.0, 0.0]

20.6 Werkzeugorientierung



① Werkzeugachse

Bild 20-28 Umfangfräsen mit 5-Achs-Transformation

Maschinendaten

Bezeichner der Euler-Winkel

Die Bezeichner, mit denen die Euler-Winkel im NC-Programm programmiert werden, ist einstellbar über::

MD10620 \$MN_EULER_ANGLE_NAME_TAB (Name der Eulerwinkel)

Standardbezeichner: "A2", "B2", "C2"

Bezeichner der Komponenten des Richtungsvektors

Die Bezeichner, mit denen die Komponenten des Richtungsvektors im NC-Programm programmiert werden, ist einstellbar über::

MD10640 \$MN_DIR_VECTOR_NAME_TAB (Name der Normalvektoren)

Die Werkzeugorientierung kann in einem beliebigen Satz stehen. Insbesondere kann sie auch alleine stehen, was zu einer Orientierungsänderung um die in Bezug auf das Werkstück feststehende Werkzeugspitze führt.

Kanalspezifische Vorgabe: Euler- oder RPY-Winkel

Ob die Programmierung der Werkzeugorientierung im Kanal mit Euler- oder RPY-Winkel erfolgt ist einstellbar über:

MD21100 \$MC_ORIENTATION_IS_EULER (Winkeldefinition bei Orientierungsprogrammierung)

G-Befehl für die Orientierungsinterpolation

Die Interpolationsart für die Orientierung wird festgelegt mit dem Maschinendatum:

MD21104 \$MC_ORI_IPO_WITH_G_CODE = <Wert>

<Wert>	Bedeutung
FALSE	G-Befehle zur Orientierungsinterpolation: ORIWKS und ORIMKS
TRUE	G-Befehle der 51. G-Gruppe zur Orientierungsinterpolation: ORIAXES, ORIVECT, ORIPLANE, ...

Kanalspezifische Grundstellung für die Werkzeugorientierung

Der G-Befehl der 25. G-Gruppe: "Bezug Werkzeugorientierung", die nach Kanal- oder Programmende-Reset wirksam wird, ist einstellbar über:

MD20150 \$MC_GCODE_RESET_VALUES[24] (25. G-Gruppe: Bezug Werkzeugorientierung)

Programmierung

Funktion

Die Programmierung der Werkzeugorientierung kann erfolgen über:

- Direkter Programmierung der Rundachsen
- Programmierung der Euler- oder RPY-Winkel
- Programmierung des Richtungsvektors

Über die Befehle ORIWKS und ORIMKS wird ausgewählt, in welchem Koordinatensystem die Bewegung ausgeführt wird.

Syntax

ORIWKS
ORIMKS

Bedeutung

ORIWKS:	Werkzeugorientierung im Werkstückkoordinatensystem (WKS) Bei einer Orientierungsänderung mit raumfester Werkzeugspitze, bewegt sich das Werkzeug auf einem Großkreis in der vom Anfangs- und Endvektor aufgespannten Ebene. Die Überführung einer Werkzeugorientierung am Satzanfang in eine Orientierung am Satzende kann nur im WKS erfolgen.	
	Wirksamkeit:	modal
ORIMKS:	Werkzeugorientierung im Maschinenkoordinatensystem (MKS) Bei einer Orientierungsänderung mit raumfester Werkzeugspitze, wird zwischen den Rundachspositionen linear interpoliert.	
	Wirksamkeit:	modal

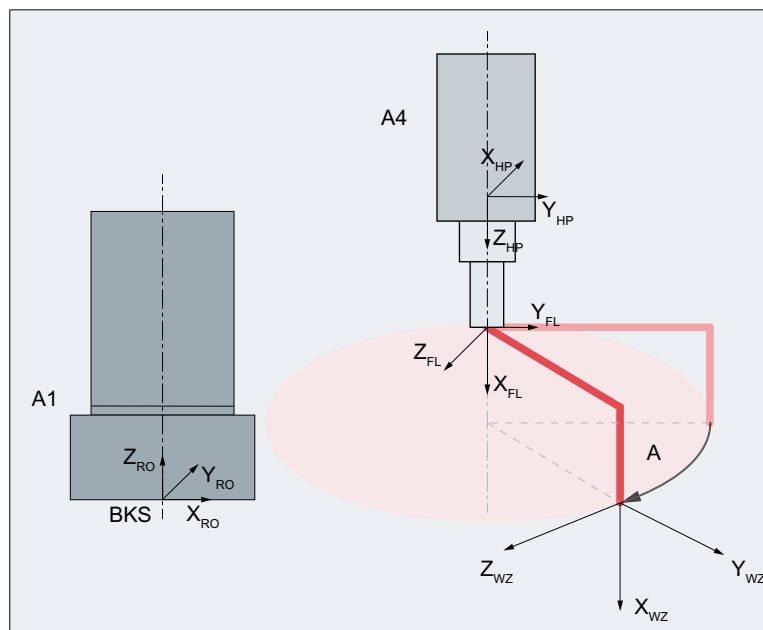
Hinweis

- Für ein Maschinen-unabhängiges Programm ist `ORIWKS` zu bevorzugen.
- Die Programmierung der Werkzeugorientierung mit Euler-Winkel, RPY-Winkel oder Richtungsvektor ist für Kinematiken mit weniger als 5 Achsen nicht möglich. Bei 4-Achskinematiken mit nur einer Rundachse existiert dann nur ein Freiheitsgrad für die Orientierung. Dieser Orientierungswinkel kann nur mit dem Orientierungswinkel "A" programmiert werden.
- Eine Orientierungsführung über `ORIMKS` ist mit dem Transformationspaket Handling nicht möglich. Mit aktiver Transformation werden nicht die Maschinenachswinkel programmiert und verfahren, sondern "Orientierungswinkel" (RPY-Winkel nach Robotik-Definition, siehe Kapitel "Positions- und Orientierungsbeschreibung mit Hilfe von Frames (Seite 777)").

20.6.1 Orientierungsprogrammierung bei 4-Achsern

Bei 4-Achs-Kinematiken existiert nur ein Freiheitsgrad für die Orientierung. Bei Programmierung der Orientierung über RPY-Winkel, Euler-Winkel oder Richtungsvektor, ist es in der Regel nicht immer möglich, die spezifizizierte Orientierung anzufahren. Wenn überhaupt, so ist diese Art der Orientierungsprogrammierung nur für bestimmte Kinematiken sinnvoll, bei denen eine Invarianz der Orientierungswinkel gegenüber den Grundachsen besteht. Dies ist zum Beispiel für die Scara- Kinematik der Fall.

Aus diesem Grund wird für Kinematiken mit 4 Achsen nur die Programmierung der Orientierung über den "Orientierungswinkel" A zugelassen. Dieser Winkel entspricht dem RPY-Winkel C der Robotik-Definition. Das heißt einer Drehung um die Achse Z_{RO} , wie aus dem nachfolgenden Bild "Orientierungswinkel beim 4-Achser" ersichtlich ist.



A Virtuellem Orientierungswinkel

Bild 20-29 Orientierungswinkel beim 4-Achser

Absolute Berechnung der Werkzeug-Orientierung bei 4-Achsern

Normalerweise ist der "virtuelle Orientierungswinkel" A bei 4-Achs-Kinematiken nur im Bereich $-180^\circ < A \leq +180^\circ$ definiert. Das bedeutet, dass der Winkel A beim Übergang von 180° nach 181° auf -179° springt. Um diesen Sprung zu vermeiden und eine kontinuierliche Führung des Winkels A zu erreichen, muss folgendes Maschinendatum gesetzt werden:

MD62643 \$MC_TRAFO6_SPECIAL_FEATURE_MASK, Bit 0 = 1

Darauf hin wird bei den folgenden 4-Achs Kinematiken der A-Winkel absolut geführt:

- CC-Kinematik
- SC-Kinematik
- CS-Kinematik

Voraussetzung dazu ist, dass die 4. Achse parallel zu letzten rotatorischen Grundachse angebracht ist (MD62606 \$MC_TRAFO6_A4PAR == 1).

Der A-Winkel wird hierbei aus der Summe der Rundachsen berechnet. Die Maschinendaten MD62617 \$MC_TRAFO6_MAMES und MD62618 \$MC_TRAFO6_AXES_DIR werden hierbei berücksichtigt.

Falls sich durch Verdrehungen über die MD62612 \$MC_TRAFO6_TIRORO_RPY und MD62611 \$MC_TRAFO6_TFLWP_RPY weitere Verdrehungen ergeben, müssen diese über das MD62642 \$MC_TRAFO6_C_ANGLE_OFFSET kompensiert werden. In das Maschinendatum MD62642 muss hierbei der Winkel, der als "virtueller Orientierungswinkel" A in der mathematischen Nullstellung der Achsen angezeigt wird, mit **invertiertem** Vorzeichen eingetragen werden.

20.6.2 Orientierungsprogrammierung bei 5-Achsern

Bei 5-Achs-Kinematiken wird bei der Programmierung über Orientierungsvektor davon ausgegangen, dass der Orientierungsvektor der x-Komponente des Werkzeugs entspricht.

Bei der Programmierung über Orientierungswinkel (RPY-Winkel nach Robotik-Definition) wird als Ausgangspunkt für die Drehungen die x-Komponente des Werkzeugs herangezogen.

Hierbei wird der Vektor in x-Werkzeugrichtung wie aus Bild "Orientierungswinkel beim 5-Achser" ersichtlich zuerst mit dem Winkel A um die Z-Achse und dann um den Winkel B um die gedrehte Y-Achse gedreht. Die Drehung um den Winkel C ist aufgrund der eingeschränkten Freiheitsgrade für die Orientierung beim 5-Achser nicht möglich.

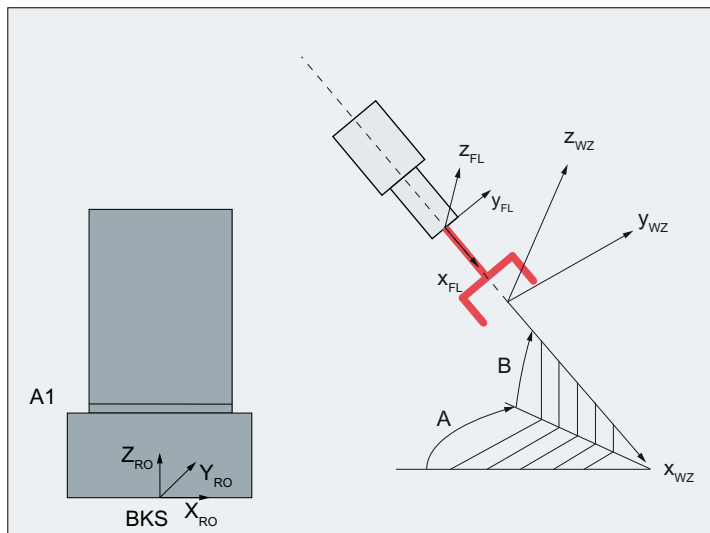


Bild 20-30 Orientierungswinkel beim 5-Achser

Anpassung des Flanschkoordinatensystems

Mit dem Maschinendatum wird das Flanschkoordinatensystem auf Anwenderseite so eingestellt, dass sich bei einem 5-Achser Z als Werkzeugrichtung einstellen lässt:

MD62636 \$MC_TRAFO6_TFL_EXT_RPY

Definition der Werkzeugrichtung

Mit dem Maschinendatum wird eingestellt, ob die Werkzeugrichtung nach Robotik-Konvention oder nach NC-Konvention eingerechnet werden soll:

MD62637 \$MC_TRAFO6_TOOL_DIR

Standard-Definition der Werkzeugrichtung

Mit dem Maschinendatum kann die bei Werkzeugmaschinen übliche Werkzeugverrechnung eingestellt werden:

MD62637 \$MC_TRAFO6_TOOL_DIR = 1

Dieses Maschinendatum hat auch Einfluss auf die Drehreihenfolge der virtuellen Orientierungsachsen. Hierbei wird der Vektor in z-Werkzeugrichtung zuerst mit dem Winkel A um die X-Achse und dann um den Winkel B um die gedrehte Y-Achse gedreht. Die Drehung um den Winkel C ist aufgrund der eingeschränkten Freiheitsgrade für die Orientierung beim 5-Achser nicht möglich.

Hinweis

Weitere Informationen finden sich in:

- Funktionshandbuch Sonderfunktionen, Kapitel "Orientierungsachsen"
 - Programmierhandbuch Arbeitsvorbereitung, Kapitel "Orientierungsachsen".
-

20.7 Singuläre Stellungen und ihre Behandlung

Die Berechnung der Maschinenachsen zu einer vorgegebenen Stellung, d. h. Position mit Orientierung, ist nicht immer eindeutig. Abhängig von der Kinematik der Maschine kann es Stellungen geben, die unendlich viele Lösungen besitzen. Diese Stellungen heißen "singulär".

Singuläre Stellungen

- Eine singuläre Stellung ist zum Beispiel dadurch gekennzeichnet, dass die fünfte Achse bei 0° liegt. Die singuläre Stellung ist hier nicht an eine bestimmte Orientierung gekoppelt. In dieser Stellung ist die vierte Achse nicht bestimmt, d. h. die vierte Achse hat keinen Einfluss auf die Position oder Orientierung.
- Bei Gelenkarm- und Scara-Kinematiken liegt auch eine singuläre Stellung vor, wenn die dritte Achse bei 0° oder bei 180° steht. Diese Stellungen nennt man Streck-/Beuge-Singularität.
- Eine weitere singuläre Stellung liegt bei Gelenkarm-Kinematiken vor, wenn der Handpunkt über der Drehachse von Achse 1 steht. Diese Stellung nennt man Überkopfsingularität.

Extreme Geschwindigkeitsüberhöhung

Wenn die Bahn in der Nähe eines Pols (Singularität) verläuft, dann kann es vorkommen, dass eine oder mehrere Achsen mit sehr hoher Geschwindigkeit fahren. Dann wird der Alarm 10910 "Irregulärer Geschwindigkeitsverlauf in einer Bahnachse" ausgelöst.

Verhalten am Pol

Das unerwünschte Verhalten schneller Ausgleichsbewegungen lässt sich durch eine Reduzierung der Geschwindigkeit in Polnähe verbessern. Ein Durchfahren des Pols mit eingeschalteter Transformation ist in der Regel nicht möglich.

20.8 Aufruf und Anwendung der Transformation

Einschalten

Die Transformation wird mit dem Befehl `TRAORI (1)` eingeschaltet.

Wenn der Befehl `TRAORI (1)` abgearbeitet wurde und damit die Transformation aktiviert ist, geht das Nahtstellensignal auf "1":

DB21, ... DBX33.6 (Transformation aktiv)

Sind die Maschinendaten für einen aufgerufenen Transformationsverbund nicht definiert, hält das NC-Programm an und die Steuerung gibt den Alarm 14100 "Orientierungstransformation nicht vorhanden" aus.

Weitere Informationen finden Sie unter:

Literatur:

Programmierhandbuch Arbeitsvorbereitung,
Kapitel: 5-Achsbearbeitung

Ausschalten

Mit `TRAFOOF` oder `TRAFOOF()` wird die gerade aktive Transformation ausgeschaltet.

Hinweis

Beim Ausschalten der Transformation "Transformationspaket Handling" wird kein Vorlaufstopp und keine Synchronisation des Vorlaufs mit dem Hauptlauf ausgeführt.

Reset/ Programmende

Das Verhalten der Steuerung bezüglich der Transformationen nach dem Hochlauf, Programmende oder `RESET` richtet sich nach Maschinendatum:

MD20110 `$MC_RESET_MODE_MASK` (Festlegung der Steuerungs-Grundstellung na)

Bit 7: Resetverhalten "aktive kinematische Transformation"

Bit 7 = 0 Damit wird die Grundstellung für die aktive Transformation nach Teileprogrammende oder `RESET` gemäß folgendem Maschinendatum festgelegt:

MD20140 `$MC_TRAFO_RESET_VALUE` (Transformationsdatensatz Hochlauf (Reset/TP-Ende))

Bedeutung:

0: Nach `RESET` ist keine Transformation aktiv.

1 bis 8: Die eingestellte Transformation ist aktiv gemäß Maschinendatum:

MD24100 `$MC_TRAFO_TYPE_1` (Definition der Transformation 1 im Kanal)

Bis Maschinendatum:

MD24460 `$MC_TRAFO_TYPE_8` (Transformation 8 im Kanal)

Bit 7 = 1 Die aktuelle Einstellung für die aktive Transformation bleibt über `RESET` oder Teileprogrammende erhalten.

20.9 Istwertanzeige

Maschinenkoordinatensystem MKS

Im Anzeigemodus MKS werden die Maschinenachsen in mm/Inch bzw. Grad angezeigt.

Werkstückkoordinatensystem WKS

Wenn die Transformation eingeschaltet ist, werden im Anzeigemodus WKS die Spitze des Werkzeugs (TCP-Tool-Center-Point) in mm/Inch und die Orientierung durch die RPY-Winkel A, B und C beschrieben. Die Werkzeugrichtung ergibt sich, indem ein Vektor in Z-Richtung zunächst mit A um die Z-Achse, dann mit B um die neue Y-Achse, und zuletzt mit C um die neue X-Achse gedreht wird.

Wenn die Transformation ausgeschaltet ist, werden die Achsen mit den Kanalachsnamen angezeigt, sonst werden die Geoachsnamen angezeigt.

20.10 Werkzeugprogrammierung

Bedeutung

Die Werkzeuglängen werden bezüglich dem Flanschkoordinatensystem angegeben. Es sind nur 3-dimensionale Werkzeugkorrekturen möglich. Je nachdem, um welche Kinematik es sich handelt, gibt es für 5- und 4-achsige Kinematiken zusätzliche Einschränkungen für das Werkzeug. Für eine Kinematik nach Bild "5-Achser NR-Kinematik" ist nur ein 1-dimensionales Werkzeug mit Länge in x-Richtung zugelassen.

Die Richtung des Werkzeugs ist von der Grundstellung der Maschine, die mit den G-Befehlen G17, G18 und G19 spezifiziert wird, abhängig. Die Werkzeuglängen beziehen sich auf die mit G17 angegebene Nullstellung. Diese sollte im Programm nicht geändert werden.

Beispiel

Als Beispiel sei ein 2-dimensionales Werkzeug gegeben, das an einem 5-Achser Scara (vgl. Bild "5-Achser CC-Kinematik") angebracht ist. Als Kennung für das Werkzeug sollte der Typ 100 (Schneidwerkzeug) angegeben werden. Die Werkzeuglängen bestimmen sich, wie aus Bild "Werkzeuglängen-Programmierung" ersichtlich. X-TOOL muss als Werkzeuglänge x, Y-TOOL als Werkzeuglänge y in den Werkzeugparametern eingegeben werden.

$\$TC_DP1[1,1] = 100 ;$	Typ Schneidwerkzeug
$\$TC_DP3[1,1] = 0.0 ;$	(z) Längenkorrekturvektor
$\$TC_DP4[1,1] = Y-TOOL ;$	(y) Längenkorrekturvektor
$\$TC_DP5[1,1] = X-TOOL ;$	(x) Längenkorrekturvektor

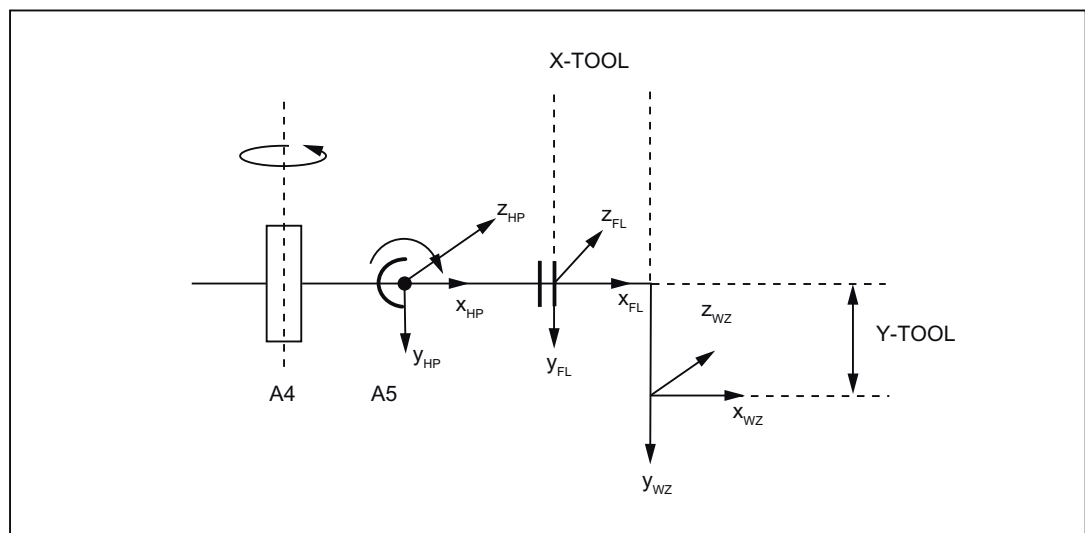


Bild 20-31 Werkzeuglängen-Programmierung

20.11 Kartesisches PTP–Fahren mit Transformationspaket Handling

Es ist möglich, mit dem Transformationspaket Handling die Funktion "Kartesisches PTP-Fahren" zu verwenden (siehe Kapitel "Kartesisches PTP-Fahren (Seite 42)"). Hierbei muss das folgende Maschinendatum gesetzt werden:

MD24100 \$MC_TRAFO_TYPE_1 = 4100 (Definition der Transformation 1 im Kanal)

20.12 Inbetriebnahme

20.12.1 Allgemeine Inbetriebnahme

Das Transformationspaket Handling ist ein Compile-Zyklus.

Die allgemeine Inbetriebnahme eines Compile-Zyklus ist beschrieben in Kapitel "TE01: Installation und Aktivierung ladbarer Compile-Zyklen (Seite 689)".

20.12.2 Funktionsspezifische Inbetriebnahme

Option

Die Funktion "Transformationspaket Handling " ist eine Option für die eine Lizenz erworben werden muss:

- Transformation Handling RCTRA, Bestell-Nr.: 6FC5800-0AM31-0YB0

Eine ausführliche Beschreibung zur Lizenzierung, wie z.B. die Zuordnung einer Lizenz zur Hardware, findet sich in:

Literatur

Inbetriebnahmehandbuch Inbetriebnahme CNC: NC, PLC, Antrieb

Kapitel: "Lizenzierung"

Anwenderspezifische Alarmtexte

Zum Anlegen von anwenderspezifischen Alarmtexten im Zusammenhang mit Comple-Zyklen siehe Kapitel "Anlegen von Alarmtexten (Seite 696)"

Transformation parametrieren

1. Tragen Sie den Transformationstyp ein in das Maschinendatum:
 - MD24100 \$MC_TRAFO_TYPE_1 = **4100** (Definition der Transformation 1 im Kanal)
2. Tragen Sie die Zuordnung der an der Transformation beteiligten Kanalachsen ein in das Maschinendatum (Achsnummern beginnend bei 1):
 - MD24110 \$MC_TRAFO_AXES_IN_1[0 ... 5] (Achszuordnung für Transformation)

3. Tragen Sie entsprechend der kartesischen Freiheitsgrade der Maschine die Geometrieachsen ein in das Maschinendatum:
 - MD24120 \$MC_TRAFO_GEOAX_ASSIGN_TAB_1[0 ... 2] (Zuordnung GEOAchse zu Kanalachsen für Transformation 1)
4. Tragen Sie die Kinematikkennung ein in das Maschinendatum:
 - MD62600 \$MC_TRAFO6_KINCLASS (Kinematikklasse)
5. Tragen Sie die Kennung für Sonderkinematiken, wenn eine solche Kinematik vorliegt, ein in Maschinendatum:
 - MD62602 \$MC_TRAFO6_SPECIAL_KIN (Sonderkinematik-Typ)
6. Tragen Sie die Achsanzahl ein in das Maschinendatum:
 - MD62605 \$MC_TRAFO6_NUM_AXES (Anzahl der transformierten Achsen)
7. Wenn die Verfahrrichtungen der beteiligten Achsen entgegengesetzt der Transformationsdefinition sind, ändern Sie die Voreinstellung im Maschinendatum:
 - MD62618 \$MC_TRAFO6_AXES_DIR[] (Anpassung der physikalischen und mathematischen Drehrichtung)
8. Tragen Sie die Daten, welche die Grundachsen beschreiben ein:
 - MD62603 \$MC_TRAFO6_MAIN_AXES (Grundachsenkennung)
 - MD62607 \$MC_TRAFO6_MAIN_LENGTH_AB (Grundachslängen A und B)
9. Tragen Sie eventuelle Achsvertauschungen ein in das Maschinendatum:
 - MD62620 \$MC_TRAFO6_AXIS_SEQ (Umordnung von Achsen)
10. Tragen Sie die Daten, welche die Hand beschreiben ein:
 - MD62604 \$MC_TRAFO6_WRIST_AXES (Handachsenkennung)
 - MD62614 \$MC_TRAFO6_DHPAR4_5A (Parameter A zur Projektierung der Hand)
 - MD62615 \$MC_TRAFO6_DHPAR4_5D (Parameter D zur Projektierung der Hand)
 - MD62616 \$MC_TRAFO6_DHPAR4_5ALPHA (Parameter ALPHA zur Projektierung der Hand)
 - MD62606 \$MC_TRAFO6_A4PAR (Achse 4 parallel/antiparallel zu letzter Grundachse)
11. Tragen Sie die Geometrieparameter ein:
 - Frame T_IRO_RO
 - Frame T_X3_P3
 - Frame T_FL_WP
12. Tragen Sie die Position in der Kalibrierposition ein in das Maschinendatum:
 - MD62617 \$MC_TRAFO6_MAMES (Verschiebung mathematischer zu mechanischer Nullpunkt)
13. Tragen Sie die kartesischen Geschwindigkeiten und Beschleunigungen ein.

20.13 Randbedingungen

Abstandsregelung

Das Transformationspaket Handling kann **nicht** zusammen mit der Technologiefunktion: "Abstandsregelung" betrieben werden.

Fahren auf Festanschlag

Das Transformationspaket Handling kann **nicht** zusammen mit der Funktion "Fahren auf Festanschlag" betrieben werden.

Mehrere Transformationen

Das Transformationspaket Handling kann nur einmal pro Kanal in allen Kanälen aktiviert werden.

Werkzeugprogrammierung

Für die Eingabe der Werkzeugparameter stehen nur Werkzeuglängen zur Verfügung. Es ist nicht möglich eine Orientierung für das Werkzeug zu programmieren.

Programmierung der Orientierung

Die Programmiermöglichkeiten der Orientierung sind abhängig von der Anzahl der an Maschine vorhandenen Achsen:

- Anzahl < 5:
 - Orientierungsachswinkel
- Anzahl = 5:
 - Orientierungsachswinkel
 - Orientierungsvektor

Singularitäten

Das direkte Durchfahren eines Pols ist mit eingeschalteter Transformation nicht möglich.

ACHTUNG

Verfahren in Polnähe

Beim Verfahren in Polnähe, kann es zu Überlastungen in den beteiligten Achsen kommen, da von der Steuerung keine automatische Reduzierung oder Begrenzung des Vorschubs, der Beschleunigung oder des Rucks vorgenommen wird. Es liegt daher in der alleinigen Verantwortung des Anwenders, den Vorschub der beteiligten Achsen bzw. den Abstand zum Pol zu begrenzen.

20.14 Datenlisten

20.14.1 Maschinendaten

20.14.1.1 Allgemeine Maschinendaten

Nummer	Bezeichner: \$MN_	Beschreibung
10620	EULER_ANGLE_NAME_TAB[n]	Name der Euler-Winkel
19410	TRAFO_TYPE_MASK, Bit 4	Optionsdatum für OEM-Transformation
60943	CC_ACTIVE_IN_CHAN_RCTR	Aktivierung der Handling-Transformation für die entsprechenden Kanäle
19610	TECHNO_EXTENSION_MASK	Optionsdatum für Handling-Transformation

20.14.1.2 Kanal-spezifische Maschinendaten

Nummer	Bezeichner: \$MC_	Beschreibung
21100	ORIENTATION_IS_EULER	Winkeldefinition bei der Orientierungsprogrammierung
21110	X_AXIS_IN_OLD_X_Z_PLANE	Koordinatensystem bei automatischer FRAME-Definition
24100	TRAFO_TYPE_1	Definition der Transformation
24110	TRAFO_AXES_IN_1	Achszuordnung für Transformation 1
24120	TRAFO_GEOAX_ASSIGN_TAB_1	Zuordnung der Geometrieachsen zu den Kanalachsen

20.14.1.3 Kanal-spezifische Maschinendaten für Compile-Zyklen

Nummer	Bezeichner: \$MC_	Beschreibung
62600	TRAFO6_KINCLASS	Kinematikkategorie
62601	TRAFO6_AXES_TYPE	Achstyp für Transformation
62602	TRAFO6_SPECIAL_KIN	Sonderkinematik-Typ
62603	TRAFO6_MAIN_AXES	Grundachsenkennung
62604	TRAFO6_WRIST_AXES	Handachsenkennung
62605	TRAFO6_NUM_AXES	Anzahl der transformierten Achsen
62606	TRAFO6_A4PAR	Achse 4 parallel/antiparallel zu letzter Grundachse
62607	TRAFO6_MAIN_LENGTH_AB	Grundachslängen A und B
62608	TRAFO6_TX3P3_POS	Anbringung der Hand (Positionsanteil)
62609	TRAFO6_TX3P3_RPY	Anbringung der Hand (Rotationsanteil)
62610	TRAFO6_TFLWP_POS	Frame zwischen Handpunkt und Flansch (Positionsanteil)
62611	TRAFO6_TFLWP_RPY	Frame zwischen Handpunkt und Flansch (Rotationsanteil)

Nummer	Bezeichner: \$MC_	Beschreibung
62612	TRAFO6_TIRORO_POS	Frame zwischen Fußpunkt und internem System (Positionsteil)
62613	TRAFO6_TIRORO_RPY	Frame zwischen Fußpunkt und internem System (Rotationsanteil)
62614	TRAFO6_DHPAR4_5A	Parameter A zur Projektierung der Hand
62615	TRAFO6_DHPAR4_5D	Parameter D zur Projektierung der Hand
62616	TRAFO6_DHPAR4_5ALPHA	Parameter ALPHA zur Projektierung der Hand
62617	TRAFO6_MAMES	Verschiebung mathematischer zu mechanischer Nullpunkt
62618	TRAFO6_AXES_DIR	Anpassung der physikalischen und mathematischen Drehrichtung
62619	TRAFO6_DIS_WRP	Mittlerer Abstand des Handpunkts zur Singularität (nicht verwendet)
62620	TRAFO6_AXIS_SEQ	Umordnung von Achsen
62621	TRAFO6_SPIN_ON	Dreiecks- oder Trapez-Spindeln vorhanden
62622	TRAFO6_SPIND_AXIS	Achse auf welche die Dreiecksspindel wirkt
62623	TRAFO6_SPINDLE_RAD_G	Radius G für Dreiecksspindel
62624	TRAFO6_SPINDLE_RAD_H	Radius H für Dreiecksspindel
62625	TRAFO6_SPINDLE_SIGN	Vorzeichen für Dreiecksspindel
62626	TRAFO6_SPINDLE_BETA	Winkelversatz für Dreiecksspindel
62627	TRAFO6_TRP_SPIND_AXIS	Über Trapezverbindung angetriebene Achsen
62628	TRAFO6_TRP_SPIND_LEN	Trapezlängen
62629	TRAFO6_VELCP	Kartesische Geschwindigkeiten (Ersatzgrenzwerte für Geschwindigkeitsregler)
62630	TRAFO6_ACCCP	Kartesische Beschleunigungen (Ersatzgrenzwerte für Geschwindigkeitsregler)
62631	TRAFO6_VELORI	Orientierungswinkel-Geschwindigkeiten (Ersatzgrenzwerte für Geschwindigkeitsregler)
62632	TRAFO6_ACCORI	Orientierungswinkel-Beschleunigungen (Ersatzgrenzwerte für Geschwindigkeitsregler)
62634	TRAFO6_DYN_LIM_REDUCE	Reduzierfaktor für Geschwindigkeitsregler beim JOG-Fahren
62635	TRAFO6_VEL_FILTER_TIME	Zeitkonstante für PT1-Filter bei aktivem Geschwindigkeitsregler
62636	TRAFO6_TFL_EXT_RPY	Anpassung des Flanschkoordinatensystems
62637	TRAFO6_TOOL_DIR	Definition der Werkzeugrichtung
62642	TRAFO6_C_ANGLE_OFFSET	Offsetwinkel für C-Winkelberechnung im BKS
62643	SPECIAL_FEATURE_MASK	Funktionsmaske

20.14.2 Signale**20.14.2.1 Signale an Kanal**

Signalname	SINUMERIK 840D sl	SINUMERIK 828D
PTP-Fahren aktivieren	DB21,DBX29.4	-

20.14.2.2 Signale von Kanal

Signalname	SINUMERIK 840D sl	SINUMERIK 828D
Transformation aktiv	DB21,DBX33.6	DB330x.DBX1.6
Nummer der aktiven G-Befehle der G-Gruppe 25 (ORIWKS, ORIMKS, ORIPATH)	DB21,DBB232	-
PTP-Fahren aktiv	DB21,DBX317.6	-

TE6: MKS-Kopplung - nur 840D sl

21.1 Kurzbeschreibung

Sind an einer Werkzeugmaschine zwei oder mehr getrennt voneinander verfahrbare Bearbeitungsköpfe vorhanden und wird zur Bearbeitung eine Transformation benötigt, können die Orientierungsachsen der Bearbeitungsköpfe nicht über die Standardkopplungsarten COUPON, TRAILON gekoppelt werden. Die Kopplungen erfolgen im Werkzeugkoordinatensystem (WKS). Die Kopplung der Orientierungsachsen muss unter den genannten Voraussetzungen aber im Maschinenkoordinatensystem (MKS) erfolgen.

Über den Compile Zyklus "MKS-Kopplung" können Kopplungen im Maschinenkoordinatensystem (MKS) mit Koppelfaktoren "1" und "-1" für eine Master-Achse, im Folgenden CC_Master genannt, und mehreren Slave-Achsen, im Folgenden CC_Slave genannt, parametrierbar werden.

Die Kopplungen sind über Befehle im Teileprogramm ein- und ausschaltbar.

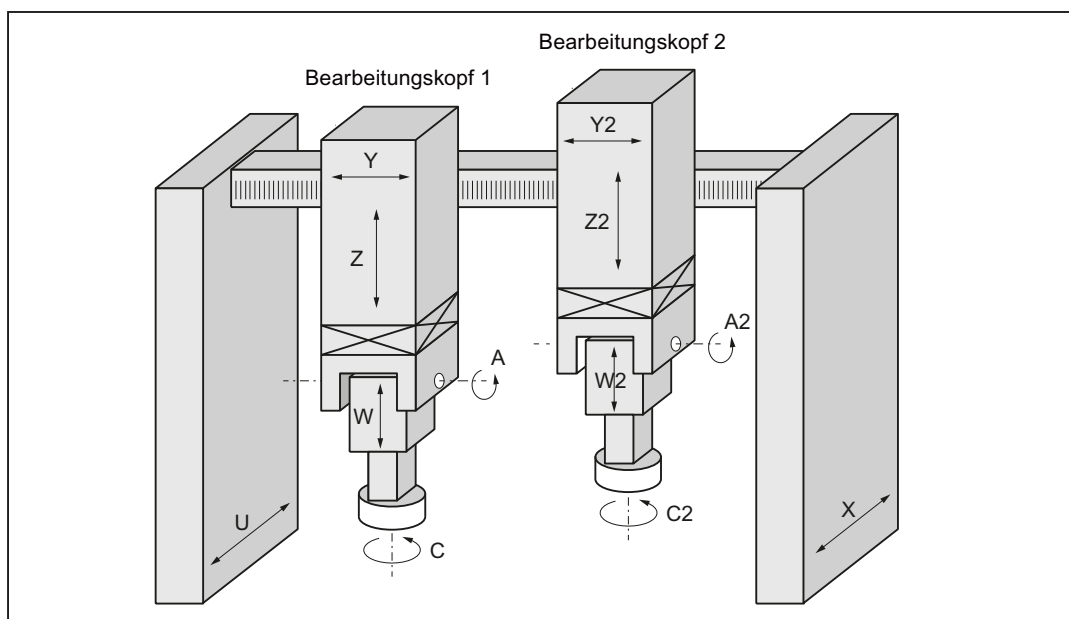


Bild 21-1 Anwendungsbeispiel: Doppelschlitten Portalfräsmaschine

Master- und Slave-Achsen

Ein CC_Master kann Master für mehrere CC_Slave sein. Ein CC_Slave kann nicht gleichzeitig CC_Master sein.

Für einen CC_Slave sind folgende Funktionen nicht möglich:

- PLC-Achse sein
- Kommandoachse
- In der Betriebsart JOG getrennt vom CC_Master verfahren

Toleranzfenster

Bei aktiver Kopplung werden die Istwerte von CC_Master und CC_Slave auf das Einhalten eines parametrierbaren Toleranzfenster überwacht.

Programmierung eines CC_Slave

Wird ein CC_Slave in einem Kanal programmiert, wird, abhängig von der Einstellung im Maschinendatum, entweder ein Alarm angezeigt oder die Achse per implizit ausgelöstem Achstausch (GET) für den Kanal angefordert:

MD30552 \$MA_AUTO_GET_TYPE = <GET-Type> (Automatisches GET bei Achse holen)

Ein-/Ausschalten der Kopplung

Eine Kopplung wird über OEM-spezifische Sprachbefehle ein- und ausgeschaltet und kann in allen Betriebsarten aktiv sein.

Kollisionsschutz

Um die Bearbeitungsköpfe im nicht gekoppelten Betrieb oder bei gespiegelter Kopplung vor Kollision zu schützen, kann ein Kollisionsschutz parametrierbar werden. Die Aktivierung erfolgt über Maschinendatum oder die NC/PLC-Nahtstelle. Die Zuordnung der geschützten Paare ist dabei unabhängig von den CC_Master und -Slave-Paaren.

21.2 Funktionsbeschreibung MKS-Kopplung

21.2.1 Kopplungspaare definieren

Einer CC_SLAVE Achse wird ihre CC_Master Achse über das folgende axiale Maschinendatum zugeordnet:

MD63540 \$MA_CC_MASTER_AXIS (Gibt zu einer CC_Slave Achse die zugehörige CC_Master Achse an)

Die an der Kopplung beteiligten Achsen können nur bei ausgeschalteter Kopplung geändert werden.

Eine CC_Slave Achse wird im folgenden axialen VDI-Out Byte angezeigt:

DB31, ... DBX97.0 (MKS-Kopplung: Slave Achse)

Voraussetzung

- CC_Master- und CC_Slave-Achse müssen entweder beide Rundachsen oder beide Linearachsen sein.
- Spindeln können nicht gekoppelt werden.
- Weder CC_Master- noch CC_Slave-Achse dürfen eine Tauschachse sein (\$MA_MASTER_CHAN[AXn]=0)

21.2.2 Kopplung EIN-/AUS schalten**Kopplung einschalten**

- Einschalten der 1:1 Kopplung. Die Toleranzfensterüberwachung ist aktiv.
CC_COPON([<Achse1>][<Achse2>][<Achse3>][<Achse4>][<Achse5>])
- Einschalten der 1:-1 Kopplung (Mirror). Die Toleranzfensterüberwachung ist nicht aktiv.
CC_COPONM([<Achse1>][<Achse2>][<Achse3>][<Achse4>][<Achse5>])

Achsname

Die Angabe einer Achse kann erfolgen mittels:

- Maschinenachsname
- Kanalachsname
- Geometrieachsname

Randbedingungen

- Eine programmierte Achse muss an einer Kopplung beteiligt sein.
- Es können CC_Master-Achsen, CC_Slave-Achsen oder beide gleichzeitig programmiert werden. Mit CC_COPON bzw. CC_COPONM werden alle definierten Kopplungen eingeschaltet.
- Wird eine Achse programmiert, die an keiner Kopplung beteiligt ist, wird ein Alarm ausgegeben.

NC/PLC-Nahtstellensignale

Alle NC/PLC-Nahtstellensignale einer Kopplung beziehen sich auf die CC_Slave-Achsen.

- DB31, ... DBX97.1 (MKS-Kopplung: Kopplung aktiv)
- DB31, ... DBX97.2 (MKS-Kopplung: Spiegeln aktiv)
- DB31, ... DBX24.2 (MKS-Kopplung: ausschalten bzw. nicht zulassen)
Bei unterdrückter Kopplung wird für die CC_Slave-Achse kein Alarm ausgegeben.

Kopplung ausschalten

CC_COPOFF([<Achse1>][<Achse2>][<Achse3>][<Achse4>][<Achse5>])

Wie Einschalten der Kopplung mit dem Unterschied, dass kein Alarm ausgegeben wird falls eine Achse programmiert ist, die an keiner Kopplung beteiligt ist.

Eine bestehende Kopplung kann auch durch das axiale NC/PLC-Nahtstellensignal der CC_Slave Achse ausgeschaltet werden.

Randbedingung

Eine Kopplung kann nur im Stillstand der Achsen ein- bzw. ausgeschaltet werden.

21.2.3 Toleranzfenster

Ein Überwachungsfenster wird über ein axiales Maschinendatum vorgegeben:

MD63541 \$MA_CC_POSITION_TOL (Überwachungsfenster)

Die Differenz der Istwerte zwischen CC_Slave Achse und CC_Master Achse darf absolut nie größer als dieser Wert sein. Bei Überschreitung wird der Alarm 70010 ausgegeben.

Die Überwachung ist nicht aktiv:

- wenn der Wert des Maschinendatums 0 ist,
- bei ausgeschalteter Kopplung,
- wenn für eine der Achsen 'Achs-/Spindelsperre' aktiviert ist,
- wenn eine Achse 'nachgeführt' wird,
- bei der 1:-1 Kopplung.

Ändert sich bei aktiver 1:1 Kopplung der zum Einschaltzeitpunkt gespeicherte Offset, wird dies angezeigt an der NC => PLC VDI-SS:

DB31, ... DBX97.3 (MKS-Kopplung: Offset-Änderung)

Hinweis

Der Offset kann sich ändern:

- wenn bei einer Achse die SW-Limitüberwachung im Hauptlauf aktiv war,
 - wenn eine Achse in Nachführen gebracht wurde,
 - wenn für eine Achse der Kollisionsschutz aktiv war.
-

21.2.4 Speicherkonfiguration: Satzspeicher

Die Technologiefunktion benötigt **zusätzliche** Daten im NC-internen Satzspeicher. Für folgende speicherkonfigurierende kanalspezifischen Maschinendaten sind die Werte zu erhöhen:

- MD28090 \$MC_MM_NUM_CC_BLOCK_ELEMENTS += 1 (Anzahl Satzelemente für Compile-Zyklen)
- MD28100 \$MN_MM_NUM_CC_BLOCK_USER_MEM += 1 (Größe des Satzspeichers für Compile-Zyklen (DRAM) in kByte)

21.3 Funktionsbeschreibung Kollisionsschutz

21.3.1 Schutzpaare definieren

Einer ProtecSlave Achse (PSlave) wird ihr ProtecMaster (PMaster) über das folgende axiale Maschinendatum zugeordnet:

MD63542 \$MA_CC_PROTECT_MASTER (Gibt zu einer PSlave Achse die zugehörige PMaster Achse an)

Die Paare können somit unabhängig von den Koppelungspaaren definiert werden. Eine PSlave Achse kann für eine andere Achse eine PMaster Achse sein. Die Achsen müssen entweder beide eine Rundachse oder Linearachse sein.

21.3.2 Kollisionsschutz EIN-/AUSschalten

Der Mindestabstand zwischen PSlave und PMaster wird in dem axialen Maschinendatum der PSlave Achse angegeben:

MD63544 \$MA_CC_COLLISION_WIN (Kollisionsschutzfenster)

Bei einem Wert kleiner 0 wird kein Kollisionsschutz eingeschaltet. Den Offset der 0 Position zwischen PSlave und PMaster gibt man in dem axialen Maschinendatum (PSlave-Achse) an:

MD63545 \$MA_CC_OFFSET_MASTER (Nullpunkt-Offset zwischen PSlave und PMaster)

Bevor der Kollisionsschutz eingeschaltet wird, muss man getrennt für jede Achse im folgenden Maschinendatum die Überwachung für die jeweilige Achse freigeben:

MD63543 \$MA_CC_PROTECT_OPTIONS

Im gleichen Maschinendatum der PSlave Achse wird angegeben, ob der Kollisionsschutz immer aktiv ist oder ob er über die VDI-Nahtstelle (PLC => NC) geschaltet wird:

DB31, ... DBX24.3

Ist der Kollisionsschutz aktiv, werden im IPO-Takt die Sollpositionen des nächsten IPO-Taktes von PSlave und PMaster mittels der aktuellen Sollposition und der aktuellen Geschwindigkeit extrapoliert und überwacht.

Wird der Mindestabstand unterschritten, bremsen die Achsen mit der projektierten Maximalbeschleunigung ab:

MD32300 \$MA_MAX_AX_ACCEL (Achsbeschleunigung)

Oder mit einer um 20% erhöhten Beschleunigung, festgelegt über das Maschinendatum:

MD63543 \$MA_CC_PROTECT_OPTIONS

Sobald die Achsen stehen, wird ein Alarm ausgegeben.

<p>⚠️ WARNUNG</p> <p>Kollisionsgefahr beim Anfahren</p> <p>Werden die Achsen zwangsgebremst, sind die angezeigten Positionen im Werkstückkoordinatensystem falsch!</p> <p>Diese werden erst mit RESET neu synchronisiert.</p>

Befinden sich die Achsen beim Einschalten des Kollisionsschutzes schon innerhalb des Mindestabstandes, können sie nur noch in eine Richtung verfahren werden (Freifahrtrichtung). Die Freifahrtrichtung wird vorgegeben im Maschinendatum:

MD63543 \$MA_CC_PROTECT_OPTIONS

In dem axialen VDI-Out Byte von PSlave wird optional der Kollisionsschutz-Zustand angezeigt:

DB31, ... DBX66.0 (MKS-Kopplung: Kollisionsschutz aktiv)

- DB31, ... DBX66.0=1 → Kollisionsschutz aktiv
- DB31, ... DBX66.0=0 → Kollisionsschutz inaktiv

Diese Ausgabe wird aktiviert über Bit7 im Maschinendatum der PSlave-Achse:

MD63543 \$MA_CC_PROTECT_OPTIONS

21.3.3 Projektierungsbeispiel

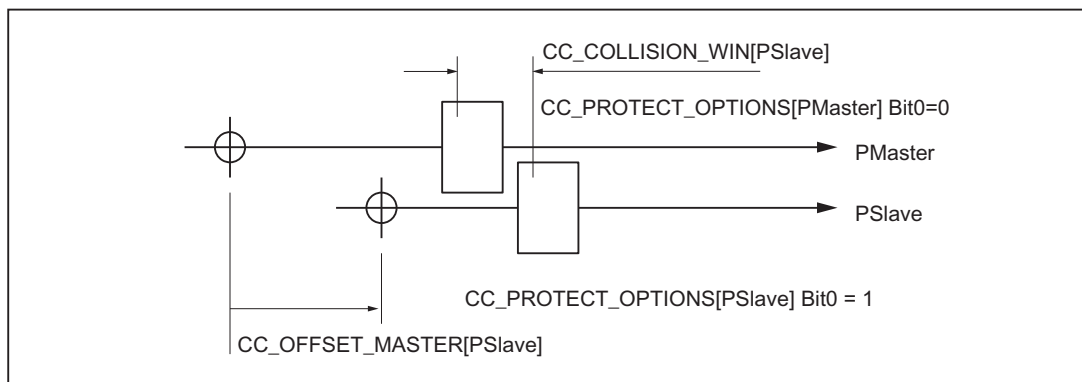


Bild 21-2 Projektierungsbeispiel

Hinweis

Da der Kollisionsschutz den Zielpunkt aus der "aktuellen Geschwindigkeit + der maximalen Beschleunigung (bzw. +20%)" extrapoliert, kann es bei reduzierten Beschleunigungen zu nicht erwartetem Auslösen des Überwachungsalarms kommen:

Beispiel:

PMaster = X, PSlave = X2, \$MA_CC_COLLISION_WIN = 10mm

Startpunkt im Teileprogramm: X=0.0 X2=20.0

```
N50 G0 X100 X2=90
```

; der Überwachungsalarm wird ausgelöst, weil X und X2 zusammen interpolieren: deshalb ist die Beschleunigung von X2 < der Maximalbeschleunigung.

Abhilfe:

- N50 G0 POS[X]=100 POS[X2]=90
 - oder die Überwachung ausschalten.
-

21.4 Anwenderspezifische Projektierungen

Bearbeitungskopf parken

Parken bedeutet hier, dass der entsprechende Bearbeitungskopf an der Werkstückbearbeitung nicht beteiligt ist. Alle Achsen sind in Lageregelung und stehen im Genauhalt.

Auch wenn nur mit einem Bearbeitungskopf gefertigt wird, sollte die Kopplung aktiv sein! Dies ist vor allem dann zwingend nötig, falls nur mit dem zweiten Kopf (Y2....) gefertigt wird. Für die Achsen des geparkten Kopfes muss dann axial (PLC -> NC) "Achs-/ Spindel- sperre" gesetzt werden.

Hinweis

Bei aktiver Achs-/Spindelsperre kann ein Teileprogramm auch dann abgearbeitet werden, wenn diese Achse nicht in Lageregelung ist.

Spindelfunktionalitäten

Da für die Spindel keine MKS-Kopplung eingeschaltet werden kann, müssen dafür andere Lösungswege projektiert werden.

- Spindel positionieren ($SPOS= \dots$)
Anstelle von $SPOS$ wird ein Zyklus aufgerufen. In diesem Zyklus wird $SPOS$ für alle aktiven Spindeln aufgerufen.
- Drehzahlvorgabe
Drehzahl- und Drehrichtungsvorgabe kann über Synchronaktionen bzw. über PLC erkannt und an alle weiteren aktiven Spindeln weitergegeben werden.
- Synchronspindelfunktion

21.5 Besondere Betriebszustände

Reset

Die Kopplungen können über $RESET$ hinweg aktiv bleiben.

Reorg

Keine vom Standart abweichenden Funktionalitäten.

Satzsuchlauf

Bei Satzsuchlauf wird immer der letzte Satz, der einen OEM-spezifischen Sprachbefehl enthält, gespeichert und dann mit dem letzten Aktionssatz ausgegeben. In nachfolgenden Beispielen soll das veranschaulicht werden. Die Ausgangspositionen der Achsen sind immer 0.

Beispiel 1:

```
N01 M3 S1000
N02 G01 F1000 X10 Y10
N03 CC_COPON( X, Y)
ZIEL:
```

Wird dieses Programm normal gestartet, fahren die Achsen X und Z ungekoppelt nach $X10$ $Z10$. Nach Satzsuchlauf auf $ZIEL$: fahren die Achsen X und Y gekoppelt auf diese Position!

Beispiel 2:

```
N01 M3 S1000
N02 CC_COPON( X)
N03 G01 F1000 X100 Y50
N04 CC_COPOFF( X)
N05 CC_COPON( Y)
```

```
N06 Y100
```

```
N10 CC_COPOFF()
```

ZIEL:

Nach Satzsuchlauf auf ZIEL: fahren die Achsen ungekoppelt auf X100 Y100.

Beispiel 3:

```
N01 CC_COPON( X, Y, Z)
```

```
N02 ...
```

```
...
```

```
N10 CC_COPOFF( Z)
```

ZIEL:

Nach Satzsuchlauf auf ZIEL: ist keine Kopplung aktiv !

Einzelatz

Es gibt keine vom Standard abweichenden Funktionalitäten.

21.6 Randbedingungen

Gültigkeit

Die Funktion ist nur für den ersten Kanal projiziert.

Abbremsverhalten

Abbremsverhalten am SW-Limit bei Bahnachsen

Der Faktor der programmierbaren Beschleunigung ACC zum Abbremsen am SW-Limit bezieht sich auf Bahnachsen.

Bei der MKS-Kopplung handelt es sich um Hauptachsen, die aufgrund ihrer geometrischen Anordnung als Geometrieachsen bezeichnet werden.

Abbremsen von Geometrieachsen mit Synchronaktionen

Die wie bei Bahnachsen gewünschte höhere Abbremsung kann bei Geometrieachsen mit einer Synchronaktion wie folgt realisiert werden:

```
ACC[x2]=190
```

Axiale Beschleunigungsbegrenzung bei G0

Das folgende Maschinendatum wird von der MKS-Kopplung nicht berücksichtigt:

```
MD32434 $MA_G00_ACCEL_FACTOR (Skalierung der Beschleunigungsbegrenzung bei G0)
```

Ein vom Standard abweichender Wert führt zu einer Beeinflussung der Bremsrampe bis zum Software-Endschalter.

21.7 Datenlisten

21.7.1 Maschinendaten

21.7.1.1 Kanal-spezifische Maschinendaten

Nummer	Bezeichner: \$MC_	Beschreibung
28090	NUM_CC_BLOCK_ELEMENTS	Anzahl der Satzelemente für Compile-Zyklen.
28100	NUM_CC_BLOCK_USER_MEM	Gesamtgröße des nutzbaren Satzspeichers für Compile-Zyklen

21.7.1.2 Achs-/Spindel-spezifische Maschinendaten

Nummer	Bezeichner: \$MA_	Beschreibung
63540	CC_MASTER_AXIS	Gibt zu einer CC_Slave-Achse die zugehörige CC_Master-Achse an.
63541	CC_POSITION_TOL	Überwachungsfenster
63542	CC_PROTEC_MASTER	Gibt zu einer PSlave-Achse die zugehörige PMaster-Achse an.
63543	CC_PROTEC_OPTIONS	
63544	CC_COLLISION_WIN	Kollisionsschutzfenster
63545	CC_OFFSET_MASTER	Nullpunkt-Offset zwischen PSlave und PMaster

21.7.2 Signale

21.7.2.1 Signale an Achse/Spindel

Signalname	SINUMERIK 840D sl	SINUMERIK 828D
MKS-Kopplung: ausschalten bzw. nicht zulassen	DB31, ... DBX24.2	-
MKS-Kopplung: Kollisionsschutz einschalten	DB31, ... DBX24.3	-

21.7.2.2 Signale von Achse/Spindel

Signalname	SINUMERIK 840D sl	SINUMERIK 828D
MKS-Kopplung: Kollisionsschutz aktiv	DB31, ... DBX66.0	-
MKS-Kopplung: Slave-Achse	DB31, ... DBX97.0	-
MKS-Kopplung: Kopplung aktiv	DB31, ... DBX97.1	-

Signalname	SINUMERIK 840D sl	SINUMERIK 828D
MKS-Kopplung: Spiegeln aktiv	DB31, ... DBX97.2	-
MKS-Kopplung: Offset-Änderung	DB31, ... DBX97.3	-

22.1 Kurzbeschreibung

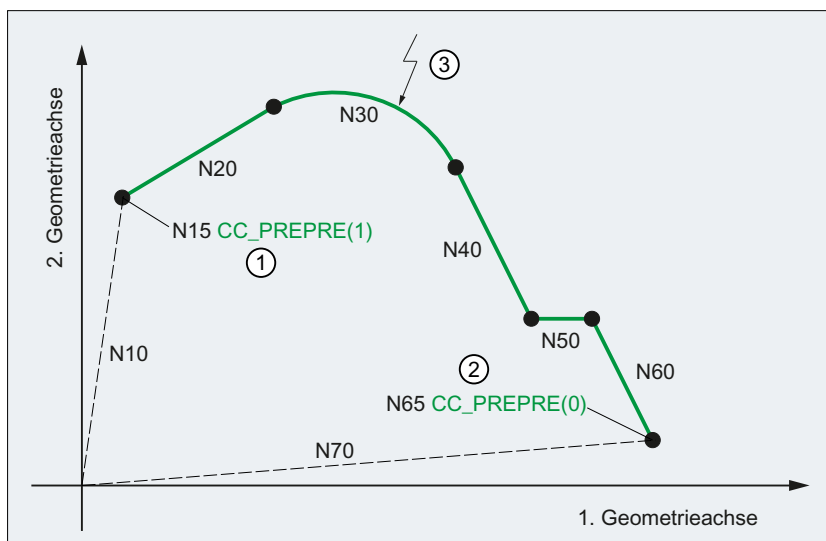
Funktion

Die Technologie-Funktion "Wiederaufsetzen - Retrace Support (RESU)" unterstützt das Wiederaufnehmen von unterbrochenen 2-dimensionalen Bearbeitungsvorgängen, wie z. B. Laserschneiden, Wasserstrahlschneiden etc.

RESU ermöglicht es dem Maschinenbediener, bei einer Störung des Bearbeitungsvorgangs, z. B. Ausfall des Laserstrahls, ohne genaue Kenntnis des aktiven Teileprogramms die Bearbeitung zu unterbrechen und vom Unterbrechungspunkt bis zu einem bearbeitungstechnisch notwendigen Wiederaufsetzpunkt entlang der Kontur zurückzufahren.

Nach Erreichen des Wiederaufsetzpunkts löst der Maschinenbediener das Wiederaufsetzen aus. Im Rahmen des Wiederaufsetzens erfolgt implizit ein Satzsuchlauf mit Berechnung an der Kontur mit anschließendem Repositionieren an die Kontur und automatischer Wiederaufnahme der Teileprogrammabarbeitung.

Die Möglichkeit zum Wiederaufsetzen innerhalb eines Bearbeitungsprogramms wird durch die Programmierung von RESU-Start- und Endpunkten über die Prozedur CC_PREPRE(...) erreicht. Innerhalb eines so definierten Konturbereichs kann der Wiederaufsetzpunkt beliebig gewählt werden.



- ① RESU-Startpunkt bzw. Anfang des RESU-fähigen Konturbereichs
- ② RESU-Endpunkt bzw. Ende des RESU-fähigen Konturbereichs
- ③ Beispielhafter Wiederaufsetzpunkt

Bild 22-1 Programmierte Kontur mit Wiederaufsetz- und Unterbrechungspunkt

Ein konturgenaues Rückwärtsfahren ist auf allen Konturen möglich, die sich aus programmierten Geraden- und Kreiselementen zusammensetzen. Andere Konturelemente wie z. B. Splines oder automatisch eingefügte nichtlineare Konturelemente (Kreis, Parabel etc. z. B. durch Werkzeugradiuskorrektur), werden beim Rückwärtsfahren als Gerade zwischen Anfangs- und Endpunkt des entsprechenden Konturelements abgebildet und erlauben daher kein konturgenaues Rückwärtsfahren.

Funktionskürzel

Das Kürzel der Technologie-Funktion "Wiederaufsetzen - Retrace Support" für funktionsspezifische Bezeichner von Programmbefehlen, Maschinendaten etc. ist:

RESU (= REtrace SUpport)

Einschränkungen

Für die Anwendung der Technologie-Funktion "Wiederaufsetzen - Retrace Support" gelten folgende Einschränkungen:

- **Die Technologie-Funktion ist nur im 1. Kanal der NC verfügbar.**
- Wiederaufsetzen bzw. Rückwärtsfahren ist nur für Teileprogrammsätze möglich, die Verfahrsätze in der projektierten RESU-Arbeitsebene (z. B. 1. und 2. Geometrieachse des Kanals, siehe Kapitel "Festlegung der RESU-Arbeitsebene (Seite 851)") enthalten.

Literatur

Die Technologie-Funktion "Wiederaufsetzen - Retrace Support" ist ein Compile-Zyklus. Für die Handhabung von Compile-Zyklen (siehe Kapitel "TE01: Installation und Aktivierung ladbarer Compile-Zyklen (Seite 689)").

22.2 Funktionsbeschreibung

22.2.1 Funktion

Satzsuchlauf mit Berechnung an der Kontur

Um an einer bestimmten Stelle eines Teileprogramms mit der unterbrochenen Bearbeitung wieder aufsetzen zu können, besteht die Möglichkeit, einen Satzsuchlauf mittels der Standard-Funktion "Satzsuchlauf mit Berechnung an der Kontur" zu verwenden. Dazu ist aber die genaue Kenntnis des Teileprogramms notwendig, um die zum Satzsuchlauf erforderliche Satznummer des Teileprogrammsatzes, auf den der Satzsuchlauf erfolgen soll, angeben zu können.

Wiederaufsetzen - Retrace Support

Die Technologie-Funktion "Wiederaufsetzen - Retrace Support" unterstützt das Wiederaufsetzen des Bearbeitungsvorgangs durch einen impliziten Satzsuchlauf mit Berechnung an der Kontur, ohne dass der Maschinenbediener den dafür notwendigen Teileprogrammsatz kennen muss.

Ein Wiederaufsetzen ist zum Beispiel erforderlich, wenn beim Laserschneiden der Laserstrahl während des Bearbeitungsvorganges ausfällt und an der Unterbrechungsstelle mit der Bearbeitung wieder aufgesetzt werden soll.

RESU unterstützt das Wiederaufsetzen durch folgende, automatisch ablaufende Teilfunktionen:

- Funktionsspezifisches Rückwärtsfahren auf der Kontur bis zum gewünschten Wiederaufsetzpunkt
- Automatische Ermittlung des zum Wiederaufsetzpunkt gehörenden Teileprogrammsatzes
- Satzsuchlauf mit Berechnung an der Kontur auf den ermittelten Teileprogrammsatz
- Repositionieren an die Kontur im Wiederaufsetzpunkt
- Fortsetzen der Teileprogrammbearbeitung

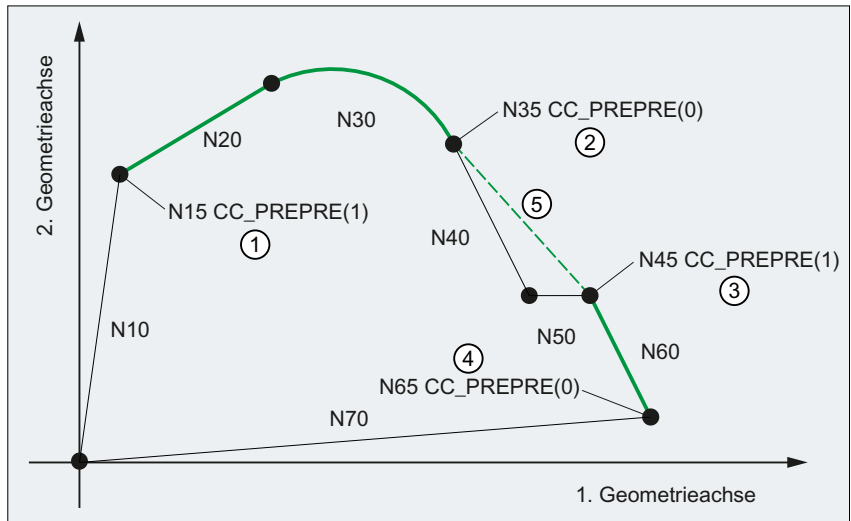
Um den erforderlichen Wiederaufsetzpunkt exakt anfahren zu können, ist während des Wiederaufsetzens auch ein mehrfacher Wechsel zwischen Rückwärts- und Vorwärtsfahren auf der Kontur möglich.

RESU-fähige Konturbereiche

RESU wird durch Programmierung des funktionsspezifischen Teileprogrammfehls `CC_PREPRE (1)` eingeschaltet. Nur der Konturbereich ist im Sinne von RESU wiederaufsetzfähig (RESU-fähig), der zwischen dem RESU-Start (`CC_PREPRE (1)`) und dem Unterbrechungspunkt (NC-Stopp) liegt.

Nach RESU-Start werden alle Teileprogrammsätze, in denen Verfahrbewegungen programmiert sind, von RESU zum eventuellen späteren Rückwärtsfahren protokolliert. Konturbereiche, für die ein Wiederaufsetzen keinen Sinn macht, können mittels RESU-Stopp `CC_PREPRE (0)` vom Protokollieren ausgenommen werden.

Nicht protokollierte Konturbereiche werden beim Rückwärts- / Vorwärtsfahren durch Geraden zwischen Anfangs- und Endpunkt der protokollierten Konturbereiche überbrückt.



- ① RESU-Startpunkt 1 bzw. Anfang des RESU-fähigen Konturbereichs 1
 - ② RESU-Endpunkt 1 bzw. Ende des RESU-fähigen Konturbereichs 1
 - ③ RESU-Startpunkt 2 bzw. Anfang des RESU-fähigen Konturbereichs 2
 - ④ RESU-Endpunkt 2 bzw. Ende des RESU-fähigen Konturbereichs 2
 - ⑤ RESU-Gerade als Ersatzkontur des **nicht** RESU-fähigen Konturbereichs
- Bild 22-2 RESU-fähige Konturbereiche

22.2.2 Begriffsdefinitionen

Unterbrechungspunkt

Der Unterbrechungspunkt ist der Punkt der Kontur, an dem die Verfahrbewegung nach NC-Stopp zum Stillstand kommt und das Rückwärtsfahren ausgelöst wird.

Wiederaufsetzpunkt

Der Wiederaufsetzpunkt ist der Punkt der Kontur, auf dem das Rückwärtsfahren beendet und das Wiederaufsetzen ausgelöst wird.

RESU-fähiger Konturbereich

RESU-fähige Konturbereiche setzen sich aus Verfahrstrichen in der projektierten RESU-Arbeitsebene (z. B. 1. und 2. Geometrieachse des Kanals) zusammen, die im Teileprogramm zwischen dem RESU-Start-Befehl `CC_PREPRE(1)` und dem RESU-Stopp-Befehl `CC_PREPRE(0)` programmiert sind (siehe "Bild 22-2 RESU-fähige Konturbereiche (Seite 846)").

22.2.3 Funktionsablauf (Prinzip)

Im Folgenden ist der prinzipielle Ablauf der Funktion RESU zwischen Unterbrechungspunkt, Wiederaufsetzpunkt und dem Fortsetzen der Teileprogrammbearbeitung beschrieben.

Voraussetzungen

Ein Teileprogramm mit Verfahrshätzen in der projektierten RESU-Arbeitsebene (Seite 851) sowie dem Befehl für den RESU-Start ist im 1. Kanal gestartet.

Funktionsablauf

1. Teileprogrammbearbeitung unterbrechen:
Die Teileprogrammbearbeitung bzw. die Verfahrbewegung kann eine beliebige Anzahl von Verfahrshätzen nach RESU-Start mittels NC-Stop unterbrochen werden.
2. Rückwärtsfahren anwählen:
Die Anwahl des Rückwärtsfahrens erfolgt per NC/PLC-Nahtstellensignal:
DB21, ... DBX0.1 = 1
3. Rückwärtsfahren:
Mit NC-Start wird die Kontur in der RESU-Arbeitsebene (Seite 851) rückwärts abgefahren. RESU wählt dazu an Stelle des aktuellen Bearbeitungsprogramms das automatisch erzeugte RESU-Hauptprogramm (Seite 859) an.
4. Rückwärtsfahren beenden:
Ist der gewünschte Wiederaufsetzpunkt auf der Kontur erreicht, wird das Rückwärtsfahren mittels NC-Stop beendet.
5. Vorwärtsfahren anwählen (optional):
Zum Vorwärtsfahren ist das Rückwärtsfahren abzuwählen per NC/PLC-Nahtstellensignal:
DB21, ... DBX0.1 = 0
6. Vorwärtsfahren (optional):
Mit NC-Start wird die Kontur in der RESU-Arbeitsebene (Seite 851) in Vorwärtsrichtung abgefahren.
7. Vorwärtsfahren beenden (optional):
Ist der gewünschte Wiederaufsetzpunkt auf der Kontur erreicht, wird das Vorwärtsfahren mittels NC-Stop beendet.

22.2 Funktionsbeschreibung

8. Wiederaufsetzen:
Das Wiederaufsetzen wird ausgelöst per PLC-Nahtstellensignal:
DB21, ... DBX0.2 = 1 (Wiederaufsetzen starten)
Zum Wiederaufsetzen wählt RESU automatisch das ursprüngliche Bearbeitungsprogramm an und veranlasst einen Satzvorlauf mit Berechnung bis zum Wiederaufsetzpunkt.
9. Fortsetzen der Teileprogrammbearbeitung:
Das Fortsetzen der Teileprogrammbearbeitung ab dem Wiederaufsetzpunkt erfolgt entsprechend der Standardfunktion "Satzvorlauf mit Berechnung" mittels zweier aufeinander folgender NC-Start-Kommandos.
Mit dem ersten NC-Start-Kommando werden die Aktionssätze ausgeführt. Das Wiederaufsetz-ASUP CC_RESU_BS_ASUP.SPF (Seite 863) wird ausgelöst mit Erreichen des letzten Aktionssatzes:
DB21, ... DBX32.6 == 1 (letzter Aktionssatz aktiv)
Mit dem zweiten NC-Start-Kommando wird der Anfahrsatz ausgeführt und anschließend die Teileprogrammbearbeitung fortgesetzt.

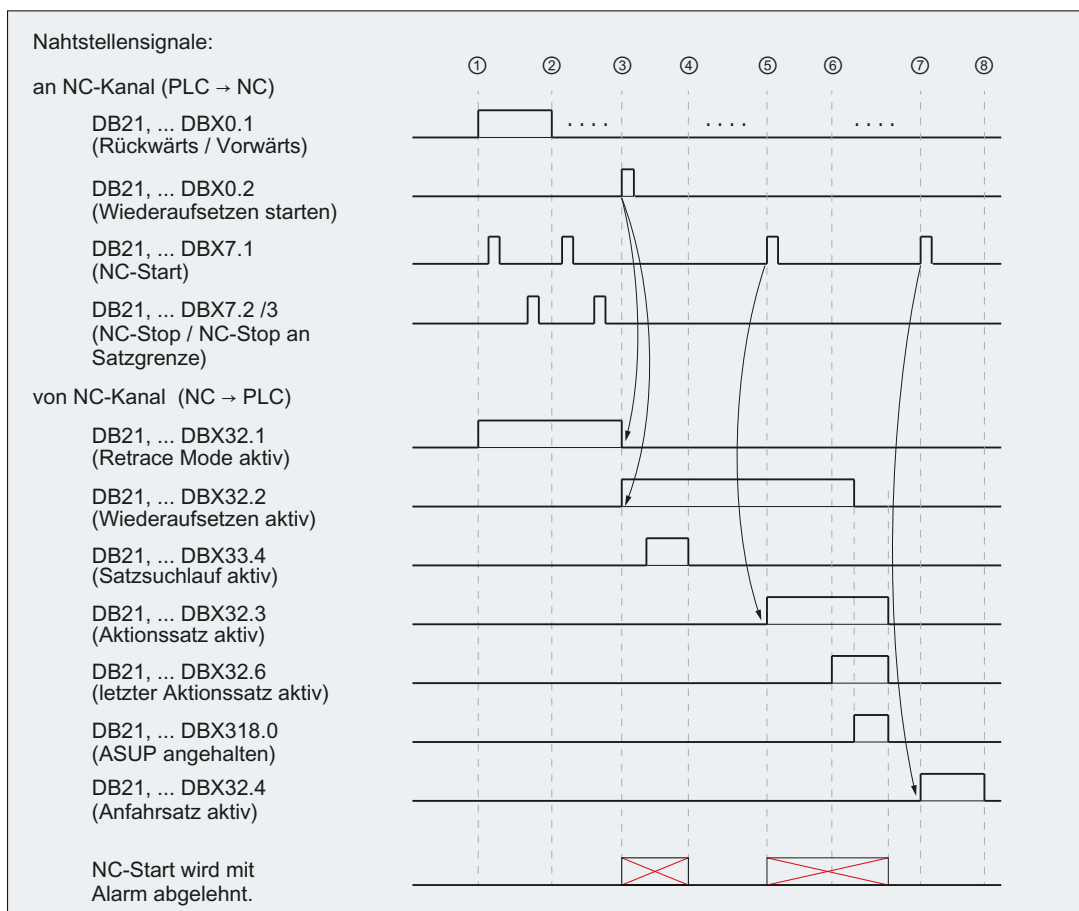
Hinweis

Die Punkte 3. bis 8. können beliebig oft wiederholt werden.

Nach dem Wiederaufsetzen ist ein erneutes Rückwärtsfahren maximal bis zum letzten Wiederaufsetzpunkt möglich.

Signalverlauf der NC/PLC-Nahtstellensignale

Der prinzipielle Ablauf der Funktion RESU ist im folgenden Bild als Signalverlauf der beteiligten NC/PLC-Nahtstellensignale dargestellt:



- ① Rückwärtsfahren wird gestartet.
- ② Vorwärtsfahren wird gestartet (optional).
- ③ Wiederaufsetzen wird gestartet (Satzsuchlauf).
- ④ Suchlaufziel (Zielsatz) wurde gefunden.
- ⑤ 1. NC-Start → Aktionssätze werden ausgegeben.
- ⑥ Letzter Aktionssatz wird aktiv.
Mit dem Aktivieren des letzten Aktionssatzes wird das RESU-ASUP CC_RESU_BS_ASUP.SPF (Seite 863) ausgelöst.
- ⑦ 2. NC-Start → Der Anfahrsatz zum Wiederaufsetzpunkt wird abgefahren.
- ⑧ Die Teileprogrammbearbeitung (Zielsatz) wird fortgesetzt

Bild 22-3 Signalverlauf

22.2.4 Maximaler RESU-fähiger Konturbereich

Beim mehrmaligen Wiederaufsetzen innerhalb eines Konturbereichs ist das Rückwärtsfahren auf der Kontur immer nur bis zum letzten Wiederaufsetzpunkt (**W**) möglich. Beim erstmaligen Rückwärtsfahren nach RESU-Start kann bis zum Anfang des Konturbereichs zurückgefahren werden.

Dieses Verhalten soll durch folgende Grafik veranschaulicht werden. Der Einfachheit halber sei der Unterbrechungspunkt (**U**) immer derselbe:

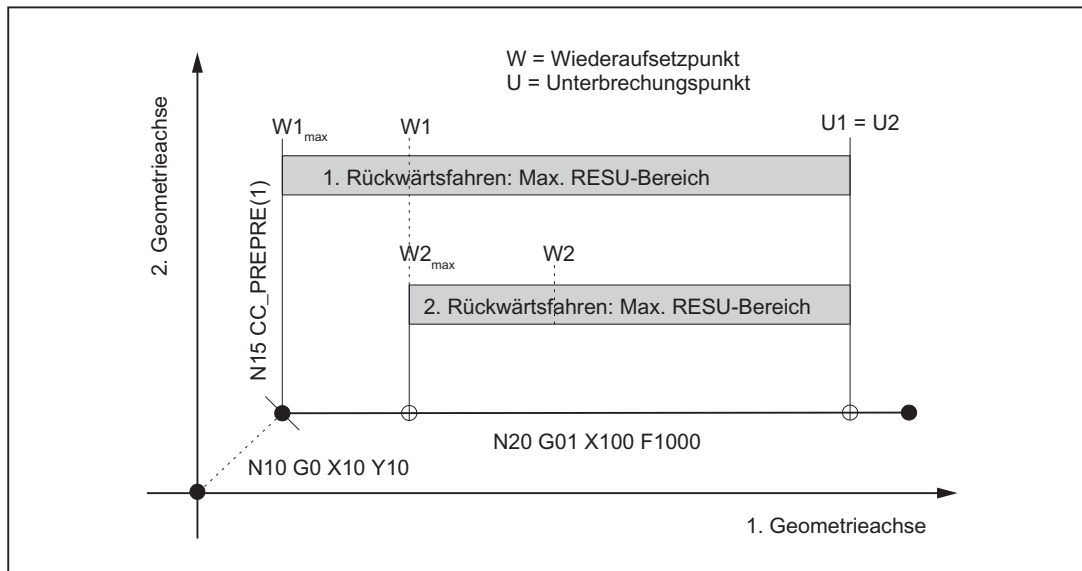


Bild 22-4 Maximaler RESU-fähiger Konturbereich

1. Rückwärtsfahren

Beim erstmaligen Rückwärtsfahren kann maximal bis zum Anfang des ersten Konturelements (N20) nach RESU-Start (N15) rückwärts gefahren werden ($W1_{max}$).

Wird bis zum Wiederaufsetzpunkt W1 zurückgefahren, legt W1 nach Wiederaufsetzen und Vorwärtsfahren den maximalen RESU-Bereich für ein eventuell weiteres Rückwärtsfahren fest.

2. Rückwärtsfahren

Ein erneutes Rückwärtsfahren kann jetzt nur noch maximal bis zum letzten Wiederaufsetzpunkt $W2_{max} = W1$ erfolgen.

Wird bis zum Wiederaufsetzpunkt W2 zurückgefahren, wird der maximale RESU-Bereich weiter eingeschränkt.

22.3 Inbetriebnahme

22.3.1 Aktivierung

Vor Inbetriebnahme der Technologie-Funktion ist sicherzustellen, dass der entsprechende Compile-Zyklus geladen und aktiviert ist (siehe auch Kapitel "TE01: Installation und Aktivierung ladbarer Compile-Zyklen (Seite 689)").

Aktivierung

Die Technologie-Funktion "Wiederaufsetzen - Retrace Support" wird über folgendes Maschinendatum aktiviert:

MD60900+i \$MN_CC_ACTIVE_IN_CHAN_RESU[0], Bit 0 = 1

Hinweis

Die Technologie-Funktion "Wiederaufsetzen - Retrace Support" ist nur im **1. Kanal** der NC verfügbar.

22.3.2 Festlegung der RESU-Arbeitsebene

Wiederaufsetzen bzw. Rückwärtsfahren ist nur für Teileprogrammsätze möglich, die Verfahrssätze in der projektierten RESU-Arbeitsebene enthalten.

Die Festlegung der RESU-Arbeitsebene erfolgt mit dem Maschinendatum:

MD62580 \$MC_RESU_WORKING_PLANE

Wert	Bedeutung
1	Die RESU-Arbeitsebene wird von der 1. und 2. Geometrieachse des 1. Kanals gebildet (für G17).
2	Die RESU-Arbeitsebene wird von der 1. und 3. Geometrieachse des 1. Kanals gebildet (für G18).
3	Die RESU-Arbeitsebene wird von der 2. und 3. Geometrieachse des 1. Kanals gebildet (für G19).

22.3.3 Speicherkonfiguration: Satzspeicher

Die Technologiefunktion benötigt **zusätzliche** Daten im NC-internen Satzspeicher. Für folgende speicherkonfigurierende kanalspezifischen Maschinendaten sind die Werte zu erhöhen:

- MD28090 \$MC_MM_NUM_CC_BLOCK_ELEMENTS += **4** (Anzahl Satzelemente für Compile-Zyklen)
- MD28100 \$MN_MM_NUM_CC_BLOCK_USER_MEM += **20** (Größe des Satzspeichers für Compile-Zyklen (DRAM) in kByte)

22.3.4 Speicherkonfiguration: Heap-Speicher

Speicherbedarf

RESU benötigt Compile-Zyklen-Heap-Speicher für folgende funktionspezifische Puffer:

- **Satzpuffer**
Je größer der Satzpuffer (siehe "Bild 22-6 RESU-spezifische Teileprogramme (Seite 859)") ist, desto mehr Teileprogrammsätze können rückwärts gefahren werden.
Pro Teileprogrammsatz werden 32 Byte benötigt.
Der Satzpuffer ist direkt parametrierbar.
- **Satzsuchlaufpuffer**
Im Satzsuchlaufpuffer liegen die zur Bearbeitung von Unterprogrammaufrufen im Rahmen von RESU benötigten Informationen.
Pro Unterprogrammaufruf werden 180 Bytes benötigt. Für den Satzsuchlaufpuffer werden daher mindestens 2880 Bytes benötigt (16 Unterprogrammebenen à 180 Bytes).
Der Satzsuchlaufpuffer ist nicht direkt parametrierbar.
Über eine funktionspezifische GUD-Variable wird die Größe des Satzsuchlaufpuffers angezeigt (zum Anlegen der GUD-Variablen, siehe Kapitel "Kanalspezifische GUD-Variable (Seite 868)").

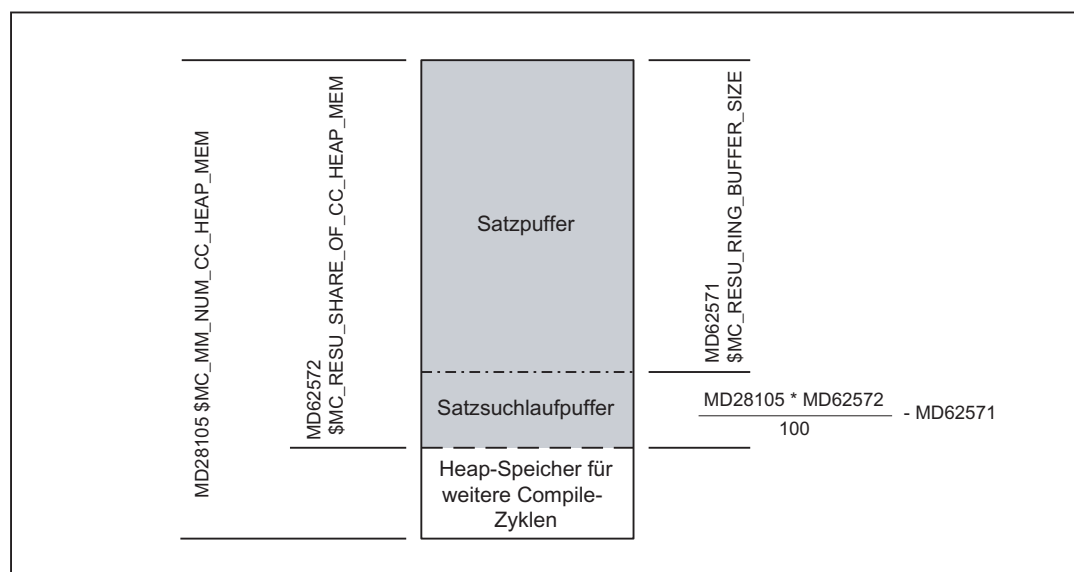


Bild 22-5 Aufteilung des Heap-Speichers für Compile-Zyklen

Speicherkonfiguration

Größe des Compile-Zyklen-Heap-Speichers

Die Größe des vom Anwender für Compile-Zyklen nutzbaren Heap-Speichers in kByte wird eingestellt über das speicherkonfigurierende kanalspezifische Maschinendatum:

MD28105 \$MC_MM_NUM_CC_HEAP_MEM

Für RESU wird der bereits vorhandene Maschinendatenwert (x) wie folgt angepasst:

MD28105 \$MC_MM_NUM_CC_HEAP_MEM = x + 50

Größe des Satzpuffers

Die Größe des Satzpuffers wird eingestellt über das Maschinendatum:

MD62571 \$MC_RESU_RING_BUFFER_SIZE

Standardmäßige Einstellung:

MD62571 \$MC_RESU_RING_BUFFER_SIZE = 1000

RESU-Anteil am gesamten Heap-Speicher

Der RESU-Anteil am gesamten, vom Anwender für Compile-Zyklen nutzbaren Heap-Speichers wird eingestellt über das Maschinendatum:

MD62572 \$MC_RESU_SHARE_OF_CC_HEAP_MEM

Standardmäßige Einstellung:

MD62572 \$MC_RESU_SHARE_OF_CC_HEAP_MEM = 100

Fehlermeldungen

Für den Satzsuchlaufpuffer müssen mindestens 2880 Bytes (entsprechend 16 Unterprogrammebenen à 180 Bytes) vorhanden sein. Ansonsten wird im Hochlauf der NC folgender Alarm angezeigt:

Alarm 75600 "Kanal 1 Retrace Support: falsche MD-Konfiguration, Fehler-Nr. 5"

Ist der Satzsuchlaufpuffer zur Laufzeit nicht ausreichend, wird folgender Alarm angezeigt:

Alarm 75606 "Kanal 1 retrace-fähige Kontur wurde verkürzt"

22.3.5 Speicherbereich des RESU-Hauptprogramms**Speicherkonfiguration**

Über das folgende Maschinendatum kann der Ablageort des RESU-Hauptprogramms CC_RESU.MPF (siehe Kapitel "Hauptprogramm (CC_RESU.MPF) (Seite 859)") eingestellt werden:

MD62574 \$MC_RESU_SPECIAL_FEATURE_MASK (Zusätzlich RESU-Eigenschaften)

Bit	Wert	Bedeutung
1	0	Das RESU-Hauptprogramm wird im dynamischen NC-Speicher abgelegt (Voreinstellung).
	1	Das RESU-Hauptprogramm wird im statischen NC-Speicher abgelegt.

Ablage im dynamischen NC-Speicher (Voreinstellung)

Wenn das RESU-Hauptprogramm im dynamischen NC-Speicher angelegt wird, dann muss der dem Anwender für die Dateiablage zur Verfügung stehende Speicherbereich wie folgt vergrößert werden:

MD18351 \$MN_MM_DRAM_FILE_MEM_SIZE = x^1 + 100

¹⁾ bereits vorhandener Maschinendatenwert

Ablage im statischen NC-Speicher

Wenn das RESU-Hauptprogramm im statischen NC-Speicher angelegt wird, dann bleibt es über POWER OFF hinaus erhalten. Da RESU das RESU-Hauptprogramm aber bei jedem Wiederaufsetzen neu erzeugt, wird diese Parametrierung nicht empfohlen.

22.3.6 Ablage der RESU-Unterprogramme

Ablage als Anwender- oder Hersteller-Zyklen

Die folgenden RESU-spezifischen Unterprogramme können als Anwender- oder Hersteller-Zyklen abgelegt werden:

- INI-Programm: CC_RESU_INI.SPF
- END-Programm CC_RESU_END.SPF
- Wiederaufsetz-ASUP CC_RESU_BS_ASUP.SPF
- RESU-ASUP CC_RESU_BS_ASUP.SPF

Die Einstellung erfolgt über das Maschinendatum:

MD62574 \$MC_RESU_SPECIAL_FEATURE_MASK (Zusätzlich RESU-Eigenschaften)

Bit	Wert	Bedeutung
2	0	Die RESU-spezifischen Unterprogramme werden als Anwender-Zyklen abgelegt (Voreinstellung).
	1	Die RESU-spezifischen Unterprogramme werden als Hersteller-Zyklen abgelegt.

Serieninbetriebnahme

Aufgrund der Voreinstellung von MD62574 Bit 2 werden im erstmaligen Hochlauf der NC nach Aktivierung der Technologie-Funktion die RESU-spezifischen Unterprogramme mit ihren voreingestellten Inhalten als Anwender-Zyklen abgelegt.

Wird anschließend eingestellt, dass die RESU-spezifischen Unterprogramme als Hersteller-Zyklen abgelegt werden, bleiben die bereits als Anwender-Zyklen angelegten RESU-spezifischen Unterprogramme auch nach einem erneuten Hochlauf erhalten und müssen gelöscht werden.

Zur Unterstützung der Serieninbetriebnahme kann eingestellt werden, dass als Anwender-Zyklen vorhandene RESU-spezifische Unterprogramme im Hochlauf der NC ohne Nachfrage gelöscht werden:

MD62574 \$MC_RESU_SPECIAL_FEATURE_MASK, Bit 3 = 1

22.3.7 ASUP-Freigabe

Hinweis

Voraussetzung für den Einsatz von ASUPs ist die Verfügbarkeit der Option "Betriebsartübergreifende Aktionen".

Zur Startfreigabe des RESU-spezifischen ASUP "CC_RESU_ASUP.SPF" während sich der Kanal im NC-Stopp-Zustand befindet, sind Maschinendaten wie folgt zu parametrieren:

MD11602 \$MN_ASUP_START_MASK, Bit 0 = 1 (Stoppgründe für ASUP ignorieren)

MD11604 \$MN_ASUP_START_PRIO_LEVEL = 1 (Prioritäten, ab der MD11602 wirksam ist)

22.3.8 PLC-Anwenderprogramm

Anforderungen

Zur Ablaufkoordination von RESU ist im PLC-Anwenderprogramm folgende Funktionalität vorzusehen:

```
IF      DB21, ... DBX32.2      "Wiederaufsetzen aktiv" == 1
THEN   DB21, ... DBX0.1      "Vorwärts/Rückwärts" = 0
       DB21, ... DBX0.2      "Wiederaufsetzen starten" = 0
```

```
IF      DB11, ... DBX0.7      "BAG-Reset" == 1
OR      DB21, ... DBX7.7      "Reset" == 1
THEN   DB21, ... DBX0.1      "Vorwärts/Rückwärts" = 0
       DB21, ... DBX0.2      "Wiederaufsetzen starten" = 0
```

Zur Sicherheit sollten folgende Signale zurückgesetzt werden:

```
IF      DB21, ... DBX0.2      "Wiederaufsetzen starten" == 1
THEN   DB21, ... DBX0.1      "Vorwärts/Rückwärts" = 0

IF      DB21, ... DBX0.1      "Vorwärts/Rückwärts" == 1
THEN   DB21, ... DBX0.2      "Wiederaufsetzen starten" = 0
```

Programmbeispiel

Der folgende Programmausschnitt realisiert die oben beschriebenen Anforderungen:

```
U      DB21, ... DBX32.2      // IF      "Wiederaufsetzen aktiv" == 1
R      DB21, ... DBX0.1      // THEN   "Vorwärts/Rückwärts" = 0
R      DB21, ... DBX0.2      //        "Wiederaufsetzen starten" = 0
```

```

O   DB11, ... DBX0.7           // IF      "BAG-Reset" == 1
O   DB21, ... DBX7.7           // OR      "Reset" == 1
R   DB21, ... DBX0.1           // THEN   "Vorwärts/Rückwärts" = 0
R   DB21, ... DBX0.2           //        "Wiederaufsetzen starten" = 0
U   DB21, ... DBX0.2           // IF      "Wiederaufsetzen starten" == 1
R   DB21, ... DBX0.1           // THEN   "Vorwärts/Rückwärts" = 0
U   DB21, ... DBX0.1           // IF      "Vorwärts/Rückwärts" == 1
R   DB21, ... DBX0.2           // THEN   "Wiederaufsetzen starten" = 0

```

22.4 Programmierung

22.4.1 RESU-Start/Stop/Reset (CC_PREPRE)

Über die Prozedur `CC_PREPRE` (**Prepare Retrace**) können für das Wiederaufsetzen die Modi "Starten", "Stoppen" und "Rücksetzen" aktiviert werden:

Syntax

```
CC_PREPRE (<Modus>)
```

Bedeutung

CC_PREPRE:	Wiederaufsetz-Modus	
	Alleine im Satz:	ja

22.5 RESU-spezifische Teileprogramme

<Modus>:	Mode	
	Datentyp:	INT
	Wertebereich:	-1, 0, 1
	Wert	Bedeutung
	-1	<p>Startet die Protokollierung der Verfahrssätze.</p> <p>Die zum Rückwärtsfahren benötigten Informationen werden satzspezifisch in einem RESU-internen Satzpuffer protokolliert. Die Verfahrinformationen beziehen sich dabei auf die beiden Geometriachsen der RESU-Arbeitsebene, z. B. die 1. und 2. Geometrieachse des Kanals:</p> <p>MD20050 \$MC_AXCONF_GEOAX_ASSIGN_TAB[x]; mit x = 0 und 1</p> <p>Bzw. bei aktiver Transformation:</p> <p>MD24120 \$MC_TRAFO_GEOAX_ASSIGN_TAB[x]; mit x = 0 und 1</p>
0	<p>Stoppt die Protokollierung der Verfahrssätze.</p> <p>Nicht relevante Konturbereiche können somit von der Protokollierung ausgenommen werden.</p> <p>Derart ausgesparte Konturbereiche werden beim Rückwärtsfahren durch eine Gerade zwischen Anfangs- und Endpunkt überbrückt.</p>	
1	<p>Schaltet die Protokollierung der Verfahrssätze aus und löscht den funktionsinternen Satzpuffer. Vor dem Ausschaltzeitpunkt liegende Konturbereiche des Teileprogramms stehen für RESU damit nicht mehr zur Verfügung.</p>	

RESET-Verhalten

Bei folgenden Reset-Ereignissen wird implizit `CC_PREPRE (-1)` ausgeführt:

- Power On (Warmstart)
- Kanal-Reset
- Programmende-Reset (M30)

22.5 RESU-spezifische Teileprogramme

22.5.1 Übersicht

Im Rahmen des Wiederaufsetzens werden folgende Programme automatisch generiert. In den gekennzeichneten Programmen können anwenderspezifische Inhalte eingebracht werden.

Funktion	Programmname	Änderbar
Hauptprogramm	CC_RESU.MPF	Nein
INI-Programm	CC_RESU_INI.SPF	Ja
END-Programm	CC_RESU_END.SPF	Ja

Funktion	Programmname	Änderbar
Wiederaufsetz-ASUP	CC_RESU_BS_ASUP.SPF	Ja
RESU-ASUP	CC_RESU_ASUP.SPF	Nein

Interner Aufbau

Das folgende Bild gibt einen Überblick über den internen Aufbau der Technologie-Funktion und den Zusammenhang der verschiedenen Teileprogramme.

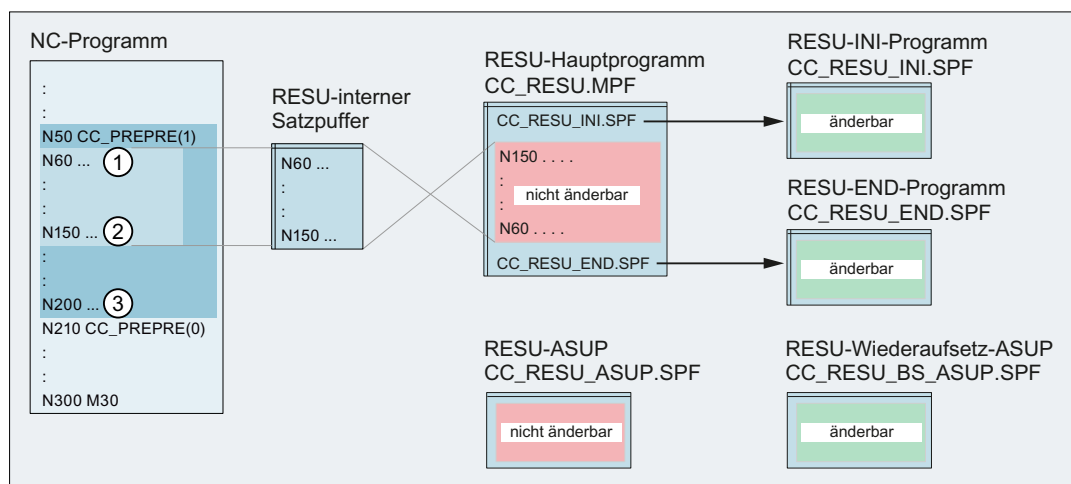


Bild 22-6 RESU-spezifische Teileprogramme

22.5.2 Hauptprogramm (CC_RESU.MPF)

Funktion

Das RESU-Hauptprogramm "CC_RESU.MPF" enthält neben den Aufrufen der RESU-spezifischen Unterprogramme die aus den protokollierten Verfahrssätzen des Satzpuffers erzeugten Verfahrssätze zum Rückwärts- / Vorwärtsfahren auf der Kontur. Es wird von RESU automatisch immer dann neu erzeugt, wenn nach Unterbrechung des Teileprogramms der Status des folgenden Nahtstellensignals wechselt:

DB21, ... DBX0.1 (Rückwärts / Vorwärts)

Hinweis

CC_RESU.MPF darf nicht verändert werden. Anwenderspezifische Anpassungen sind im entsprechenden RESU-spezifischen Unterprogramm durchzuführen.

Fehlermeldungen

Standardmäßig generiert RESU Verfahrssätze für den gesamten im Satzpuffer protokollierten RESU-fähigen Konturbereich. Ist zur Generierung aller Verfahrssätze im parametrisierten Speicherbereich des RESU-Hauptprogramms (siehe Kapitel "Speicherbereich des RESU-Hauptprogramms (Seite 854)") zuwenig Speicherplatz vorhanden, reduziert RESU die Anzahl der generierten Verfahrssätze.

Der fehlende Speicher bzw. die Reduzierung der generierten Verfahrssätze wird mit einem Alarm angezeigt:

RESU-Alarm 75608 "Kanal *Nummer* NC-Speichergrenze erreicht, RAM-Typ *Typ*"

Wird das RESU-Hauptprogramm im statischen Anwenderspeicher angelegt, wird gleichzeitig mit dem oben genannten RESU-Alarm folgender System-Alarm angezeigt:

Alarm 6500 "NC-Speichergrenze erreicht"

Hinweis

Wird aufgrund fehlenden Speichers die Anzahl der generierten Verfahrssätze reduziert, kann dennoch die gesamte RESU-fähige Kontur zum Wiederaufsetzen abgefahren werden. Dazu ist folgendes Vorgehen notwendig:

- Rückwärtsfahren bis zum Ende des RESU-Hauptprogramms
- Zweimaliger Wechsel des Nahtstellensignals:
DB21, ... DBX0.1 (Rückwärts / Vorwärts)

RESU generiert dadurch, ausgehend von der aktuellen Position als neuer Unterbrechungspunkt, ein neues RESU-Hauptprogramm.

Anschließend kann bis zum Ende des RESU-fähigen Konturbereichs oder bei erneuter Begrenzung bis zum Anfangspunkt des letzten generierbaren Verfahrssatzes gefahren werden.

Das beschriebene Vorgehen ist sowohl zum Rückwärts- als auch zum Vorwärtsfahren beliebig oft wiederholbar.

22.5.3 INI-Programm (CC_RESU_INI.SPF)

Funktion

Das RESU-spezifische Unterprogramm "CC_RESU_INI.SPF" enthält die für das Rückwärtsfahren notwendigen Voreinstellungen:

- Metrisches Eingabesystem: G71
- Maßangaben absolut: G90
- Ausschalten der einstellbaren Nullpunktverschiebungen / Frames (siehe Kapitel "Frames (Seite 873)": G500
- Ausschalten der aktiven Werkzeugkorrekturen (siehe T0 Kapitel "Werkzeugkorrekturen (Seite 873)":

- Ausschalten der Werkzeugradiuskorrektur: G40
- Verfahrgeschwindigkeit: F200

Programmstruktur

CC_RESU_INI.SPF hat folgenden voreingestellten Inhalt:

```

PROC CC_RESU_INI
  G71 G90 G500 T0 G40 F200
  ;vorhandene Systemframes werden deaktiviert
  ;Ist-Wert und Ankratzen
  if $MC_MM_SYSTEM_FRAME_MASK B_AND 'H01'
    $P_SETFRAME = ctrans()
  endif
  ;externe Nullpunktverschiebung
  if $MC_MM_SYSTEM_FRAME_MASK B_AND 'H02'
    $P_EXTFRAME = ctrans()
  endif
  ;Werkzeugtraeger
  if $MC_MM_SYSTEM_FRAME_MASK B_AND 'H04'
    PAROTOF
  endif
  if $MC_MM_SYSTEM_FRAME_MASK B_AND 'H08'
    TOROTOF
  endif
  ;Werkstueckbezugspunkte
  if $MC_MM_SYSTEM_FRAME_MASK B_AND 'H10'
    $P_WPFRAME = ctrans()
  endif
  ;Zyklen
  if $MC_MM_SYSTEM_FRAME_MASK B_AND 'H20'
    $P_CYCFRAME = ctrans()
  endif
  ;Transformationen
  if $MC_MM_SYSTEM_FRAME_MASK B_AND 'H40'
    $P_TRAFRAME = ctrans()
  endif
  ; Bitmaske fuer globale Basisframes
  $P_NCBFRMASK = 0
  ;Bitmaske für kanalspezifische Basisframes
  $P_CHBFRMASK = 0
  ;programmierbare Frame
  $P_PFRAME = ctrans()

```

M17

 **VORSICHT**

Programmänderungen

Mit der Veränderung des Inhalts des RESU-spezifischen Unterprogramms "CC_RESU_INI.SPF" übernimmt der Anwender (Maschinenhersteller) die Verantwortung für den korrekten Ablauf der Technologie-Funktion.

Hinweis

CC_RESU_INI.SPF darf verändert werden.

CC_RESU_INI.SPF darf keine RESU-Teileprogrammbefehle CC__PREPRE (x) enthalten.

22.5.4 END-Programm (CC_RESU_END.SPF)

Funktion

Das RESU-spezifische Unterprogramm "CC_RESU_END.SPF" hat die Aufgabe, das Rückwärtsfahren anzuhalten, wenn das Ende der RESU-fähigen Kontur erreicht wurde. Bei geeigneter Parametrierung von RESU wird dieser Fall in der Regel nicht eintreten.

Programmstruktur

CC_RESU_END.SPF hat folgenden voreingestellten Inhalt:

```
PROC CC_RESU_END
      M0
M17
```

 **VORSICHT**

Programmänderungen

Mit der Veränderung des Inhalts des RESU-spezifischen Unterprogramms "CC_RESU_END.SPF" übernimmt der Anwender (Maschinenhersteller) die Verantwortung für den korrekten Ablauf der Technologie-Funktion.

Hinweis

CC_RESU_END.SPF darf verändert werden.

CC_RESU_END.SPF darf keine RESU-Teileprogrammbefehle CC__PREPRE (x) enthalten.

22.5.5 Wiederaufsetz-ASUP (CC_RESU_BS_ASUP.SPF)

Funktion

Mittels des RESU-spezifischen ASUP "CC_RESU_BS_ASUP.SPF" wird die NC veranlasst, beim Wiederaufsetzen an den aktuellen Bahnpunkt anzufahren:

- Wiederanfahren an den nächstliegenden Bahnpunkt: RMN
- Anfahren auf einer Geraden mit allen Achsen: REPOSA

Programmstruktur

CC_RESU_BS_ASUP.SPF hat folgenden voreingestellten Inhalt:

```
PROC CC_RESU_BS_ASUP SAVE
    RMN
    REPOSA
M17
```

VORSICHT

Programmänderungen

Mit der Veränderung des Inhalts des RESU-spezifischen Unterprogramms "CC_RESU_BS_ASUP.SPF" übernimmt der Anwender (Maschinenhersteller) die Verantwortung für den korrekten Ablauf der Technologie-Funktion.

Hinweis

CC_RESU_BS_ASUP.SPF darf verändert werden.

Anwenderspezifische Anpassungen sind vor dem Teileprogrammsatz RMN einzufügen.

22.5.6 RESU-ASUP (CC_RESU_ASUP.SPF)

Funktion

Das RESU-spezifische ASUP "CC_RESU_ASUP.SPF" wird funktionsintern benötigt. Das ASUP wird ausgelöst, wenn im NC-Stop-Zustand das folgende RESU-Nahtstellensignal umgeschaltet wird:

DB21, ... DBX0.1 (Vorwärts / Rückwärts)

Programmstruktur

CC_RESU_ASUP.SPF hat folgenden Inhalt:

PROC CC_RESU_ASUP	
	; siemens system asup - do not change G4 F0.001 M0 REPOSA
M17	

Hinweis

CC_RESU_ASUP.SPF darf nicht verändert werden.

22.6 Wiederaufsetzen

22.6.1 Allgemeine Informationen

Wiederaufsetzen bezeichnet den gesamten Vorgang vom Auslösen des Wiederaufsetzens über das Nahtstellensignal DB21, ... DBX0.2 = 1 (Wiederaufsetzen starten) bis zum Fortsetzen der Teileprogrammbearbeitung auf der programmierten Kontur.

Voraussetzung

Als Voraussetzung für das Wiederaufsetzen muss der Retrace Mode, ausgelöst durch die Anforderung zum Rückwärtsfahren, im Kanal aktiv sein:

DB21, ... DBX32.1 = 1 (Retrace Mode aktiv)

(Siehe Kapitel "Funktionsablauf (Prinzip) (Seite 847)")

Teilfunktionen

Die beiden wesentlichen Teilfunktionen des Wiederaufsetzens sind die Standard-NC-Funktionen:

- Satzsuchlauf mit Berechnung an der Kontur
- Repositionieren an die Kontur auf kürzestem Weg (REPOS RMN)

22.6.2 Satzsuchlauf mit Berechnung an der Kontur

Funktion

Der im Rahmen des Wiederaufsetzens implizit von RESU ausgelöste Satzsuchlauf mit Berechnung an der Kontur hat folgende Aufgaben:

- Programmzeiger auf den Teileprogrammsatz stellen, auf den mittels Rückwärts- / Vorwärtsfahren zurückpositioniert wurde.
- Berechnung der Achspositionen aufgrund der programmierten Verfahrssätze vom Teileprogrammanfang bis zum Zielsatz.
- Aufsammeln der vom Teileprogrammanfang bis zum Zielsatz programmierten Anweisungen, die im Aktionssatz ausgeführt werden, wie z. B.:
 - Hilfsfunktionen
 - Werkzeugwechsel
 - Spindelfunktionen
 - Vorschubprogrammierung

Alle Teileprogrammanweisungen, die nicht im Aktionssatz ausgeführt werden, aber zum Wiederaufsetzen im Teileprogramm benötigt werden, müssen manuell in das RESU-spezifische Wiederaufsetz-ASUP "CC_RESU_BS_ASUP.SPF" eingetragen werden, z. B.:

- Synchronaktionen
- M-Funktionen

Literatur

Die vollständige Beschreibung der Funktion "Satzsuchlauf" findet sich in:
Funktionshandbuch Grundfunktionen; BAG, Kanal, Programmbetrieb (K1), Programmtest

22.6.3 Repositionieren

Funktion

Nach dem Ende des letzten Aktionssatzes (letzter Verfahrssatz vor dem Repositionieren) erfolgt mit NC-Start die Ausgabe des Anfahrsatzes zum Repositionieren aller im Teileprogramm bis zum Zielsatz programmierten Kanalachsen.

Geometrieachsen

Im Anfahrtsatz verfahren die Geometrieachsen der RESU-Arbeitsebene (z. B. die 1. und 2. Geometrieachse des Kanals) auf kürzestem Weg zum Wiederaufsetzpunkt an die Kontur.

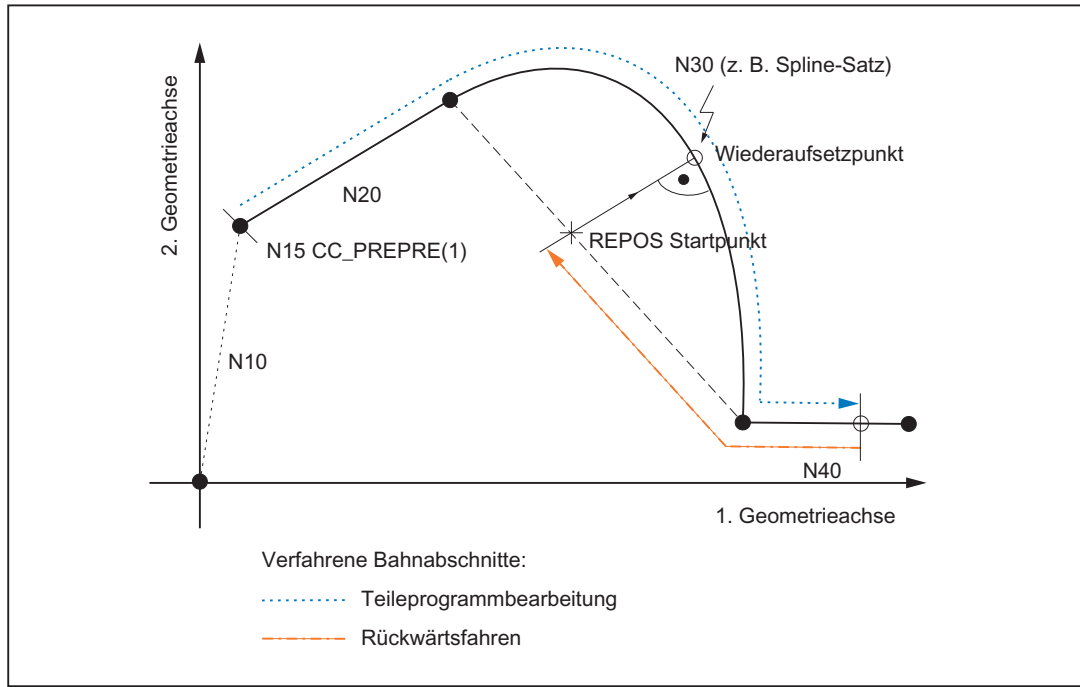


Bild 22-7 RESU-fähige Konturbereiche und REPOS

Kanalachsen

Alle anderen im Teileprogramm programmierten Kanalachsen verfahren auf ihre jeweilige im Satzsuchlauf berechnete Position.

22.6.4 Zeitliche Bedingungen bezüglich NC-Start

Im Rahmen des Wiederaufsetzens ist vom Maschinenbediener zweimal NC-Start auszulösen (siehe Kapitel "Funktionsablauf (Prinzip) (Seite 847)").

Dabei sind folgende Bedingungen zu beachten:

- beim NC-Start zum Ausgeben der Aktionssätze:
 - Der Satzsuchlauf muss abgeschlossen sein:
DB21, ... DBX33.4 = 0 (Satzsuchlauf aktiv)
- beim NC-Start zum Ausgeben des Anfahrtsatzes:
 - Das RESU-ASUP "CC_RESU_BS_ASUP" muss beendet sein:
DB21, ... DBX318.0 = 1 (ASUP angehalten)

22.6.5 Satzsuchlauf ab letztem Hauptsatz

Der im Rahmen des Wiederaufsetzens durchgeführte Satzsuchlauf mit Berechnung an der Kontur kann selbst bei Verwendung der leistungsfähigsten NCU bei sehr großen Teileprogrammen zu Rechenzeiten von mehreren Minuten bis zum Erreichen des Zielsatzes führen.

Eine signifikante Verkürzung dieser Wartezeit ist möglich durch die Verwendung der Funktion "Satzsuchlauf ab dem letzten Hauptsatz".

Funktionalität

Beim Wiederaufsetzen mit Satzsuchlauf ab dem letzten Hauptsatz erfolgt das Auffinden des Zielsatzes in 2 Schritten:

1. Satzsuchlauf ohne Berechnung vom Anfang des Bearbeitungsprogramms bis zum letzten Hauptsatz vor dem Zielsatz. Während dieses Satzsuchlaufs werden keine Unterprogramme berücksichtigt. Diese Suche erfolgt somit ausschließlich im Hauptprogramm.
2. Satzsuchlauf mit Berechnung an der Kontur ab dem Hauptsatz bis zum Zielsatz. Dieser Satzsuchlauf berücksichtigt auch Unterprogramme.

Voraussetzung

Voraussetzung dafür, dass zum Wiederaufsetzen ein Satzsuchlauf ab dem letzten Hauptsatz verwendet werden kann, ist die Programmierung mindestens eines Hauptsatzes nach dem RESU-Start `CC_PREPRE (1)`.

Hauptsatz

In einem Hauptsatz müssen alle Anweisungen programmiert sein, die zum Abarbeiten des nachfolgenden Teileprogrammabschnitts erforderlich sind.

Hauptsätze sind durch eine Hauptsatznummer zu kennzeichnen, die aus dem Zeichen ":" und einer positiven ganzen Zahl (Satznummer) besteht.

Literatur:

Programmierhandbuch Grundlagen; Grundlagen der NC-Programmierung, Sprachelemente der Programmiersprache

Aktivierung

Das Aktivieren des Satzsuchlaufs ab dem letzten Hauptsatz erfolgt über das RESU-spezifische Maschinendatum:

MD62575 \$MC_RESU_SPECIAL_FEATURE_MASK_2, Bit 0 (Zusätzlich RESU-Eigenschaften)

Bit	Wert	Bedeutung
0	0	Das Wiederaufsetzen erfolgt mittels Satzsuchlauf mit Berechnung an der Kontur.
	1	Das Wiederaufsetzen erfolgt mittels Satzsuchlauf ab dem letzten Hauptsatz.

Randbedingungen

Damit nach einem Wiederaufsetzen mit Satzsuchlauf ab dem letzten Hauptsatz ein erneutes Wiederaufsetzen erfolgen kann, muss der RESU-Start `CC_PREPRE(1)` im Wiederaufsetz-ASUP "CC_RESU_BS_ASUP.SPF" programmiert sein.

Programmierbeispiel:

```
PROC CC_RESU_BS_ASUP SAVE
    ; (zum Wiederaufsetzen benötigte
    ; Synchronaktionen, M-Funktionen, etc.)
    CC_PREPRE(1)
    RMN
    REPOSA
M17
```

22.7 Funktionsspezifische Anzeigedaten

22.7.1 Kanalspezifische GUD-Variable

Als Anzeigedatum für die Größe des Satzsuchlaufpuffers stellt RESU folgende kanalspezifische GUD-Variable zur Verfügung:

GUD-Variable	Bedeutung	Einheit	Zugriff
CLC_RESU_LENGTH_BS_BUFFER	Größe des Satzsuchlaufpuffers	Byte	nur lesen

Nach erfolgter Inbetriebnahme der Technologie-Funktion wird die GUD-Variable nicht automatisch auf der Bedienoberfläche angezeigt.

Anlegen und Anzeigen der GUD-Variable

Zum Anlegen und Anzeigen der GUD-Variablen sind auf der Bedienoberfläche folgende Handlungsschritte durchzuführen:

1. Kennwort setzen:
Es ist das Kennwort der Schutzstufe 1 (Maschinenhersteller) einzugeben.
2. Anzeige der "Definitionen" aktivieren.
3. Falls noch keine Datei "SGUD.DEF" vorhanden ist, wird die Datei neu angelegt:
Name: SGUD
Typ: GlobaleDaten / System
Eingaben mit **OK** bestätigen.
Die Datei wird daraufhin im Editor geöffnet.
4. GUD-Variablen-Definitionen editieren:
DEF CHAN REAL CLC_RESU_LENGTH_BS_BUFFER
M30

5. Datei speichern und Editor schließen.

6. Datei "SGUD.DEF" aktivieren.

Die GUD-Variable wird jetzt auf der Bedienoberfläche angezeigt.

Hinweis

Die neu angelegte und bereits angezeigte GUD-Variable wird von RESU erst nach einem NC-POWER ON-Reset erkannt und mit dem aktuellen Wert versorgt. Nach dem Anlegen muss daher ein NC-POWER ON-Reset ausgelöst werden.

Literatur

Das genaue Vorgehen beim Anlegen und Anzeigen von GUD-Variablen ist abhängig von der Software-Version der vorhandenen Bedienoberfläche und beschrieben im:

- Bedienhandbuch

22.8 Funktionsspezifische Alarmtexte

Zum Vorgehen beim Anlegen von funktionsspezifischen Alarmtexten siehe Kapitel "Anlegen von Alarmtexten (Seite 696)".

22.9 Randbedingungen

22.9.1 Funktionsspezifische Randbedingungen

22.9.1.1 Wiederaufsetzen innerhalb von Unterprogrammen

Unterprogrammaufruf außerhalb oder innerhalb einer Programmschleife

Ein eindeutiges Wiederaufsetzen innerhalb von Unterprogrammen ist davon abhängig, ob der Unterprogrammaufruf außerhalb oder innerhalb einer Programmschleife erfolgt:

- Außerhalb
Wird ein Unterprogramm außerhalb einer Programmschleife aufgerufen, ist ein eindeutiges Wiederaufsetzen möglich.
- Innerhalb
Wird ein Unterprogramm innerhalb einer Programmschleife aufgerufen, ist unter Umständen kein eindeutiges Wiederaufsetzen möglich (siehe Kapitel "Wiederaufsetzen innerhalb von Programmschleifen (Seite 870)").

Durchlaufzahl P

Unterprogramm-Wiederholungen durch Verwendung der Durchlaufzahl P werden beim Wiederaufsetzen berücksichtigt. D. h. das Wiederaufsetzen im Teileprogramm erfolgt mit dem korrekten Bezug von Teileprogrammsatz und Durchlaufzahl P zum Wiederaufsetzpunkt der Kontur.

22.9.1.2 Wiederaufsetzen innerhalb von Programmschleifen

Innerhalb der NC-Hochsprache sind Programmschleifen programmierbar mittels:

- LOOP ENDLOOP
- FOR ENDFOR
- WHILE ENDWHILE
- REPEAT UNTIL
- CASE/IF-ELSE-ENDIF in Zusammenhang mit GOTOB

WARNUNG

Kollisionsgefahr

Wenn der Wiederaufsetzpunkt an der programmierten Kontur das Ergebnis eines Schleifendurchlaufs ungleich dem ersten Schleifendurchlauf ist, können sich im weiteren Verlauf der Bearbeitung erhebliche Konturabweichungen ergeben.

Bei einem Wiederaufsetzen innerhalb von Programmschleifen erfolgt das Wiederaufsetzen immer im ersten Schleifendurchlauf.

22.9.1.3 Wiederaufsetzen an Vollkreisen

Bei Vollkreisen fallen Satzanfangs- und -endpunkt in einem Konturpunkt zusammen. Da in diesem Fall keine eindeutige Unterscheidung möglich ist, wird beim Wiederaufsetzen an einem derartigen Konturpunkt immer vom Satzanfangspunkt ausgegangen. Der erste Teileprogrammsatz nach Wiederaufsetzen ist dann der Kreissatz.

Um zu vermeiden, dass nach dem Wiederaufsetzen der Kreissatz gefahren wird, ist als Wiederaufsetzpunkt ein Konturpunkt kurz vor dem Satzendpunkt des Kreissatzes zu wählen.

22.9.1.4 Automatisch generierte Konturelemente

Die automatische Generierung von nicht-linearen / nicht-kreisförmigen Konturelementen durch die NC erfolgt z. B. bei Programmierung folgender NC-Funktionen im Teileprogramm:

- RND
- G641/G642
- Werkzeugradiuskorrektur

Für das Rückwärts- / Vorwärtsfahren im Rahmen von RESU werden diese Konturelemente durch Geraden zwischen Satzanfangs- und Satzendpunkt ersetzt.

22.9.2 Randbedingungen bezüglich Standardfunktionen

22.9.2.1 Achstausch

Solange RESU aktiv ist, dürfen die beiden Geometrieachsen der RESU-Arbeitsebene (z. B. 1. und 2. Geometrieachse des Kanals) nicht per Achstausch (`RELEASE (x) /GET (x)`) an einen anderen Kanal übergeben werden.

Die RESU-Aktivität:

- beginnt:
 - mit dem Teileprogrammbefehl `CC_PREPRE (1)`
- endet mit:
 - dem Programmende
oder
 - dem Teileprogrammbefehl `CC_PREPRE (-1)`

22.9.2.2 Verfahrbewegungen von Kanalachsen

Andere Kanalachsen außer den beiden Geometrieachsen der RESU-Arbeitsebene werden durch RESU nicht betrachtet.

Sind zum Wiederaufsetzen bzw. Rückwärtsfahren Verfahrbewegungen in anderen Kanalachsen erforderlich, so können diese entweder vom Maschinenbediener von Hand vorgenommen werden, oder im RESU-spezifischen Unterprogramm "CC_RESU_INI.SPF" als Verfahrersatz programmiert werden.

 WARNUNG
Kollisionsgefahr bei Verfahrbewegung
Während des gesamten Wiederaufsetzvorgangs im Rahmen der Technologie-Funktion RESU ist vom Maschinenbediener die Kollisionsfreiheit der damit verbundenen Verfahrbewegungen sicherzustellen.

22.9.2.3 Satznummern

Die folgenden RESU-spezifischen Unterprogramme und deren Unterprogramme dürfen keine Satznummern enthalten:

- `CC_RESU_INI.SPF`
- `CC_RESU_END.SPF`

Im Fehlerfall wird folgender Alarm angezeigt:

Alarm 75604 "Rückwärtsfahren nicht möglich, Fehler-Nr. *Nummer*"

22.9.2.4 Satzsuchlauf

Satzsuchlauf mit Berechnung

Im Rahmen der Standardfunktion "Satzsuchlauf mit Berechnung (an der Kontur / am Satzende)" gelten bezüglich RESU folgende Randbedingungen:

- Im Zielsatz ist der letzte während des Satzsuchlaufs durchlaufene RESU-Teileprogrammbefehl `CC_PREPRE (x)` wirksam.
- Der RESU-fähige Konturbereich beginnt mit dem `REPOS`-Anfahrtsatz.

Satzsuchlauf ohne Berechnung

Bei der Funktion "Satzsuchlauf ohne Berechnung" werden die RESU-Teileprogrammbefehle `CC_PREPRE (x)` **nicht** wirksam.

22.9.2.5 Transformationen

RESU kann mit Einschränkungen auch bei aktiver kinematischer Transformation (z. B. 5-Achstransformation) verwendet werden, da die Verfahrbewegungen der beiden Geometrieachsen der RESU-Arbeitsebene im Basis-Koordinatensystem (BKS) und somit vor der Transformation protokolliert werden (siehe auch Kapitel "TE4: Transformationspaket Handling - nur 840D sl (Seite 775)").

Transformationswechsel

Während RESU aktiv ist, darf kein Wechsel der Transformation bzw. kein Ein- / Ausschalten der Transformation erfolgen.

Die RESU-Aktivität:

- beginnt:
 - mit dem Teileprogrammbefehl `CC_PREPRE (1)`
- endet mit:
 - dem Programmende
oder
 - dem Teileprogrammbefehl `CC_PREPRE (-1)`

Literatur

Eine vollständige Beschreibung der Transformationen findet sich in:
Funktionshandbuch Erweiterungsfunktionen, Kinematische Transformation (M1)

22.9.2.6 Kompensationen

RESU kann im Zusammenhang mit Kompensationen verwendet werden, da die Verfahrbewegungen der beiden Geometrieachsen der RESU-Arbeitsebene im Basis-Koordinatensystem (BKS) und somit vor der Kompensation protokolliert werden.

Eine vollständige Beschreibung der Kompensationen findet sich in:

Literatur:

Funktionshandbuch Erweiterungsfunktionen; Kompensationen (K3)

22.9.2.7 Frames

RESU kann im Zusammenhang mit Frames verwendet werden.

Da die Verfahrbewegungen der beiden Geometrieachsen der RESU-Arbeitsebene aber im Basis-Koordinatensystem (BKS) und somit nach Einrechnung der Frames protokolliert werden, müssen während des Wiederaufsetzens (Rückwärts- / Vorwärtsfahren) die Frame-Korrekturen ausgeschaltet sein.

Das Ausschalten der Frame-Korrekturen während des Wiederaufsetzens erfolgt durch die standardmäßigen Voreinstellungen des RESU-spezifischen Unterprogramms "CC_RESU_INI.SPF" (siehe Kapitel "INI-Programm (CC_RESU_INI.SPF) (Seite 860)").

Eine vollständige Beschreibung der Frames findet sich in:

Literatur:

Funktionshandbuch Grundfunktionen; Achsen, Koordinatensysteme, Frames (K2)

22.9.2.8 Werkzeugkorrekturen

RESU kann im Zusammenhang mit Werkzeugkorrekturen verwendet werden.

Da die Verfahrbewegungen der beiden Geometrieachsen der RESU-Arbeitsebene aber im Basis-Koordinatensystem (BKS) und somit nach Einrechnung der Werkzeugkorrekturen protokolliert werden, müssen während des Wiederaufsetzens (Rückwärts- / Vorwärtsfahren) die Werkzeugkorrekturen ausgeschaltet sein.

Das Ausschalten der Werkzeugkorrekturen während des Wiederaufsetzens erfolgt durch die standardmäßigen Voreinstellungen des RESU-spezifischen Unterprogramms "CC_RESU_INI.SPF" (siehe Kapitel "INI-Programm (CC_RESU_INI.SPF) (Seite 860)").

In spezifischen Fällen der Werkzeugradiuskorrektur, z. B. Korrekturen an Außenecken $G450\text{DISC}=x$, kann es zu Konturabweichungen zwischen der während des Wiederaufsetzens gefahrenen Kontur und der im Bearbeitungsprogramm programmierten Kontur kommen.

Konturabweichungen ergeben sich immer dann, wenn die Werkzeugradiuskorrektur Konturelemente erzeugt, die nicht linear oder kreisförmig sind. Z. B. erzeugt $G450\text{DISC}=x$ mit $x > 0$ parabel- oder hyperbelförmige Konturelemente.

Eine vollständige Beschreibung der Werkzeugkorrekturen findet sich in:

Literatur:

Funktionshandbuch Grundfunktionen, Werkzeugkorrektur (W1)

22.10 Datenlisten

22.10.1 Maschinendaten

22.10.1.1 Allgemeine Maschinendaten

Nummer	Bezeichner: \$MN_	Beschreibung
11602	ASUP_START_MASK	Ignoriere Stoppgründe, wenn ein Asup läuft.
11604	ASUP_START_PRIO_LEVEL	Legt fest, ab welcher Asup-Priorität MD11602 wirksam ist.
18351	MM_DRAM_FILE_MEM_SIZE	Größe des Teileprogrammspeicher im DRAM (in kByte)

22.10.1.2 Kanal-spezifische Maschinendaten

Nummer	Bezeichner: \$MC_	Beschreibung
20050	AXCONF_GEOAX_ASSIGN_TAB	Zuordnung Geometrie - Kanalachse
24120	TRAFO_GEOAX_ASSIGN_TAB_1	Zuordnung Geometrie - Kanalachse für Transformati- on 1
28090	MM_NUM_CC_BLOCK_ELEMENTS	Anzahl Blockelemente für Compile-Zyklen (CC)
28100	MM_NUM_CC_BLOCK_USER_MEM	Größe Satzspeicher für CC in kByte
28105	MM_NUM_CC_HEAP_MEM	Heap-Speicher in kByte für CC-Applikationen (DRAM)
62571	RESU_RING_BUFFER_SIZE	Größe des Ringpuffers (RESU-interner Satzpuffer)
62572	RESU_SHARE_OF_CC_HEAP_MEM	RESU-Anteil am gesamten CC-Heap-Speicher
62573	RESU_INFO_SA_VAR_INDEX	Indizes der benutzten Synchronaktionsvariablen
62574	RESU_SPECIAL_FEATURE_MASK	Zusätzlich RESU-Eigenschaften
62575	RESU_SPECIAL_FEATURE_MASK_2	Zusätzlich RESU-Eigenschaften
62580	RESU_WORKING_PLANE	Anwahl der RESU-Arbeitsebene

22.10.2 Signale

22.10.2.1 Signale an Kanal

Signalname	SINUMERIK 840D sl	SINUMERIK 828D
RESU: Rückwärts / Vorwärts	DB21,DBX0.1	-
RESU: Wiederaufsetzen starten	DB21,DBX0.2	-

22.10.2.2 Signale von Kanal

Signalname	SINUMERIK 840D sl	SINUMERIK 828D
RESU: Retrace Mode aktiv	DB21,DBX32.1	-
RESU: Wiederaufsetzen aktiv	DB21,DBX32.2	-

TE8: Taktunabhängige bahnsynchrone Schaltsignalausgabe - nur 840D sl

23

23.1 Kurzbeschreibung

Funktion

Die Technologie-Funktion "Taktunabhängige bahnsynchrone Schaltsignalausgabe" dient dem schnellen Ein- und Ausschalten bei zeitkritischen, positionsabhängigen Bearbeitungsprozessen, z. B. dem Hochgeschwindigkeits-Laserschneiden (HSLC; High Speed Laser Cutting).

Die Schaltsignalausgabe kann satzbezogen oder bahnlangenbezogen erfolgen:

- **Satzbezogene Schaltsignalausgabe**
Die Schaltsignalausgabe und damit das Ein- / Ausschalten der Bearbeitung erfolgt abhängig von den Zustandsänderungen:
 - Eilgang G00 aktiv / inaktiv
 - Programmierte Vorschub-Schwelle unter- / überschritten
- **Bahnlangenbezogene Schaltsignalausgabe**
Die Schaltsignalausgabe und damit das Ein- / Ausschalten der Bearbeitung erfolgt in einem kontinuierlichen Wechsel und wird über die zurückgelegte Wegstrecke gesteuert. Dies ermöglicht eine regelmäßige Bearbeitung, ohne dass die einzelnen Schaltpositionen explizit programmiert werden müssen.

Die Aktivierung bzw. Auswahl, welche der genannten Möglichkeiten zur Steuerung der Ausgabe des digitalen Signals führt, ist per Teileprogrammbefehl programmierbar.

I / O-Peripherie

Als digitale I / O-Peripherie, über die das Schaltsignal ausgegeben wird, kann nur die Onboard-Peripherie der NC-Baugruppe verwendet werden. Das Schaltsignal kann nur über einen der 4 digitalen Onboard-Ausgänge der NCU-Baugruppe ausgegeben werden.

Einschränkungen

Für die Anwendung der Technologie-Funktion "Taktunabhängige bahnsynchrone Schaltsignalausgabe" gilt folgende Einschränkung:

- **Die Technologie-Funktion ist nur in einem Kanal der NC verfügbar.**

Literatur

Die Technologie-Funktion "Taktunabhängige bahnsynchrone Schaltsignalausgabe" ist ein Compile-Zyklus. Für die Handhabung von Compile-Zyklen (siehe Kapitel "TE01: Installation und Aktivierung ladbarer Compile-Zyklen (Seite 689)").

23.2 Funktionsbeschreibung

23.2.1 Allgemeine Informationen

Hinweis

Die Beschreibung der Funktionalität erfolgt beispielhaft anhand der Technologie "Hochgeschwindigkeits-Laserschneiden" (HSLC, High Speed Laser Cutting).

23.2.2 Ermittlung der Schaltpositionen

23.2.2.1 Satzbezogene Schaltsignalausgabe

Schaltkriterien

Beim Hochgeschwindigkeits-Laserschneiden, z. B. bei der Fertigung von Lochblechen, ist es unbedingt erforderlich, den Laserstrahl während des Bearbeitungsprozesses exakt an den programmierten Sollpositionen ein- bzw. auszuschalten.

Zur Minimierung des Programmieraufwandes werden die Schaltpositionen der Technologie-Funktion über die im Teileprogrammsatz programmierte Geschwindigkeit der Geometrieachsen ermittelt.

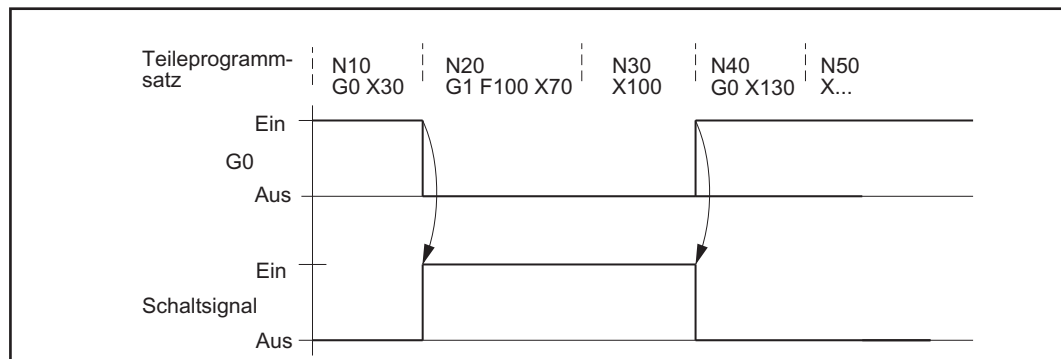
Folgende Kriterien definieren die im Teileprogrammsatz programmierte Sollposition (Satzendposition) als Schaltposition:

1. G0-Flankenwechsel
2. Über- bzw. Unterschreiten eines frei programmierbaren Geschwindigkeits-Schwellwerts

G0-Flankenwechsel als Schaltkriterium

Ist G0 (Eilgang) in einem Teileprogrammsatz **aktiv** (programmiert oder modal), wird das Schaltsignal **ausgeschaltet**. Umgekehrt gilt: ist G0 (Eilgang) in einem Teileprogrammsatz **nicht aktiv**, wird das Schaltsignal **eingeschaltet**. Der G0-Flankenwechsel markiert dabei die programmierte Satzendposition des vorhergehenden Satzes als Schaltposition.

Beispiel:



Folgende Satzendpositionen wirken als Schaltpositionen:

- Position X30 beim G0-Flankenwechsel von N10 nach N20
- Position X100 beim G0-Flankenwechsel von N30 nach N40

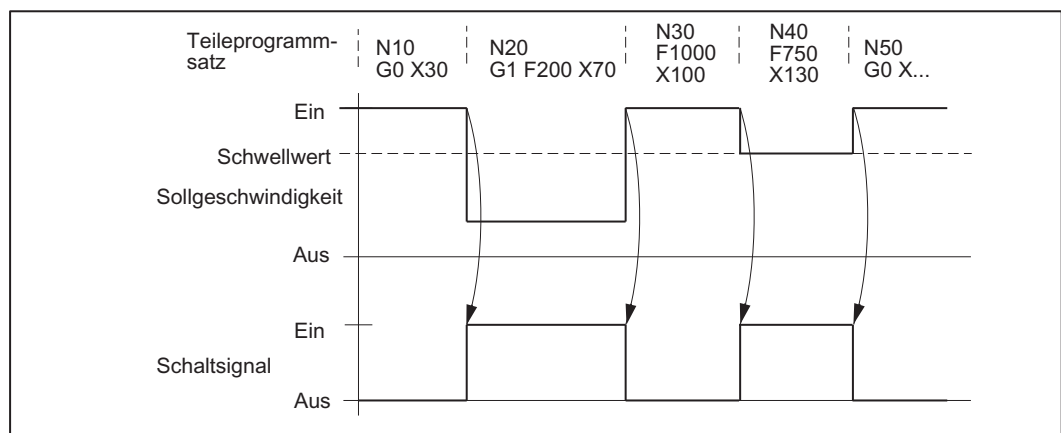
Frei programmierbarer Geschwindigkeits-Schwellwert als Schaltkriterium

Über einen frei programmierbaren Geschwindigkeits-Schwellwert wird definiert, ab welcher im Teileprogrammsatz programmierten Sollgeschwindigkeit das Schaltsignal ein- bzw. ausgeschaltet wird:

- Liegt die im Teileprogrammsatz programmierte Sollgeschwindigkeit **über** dem programmierten Schwellwert, wird das Schaltsignal **ausgeschaltet**.
- Liegt die Sollgeschwindigkeit **auf** bzw. **unter** dem Schwellwert, wird das Schaltsignal **eingeschaltet**.

Der Flankenwechsel markiert dabei die programmierte Satzendposition des vorhergehenden Satzes als Schaltposition.

Beispiel:



23.2 Funktionsbeschreibung

Folgende Satzendpositionen wirken als Schaltpositionen:

- Position X30 beim Flankenwechsel von N10 nach N20
- Position X70 beim Flankenwechsel von N20 nach N30

Hinweis

Durch G0 wird das Schaltsignal, unabhängig vom Schwellwert, immer ausgeschaltet.

23.2.2.2 Bahnlängenbezogene Schaltsignalausgabe

Programmierbare Wegstrecken als Schaltkriterium

Bei der bahnlängenbezogenen Schaltsignalausgabe werden die Schaltpositionen durch die beiden frei programmierbaren Wegstrecken s_1 und s_2 definiert.

Funktionsablauf

Die bahnlängenbezogene Schaltsignalausgabe beginnt mit einem Einschaltsignal am Anfang des ersten Verfahrssatzes nach der Aktivierung mit CC_FAST_CONT (siehe ① im Bild).

Die Bearbeitung ist aktiv, bis nach dem Verfahren einer programmierbaren Wegstrecke s_1 ein Ausschaltsignal gesetzt wird (siehe ② im Bild). Dadurch wird die Bearbeitung unterbrochen, bis nach dem Verfahren einer programmierbaren Wegstrecke s_2 erneut ein Einschaltsignal gesetzt wird (siehe ③ im Bild).

Der Wechsel zwischen Ein- und Ausschaltsignal und damit der Wechsel zwischen Bearbeitungs- und Unterbrechungsphase erfolgt also bahnlängenabhängig am Ende eines jeden Streckenabschnitts s_1 und s_2 . Dies ermöglicht eine kontinuierliche und regelmäßige Bearbeitung, ohne dass die einzelnen Schaltpositionen explizit programmiert werden müssen.

Die bahnlängenbezogene Schaltsignalausgabe endet mit Beginn des ersten Verfahrssatzes (oder eines anderen ausführbaren Satzes) nach der Deaktivierung mit CC_FASTOFF (siehe ④ im Bild).

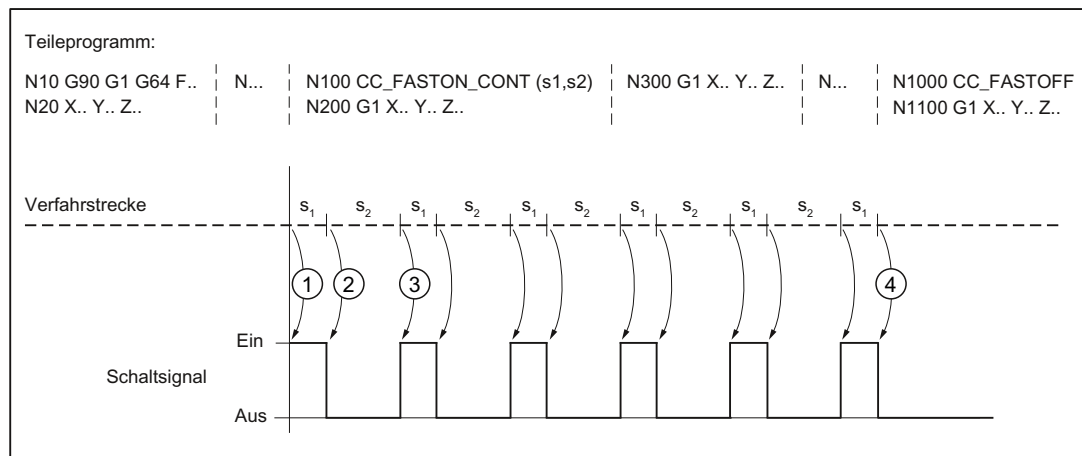


Bild 23-1 Bahnlängenbezogene Schaltsignalausgabe

23.2.3 Berechnung der Schaltzeitpunkte

Um ein möglichst exaktes Schalten an den ermittelten Schaltpositionen zu erzielen, berechnet die Steuerung in jedem Lagereglertakt die Wegdifferenz zwischen der Istposition der beteiligten Geometrieachsen und der Schaltposition.

Wird die Wegdifferenz kleiner 1,5 Lagereglertakte, rechnet die Steuerung sie unter Berücksichtigung der aktuellen Bahngeschwindigkeit und -beschleunigung der Geometrieachsen in eine Zeitdifferenz um.

Mit der Zeitdifferenz als Vorgabe wird ein Hardware-Timer gestartet, der dann, unabhängig vom Lagereglertakt, exakt zum vorausberechneten Zeitpunkt das Schaltsignal triggert.

23.2.4 Schaltfrequenz und Schaltpositionsabstand

Maximale Schaltfrequenz

Die maximale Schaltfrequenz beträgt: 1 Signalflankenwechsel pro IPO-Takt

Hinweis

Sonderfall: IPO-Taktzeit = Lageregler-Taktzeit

In diesem Fall beträgt die maximale Schaltfrequenz:

1 Signalflankenwechsel pro **2** IPO-Takte

Minimaler Schaltpositionsabstand

Der minimal mögliche Abstand zwischen aufeinander folgenden Schaltpositionen ist abhängig von:

- der Dauer eines IPO-Takts
- der Vorschubgeschwindigkeit

Aus diesen Größen lässt sich der theoretisch mögliche Minimalabstand wie folgt ermitteln:

$\text{Minimaler Schaltpositionsabstand} = \text{Programmierte Vorschubgeschwindigkeit} * \text{IPO-Taktzeit}$
--

Beispiel:

Bei IPO-Taktzeiten von 2 ms und Lageregler-Taktzeiten von 1 ms sowie einer Vorschubgeschwindigkeit von 20000 mm/min wird der theoretisch mögliche Minimalabstand zwischen aufeinander folgenden Schaltpositionen begrenzt auf:

$$20000 \text{ mm/min} * 2 \text{ ms} = 0,667 \text{ mm}$$

Unterschreiten des minimalen Schaltpositionsabstandes

Bei der bahnlängenbezogenen Schaltsignalausgabe kann es zu einem Unterschreiten des minimalen Schaltpositionsabstandes kommen, z. B. durch:

- Erhöhung der Vorschubgeschwindigkeit
- Verringerung der programmierbaren Schaltpositionsabstände s_1 und s_2

Ein Unterschreiten hat folgende Reaktionen zur Folge:

- Alarm 75501 "Kanal %1 HSLC: CC_FASTON_CONT Geschwindigkeit zu hoch" wird angezeigt.
- Das Schaltsignal an der aktuellen Schaltposition wird ausgelassen. Damit der Bearbeitungsrythmus erhalten bleibt, unterdrückt die Funktion auch das Schaltsignal an der darauf folgenden Schaltposition.
Die Lage aller nachfolgenden Schaltpositionen wird dadurch nicht beeinflusst. Im weiteren Bahnverlauf kommt es daher abwechselnd zur Ausgabe und Unterdrückung von jeweils zwei aufeinander folgenden Schaltsignalen.

23.2.5 Angenäherte Schaltposition

Wird bei der satzbezogenen Schaltsignalausgabe eine Schaltposition nicht exakt erreicht, z. B. bei Bahnsteuerbetrieb und Verfahren in mehr als einer Geometrieachse, wird zu dem Zeitpunkt geschaltet, ab dem sich die Wegdifferenz zwischen der Istposition der beteiligten Geometrieachsen und der programmierten Schaltposition wieder vergrößert.

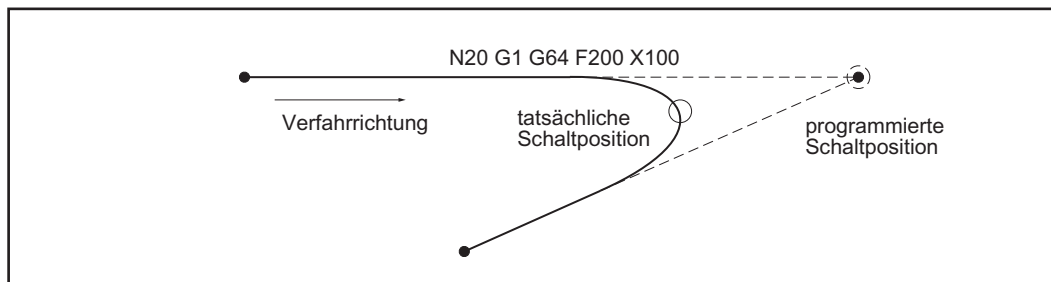


Bild 23-2 Schaltpositionsverschiebung bei Bahnsteuerbetrieb

23.2.6 Programmierte Schaltpositionsverschiebung

Programmierte Schaltpositionsverschiebung

Für die satzbezogene Schaltsignalausgabe kann eine wegbezogene Verschiebung der Schaltposition programmiert werden:

- Verschiebungsweg **negativ** = Vorhalt
Mit einem negativen Verschiebungsweg wird die Schaltposition **vor** die im Teileprogrammsatz programmierte Sollposition verschoben.
Wird ein zu großer negativer Verschiebungsweg programmiert, d. h. beim Erkennen des Flankenwechsels ist die Schaltposition bereits überschritten, wird das Schaltsignal sofort geschaltet.
- Verschiebungsweg **positiv** = Nachhalt
Mit einem positiven Verschiebungsweg wird die Schaltposition **hinter** die im Teileprogrammsatz programmierte Sollposition verschoben.

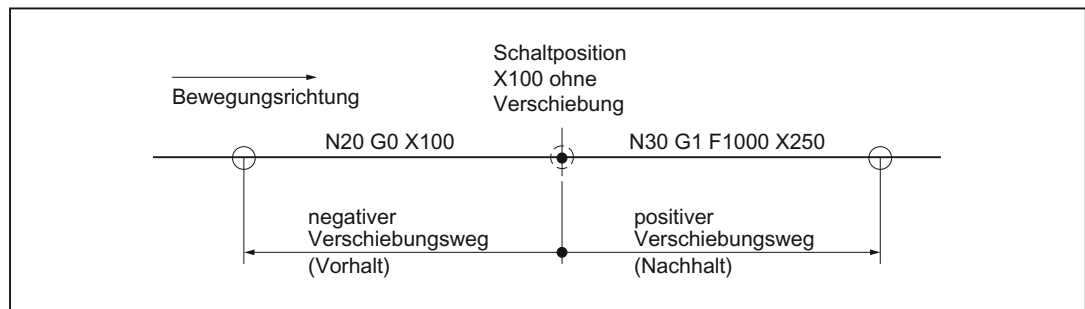


Bild 23-3 Programmierte Schaltpositionsverschiebung

Bahnbezug

Der Verschiebungsweg ist eine Wegangabe, die sich auf die programmierte Bahn bezieht. Dabei wird vereinfachend von einer linearen Bewegung ausgegangen. Bahnkrümmungen werden nicht berücksichtigt.

Verhalten bei Einzelsatz und G60

Aufgrund der internen Verfahrlogik wirken negative Verschiebungswege (Vorhalt) im Zusammenhang mit folgenden Standardfunktionen nicht:

- Einzelsatz
- Genauhalt am Satzende (G60)

23.2.7 Verhalten bei Teileprogramm-Unterbrechung

Nach einer Unterbrechung des Teileprogramms (NC-STOP) und anschließendem Wechsel in die Betriebsart JOG wird die Technologie-Funktion ausgeschaltet bzw. keine Schaltsignale mehr ausgegeben.

23.3 Inbetriebnahme

Die Technologie-Funktion wird erst wieder eingeschaltet bzw. Schaltsignale ausgegeben, nachdem wieder in die Betriebsart AUTOMATIK gewechselt und das Teileprogramm fortgesetzt wurde (NC-START).

23.3 Inbetriebnahme

23.3.1 Aktivierung

Vor Inbetriebnahme der Technologie-Funktion ist sicherzustellen, dass der entsprechende Compile-Zyklus geladen und aktiviert ist (siehe auch Kapitel "TE01: Installation und Aktivierung ladbarer Compile-Zyklen (Seite 689)").

Aktivierung

Die Technologie-Funktion "Taktunabhängige bahnsynchrone Schaltsignalausgabe" wird über folgendes Maschinendatum aktiviert:

MD60948 \$MN_CC_ACTIVE_IN_CHAN_HSLC[0], Bit 0 = 1

Hinweis

Die Technologie-Funktion "Taktunabhängige bahnsynchrone Schaltsignalausgabe" ist nur in **einem** Kanal der NC verfügbar.

23.3.2 Speicherkonfiguration

Die Technologiefunktion benötigt **zusätzliche** Daten im NC-internen Satzspeicher. Für folgende speicherkonfigurierende kanalspezifischen Maschinendaten sind die Werte zu erhöhen:

- MD28090 \$MC_MM_NUM_CC_BLOCK_ELEMENTS += 1 (Anzahl Satzelemente für Compile-Zyklen)
- MD28100 \$MN_MM_NUM_CC_BLOCK_USER_MEM += 10 (Größe des Satzspeichers für Compile-Zyklen (DRAM) in kByte)

23.3.3 Parametrierung der digitalen Onboard-Ausgänge

Parametrierung

Für das Schaltsignal wird ein digitaler Ausgang der lokalen I / O-Peripherie benötigt.

Dazu muss über das folgende Maschinendatum mindestens 1 digitales Ausgangbyte festgelegt sein:

MD10360 \$MN_FASTIO_DIG_NUM_OUTPUTS ≥ 1 (Anzahl der aktiven digitalen Ausgangsbytes)

Literatur

Die vollständige Beschreibung der Parametrierung eines digitalen Ausgangs findet sich in:

- Funktionshandbuch Erweiterungsfunktionen; Digitale und analoge NC-Peripherie (A4)

23.3.4 Parametrierung des Schaltsignals

Ausgangsnummer des Schaltsignals

Nach der Inbetriebnahme des Compile-Zyklus wird das folgende funktionspezifische Maschinendatum unter den kanalspezifischen Maschinendaten angezeigt:

MD62560 \$MC_FASTON_NUM_DIG_OUTPUT (Nummer des digitalen Ausgangs des Schaltsignals)

Darin ist die Nummer n des digitalen Onboard-Ausgangs einzutragen, über den das Schaltsignal ausgegeben werden soll:

n = 1, 2, 3 oder 4

Deaktivierung

Wird als Nummer des digitalen Ausgangs n = 0 vorgegeben, wird die Funktion ausgeschaltet. Es erfolgt **keine** Meldung oder Alarm.

Auswirkung auf andere Ausgangssignale

Die Hardware-Timer-gesteuerte Ausgabe des Schaltsignals auf den parametrierten Ausgang bewirkt eine Verzögerung der Signalausgabe für die anderen digitalen Onboard-Ausgänge, z. B. durch Synchronaktionen, um 2 IPO-Takte.

23.3.5 Parametrierung der Geometrieachsen

Standardeinstellung

Maschinen für das Hochgeschwindigkeits-Laserschneiden haben im Normalfall zwei Geometrieachsen, die in den beiden folgenden Maschinendaten projiziert sind:

MD20050_\$MC_AXCONF_GEOAX_ASSIGN_TAB[0]

MD20050_\$MC_AXCONF_GEOAX_ASSIGN_TAB[1]

Von diesen beiden Geometrieachsen wird die Berechnung der Schaltzeitpunkte abgeleitet.

Hinweis

Die projektierte Achsauswahl für die Berechnung der Schaltzeitpunkte kann durch eine Neudefinition der ersten und zweiten Geometrieachse über die Programmanweisungen `GEOAX (1, <Achsname>)` und `GEOAX (2, <Achsname>)` im Teileprogramm verändert werden.

Wichtig:

Damit die Änderung von der Funktion korrekt berücksichtigt werden kann, muss sie vor der Interpretation des `CC_FASTON`-Befehls erfolgen.

Änderung der Standardeinstellung

Die Standardeinstellung kann bei einer abweichenden Maschinenkonfiguration (z. B. Definition einer dritten Geometrieachse) über das folgende Maschinendatum angepasst werden:

MD60948 \$MN_CC_ACTIVE_IN_CHAN_HSCL[1]

Wert	Bedeutung
\$MN_CC_ACTIVE_IN_CHAN_HSCL[1]='H3'	Standardeinstellung. Die Berechnung der Schaltzeitpunkte wird von der ersten und zweiten Geometrieachse abgeleitet.
\$MN_CC_ACTIVE_IN_CHAN_HSCL[1]='H7'	Die Berechnung der Schaltzeitpunkte wird von der ersten, zweiten und dritten Geometrieachse abgeleitet.

Eine Änderung wird nach dem nächsten NC-Hochlauf wirksam.

23.4 Programmierung

23.4.1 Einschalten der satzbezogenen Schaltsignalausgabe (CC_FASTON)

Syntax

`CC_FASTON (DIFFON, DIFFOFF [, FEEDTOSWITCH])`

`CC_FASTON ()` ist ein Prozedur-Aufruf und muss daher in einem eigenen Teileprogrammsatz programmiert werden.

Parameter

Die Parameter der Prozedur `CC_FASTON ()` haben folgende Bedeutung:

Parameter	Bedeutung
<DIFFON>	Länge* des Verschiebungswegs für das Einschalten des Schaltsignals.
<DIFFOFF>	Länge* des Verschiebungswegs für das Ausschalten des Schaltsignals.

Parameter	Bedeutung
<FEEDTOSWITCH>	<p>Der Parameter ist optional.</p> <p>Wird der Parameter im Prozedur-Aufruf nicht angegeben, wird als Schaltkriterium der G0-Flankenwechsel verwendet.</p> <p>Wird der Parameter im Prozedur-Aufruf angegeben, beinhaltet er als Schaltkriterium den Geschwindigkeits-Schwellwert, bei dessen Unter- bzw. Überschreiten das Schaltsignal ein- bzw. ausgeschaltet wird.</p>
* Die zugrunde liegende Einheit (inch oder mm) ist abhängig von der aktuellen Maßangaben-Programmierung (G70 / G71 / G700 / G710).	

Programmierbeispiel

Programmierung	Kommentar
DEF REAL DIFFON = -0.08	; Länge des Verschiebungswegs für das Einschalten des Schaltsignals = -0.08
DEF REAL DIFFOFF = 0.08	; Länge des Verschiebungswegs für das Ausschalten des Schaltsignals = 0.08
DEF REAL FEEDTOSWITCH = 20000	; Geschwindigkeits-Schwellwert = 20000
CC_FASTON(DIFFON,DIFFOFF,FEEDTOSWITCH)	; Satzbezogene Schaltsignalausgabe einschalten (mit Geschwindigkeits-Schwellwert als Schaltkriterium)

Ändern von Parametern

Die Parameter der Prozedur `CC_FASTON()` können im Verlauf des Teileprogramms jederzeit geändert werden. Dazu ist der Prozedur-Aufruf mit den neuen Parameterwerten erneut anzugeben. Ein Wechsel des Schaltkriteriums (G0-Flankenwechsel / Geschwindigkeits-Schwellwert) ist dabei ebenfalls erlaubt.

Reset-Verhalten

Bei Reset (NC-RESET oder Programmende) wird die Funktion ausgeschaltet.

23.4.2 Einschalten der bahnlangenbezogenen Schaltsignalausgabe (CC_FASTON_CONT)

Syntax

```
CC_FASTON_CONT (PATH_DISTANCE_ON, PATH_DISTANCE_OFF)
```

`CC_FASTON_CONT()` ist ein Prozedur-Aufruf und muss daher in einem eigenen Teileprogrammsatz programmiert werden.

Parameter

Die Parameter der Prozedur `CC_FASTON_CONT()` haben folgende Bedeutung:

Parameter	Bedeutung
< PATH_DISTANCE_ON >	Länge* der Streckenabschnitte mit Bearbeitung (s_1).
< PATH_DISTANCE_OFF >	Länge* der Streckenabschnitte ohne Bearbeitung (s_2).
* Die zugrunde liegende Einheit (inch oder mm) ist abhängig von der aktuellen Maßangaben-Programmierung (G70 / G71 / G700 / G710).	

Programmierbeispiel

Programmierung	Kommentar
DEF REAL PATH_DISTANCE_ON = 0.5	; Länge der Streckenabschnitte mit Bearbeitung = 0.5
DEF REAL PATH_DISTANCE_OFF = 1.0	; Länge der Streckenabschnitte ohne Bearbeitung = 1.0
CC_FASTON_CONT(PATH_DISTANCE_ON, PATH_DISTANCE_OFF)	; Bahnlängenbezogene Schaltsignalausgabe einschalten.

Ändern von Parametern

Die Parameter der Prozedur `CC_FASTON_CONT()` können im Verlauf des Teileprogramms jederzeit geändert werden. Dazu ist der Prozedur-Aufruf mit den neuen Parameterwerten erneut anzugeben.

Reset-Verhalten

Bei Reset (NC-RESET oder Programmende) wird die Funktion ausgeschaltet.

23.4.3 Ausschalten (CC_FASTOFF)

Syntax

```
CC_FASTOFF
```

`CC_FASTOFF` ist ein Prozedur-Aufruf und muss daher in einem eigenen Teileprogrammsatz programmiert werden.

Funktionalität

Mit dem Prozedur-Aufruf `CC_FASTOFF` wird die "Taktunabhängige, bahnsynchrone Schaltsignalausgabe" ausgeschaltet.

23.5 Funktionsspezifische Alarmtexte

Zum Vorgehen beim Anlegen von funktionsspezifischen Alarmtexten siehe Kapitel "Anlegen von Alarmtexten (Seite 696)".

23.6 Randbedingungen

23.6.1 Satzsuchlauf

Schaltsignalausgabe bei Satzsuchlauf

Erfolgt ein Satzsuchlauf auf einen Teileprogrammsatz, der nach einem Prozedur-Aufruf `CC_FASTON()` zum Einschalten der Technologie-Funktion liegt, wird mit der nächsten Verfahrbewegung das Schaltsignal eingeschaltet. Dies führt insbesondere dazu, dass von der Startposition der Geometrieachsen bis zum Wiederaufsetzpunkt an der Kontur mit eingeschaltetem Schaltsignal verfahren wird.

Beispiel

Normaler Ablauf:

Im normalen Ablauf der Teileprogrammbearbeitung wird das Schaltsignal erstmalig zu Beginn des Teileprogrammsatzes `N60` eingeschaltet.

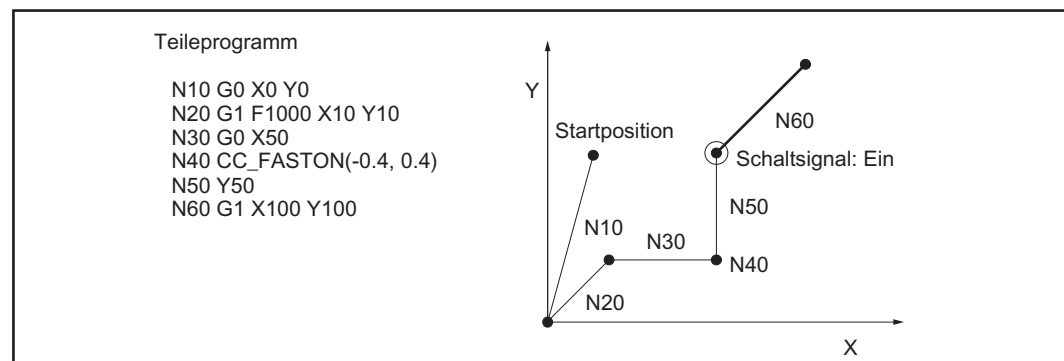


Bild 23-4 Schaltsignal bei Teileprogrammbearbeitung

Ablauf nach Satzsuchlauf:

Wird ein Satzsuchlauf auf den Satzendpunkt des Teileprogrammsatzes `N60` ausgeführt, wird das Schaltsignal ab der Startposition der Geometrieachsen eingeschaltet.

23.6 Randbedingungen

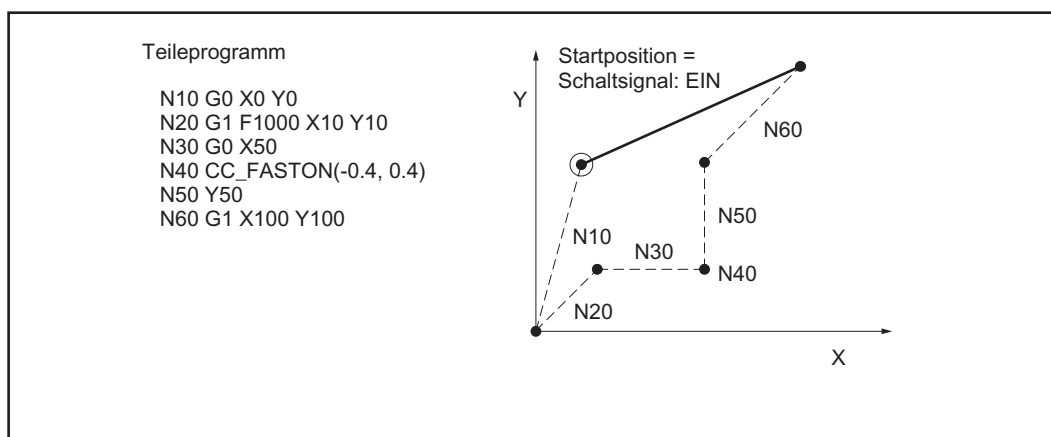


Bild 23-5 Schaltsignal nach Satzsuchlauf

Unterdrückung der Schaltsignalausgabe

Um bei oben genannter Konstellation das Einschalten des Schaltsignals im Wiederanfahrtsatz zu unterdrücken, müssen vom Anwender (Maschinenhersteller) geeignete Maßnahmen, z. B. Sperren des Schaltsignals, ergriffen werden.

Hinweis

Das Unterdrücken der Schaltsignalausgabe beim Wiederanfahren an die Kontur, z. B. nach Satzsuchlauf, liegt in der alleinigen Verantwortung des Anwenders (Maschinenhersteller).

Literatur

Eine Beschreibung des Satzsuchlaufs findet sich in:

Funktionshandbuch Grundfunktionen; BAG, Kanal, Programmbetrieb, Reset-Verhalten (K1)

23.6.2 Transformationen

Die Funktion arbeitet nur bei ausgeschalteter Transformation korrekt. Eine Überwachung findet nicht statt.

Zur Beschreibung der Transformationen (siehe Kapitel "TE4: Transformationspaket Handling - nur 840D sl (Seite 775)").

Literatur:

Funktionshandbuch Erweiterungsfunktionen; Kinematische Transformation (M1)

23.6.3 Kompensationen

Folgende Kompensationen werden bei der Berechnung der Schaltpositionen nicht berücksichtigt:

- Temperaturkompensation
- Durchhangkompensation

Eine Beschreibung der Kompensationen findet sich in:

Literatur:

Funktionshandbuch Erweiterungsfunktionen; Kompensationen (K3)

23.6.4 Werkzeugradiuskorrektur (WRK)

Im Rahmen der Werkzeugradiuskorrektur werden steuerungsintern Teileprogrammsätze (Korrektursätze) in das Teileprogramm eingefügt. Ein Korrektursatz wird, bezogen auf die Schaltsignalausgabe, immer dem nächsten programmierten Teileprogrammsatz zugeschlagen.

Eine Beschreibung der Werkzeugradiuskorrektur findet sich in:

Literatur:

Funktionshandbuch Grundfunktionen; Werkzeugkorrektur (W1), Kapitel: Werkzeugradiuskorrektur

23.6.5 Bahnsteuerbetrieb

Bahnsteuerbetrieb

Obwohl die Prozedur-Aufrufe `CC_FASTON()`, `CC_FASTON_CONT()` und `CC_FASTOFF` jeweils in eigenen Teileprogrammsätzen programmiert werden müssen, führt dies während eines aktiven Bahnsteuerbetriebs (G64, G641, ...) zu keinem Geschwindigkeitseinbruch.

Bahnsteuerbetrieb (ADIS)

Wenn beim Bahnsteuerbetrieb mit programmierbarem Überschleifverhalten (G641 ADIS) steuerungsintern ein Teileprogrammsatz in das Teileprogramm eingefügt wird, dann wird die ursprünglich programmierte Schaltposition nicht erreicht und die Schaltsignalausgabe erfolgt stattdessen in der Mitte des Überschleifsatzes.

Literatur

Eine Beschreibung des Bahnsteuerbetriebs findet sich in:

Funktionshandbuch Grundfunktionen; Bahnsteuerbetrieb, Genauhalt und LookAhead (B1)

23.6.6 Softwarenocken

Da der Hardware-Timer auch für die Funktion "Software Nocken" verwendet wird, ist eine gleichzeitige Nutzung der Funktion "Taktunabhängige bahnsynchrone Schaltsignalausgabe" mit Software-Nocken nicht möglich.

Im Fehlerfall wird folgender Alarm angezeigt:

Alarm 75500 "Kanal *Kanalnummer*, falsche Konfiguration der Funktion: Taktunabhängige Schaltsignalausgabe"

Eine Beschreibung der Softwarenocken findet sich in:

Literatur:

Funktionshandbuch Erweiterungsfunktionen; Softwarenocken, Wegschaltsignale (N3)

23.7 Datenlisten

23.7.1 Maschinendaten

23.7.1.1 Allgemeine Maschinendaten

Nummer	Bezeichner: \$MN_	Beschreibung
10360	FASTO_NUM_DIG_OUTPUTS	Anzahl digitaler Ausgangsbytes

23.7.1.2 Kanal-spezifische Maschinendaten

Nummer	Bezeichner: \$MC_	Beschreibung
20050	AXCONF_GEOAX_ASSIGN_TAB	Zuordnung Geometrieachse zu Kanalachse
28090	MM_NUM_CC_BLOCK_ELEMENTS	Anzahl Blockelemente für CC
28100	MM_NUM_CC_BLOCK_USER_MEM	Größe Satzspeicher für CC
62560	FASTON_NUM_DIG_OUTPUT	Nummer des digitalen Onboard-Ausgangs für das Schaltsignal

TE9: Achspaar-Kollisionsschutz

24.1 Kurzbeschreibung

Hinweis

Compile-Zyklus

Vor Inbetriebnahme der Funktion ist sicherzustellen, dass der entsprechende Compile-Zyklus geladen und aktiviert ist (siehe Kapitel "TE01: Installation und Aktivierung ladbarer Compile-Zyklen (Seite 689)").

Funktion

Die Funktion "Achspaar-Kollisionsschutz" ermöglicht die paarweise Überwachung von Maschinenachsen, die auf einem gemeinsamen Führungselement einer Maschine angeordnet sind, auf Kollision und maximalen Abstand.

Funktionskürzel

Das Kürzel für funktionsspezifische Bezeichner von Maschinendaten, Systemvariablen etc. der Funktion ist:

PROTECT (axial collision PROTECTion)

Maximale Anzahl Achspaare

Es können maximal **20** Achspaare parametrierbar werden.

24.2 Funktionsbeschreibung

Die Funktion "Achspaar Kollisionsschutz" ist eine Schutzfunktion für Maschinenachsen, die in einer Werkzeugmaschine so angebracht sind (z. B. auf einer gemeinsamen Führungsschiene), dass sie aufgrund fehlerhafter Bedienung oder Programmierung miteinander kollidieren könnten.

Die Überwachung der Maschinenachsen erfolgt immer paarweise, d. h. es sind immer jeweils zwei Maschinenachsen zu parametrieren, die gegeneinander überwacht werden. Die überwachten Maschinenachsen können dabei in unterschiedlichen Maschinenkoordinatensystemen liegen.

Kollisionsschutz

Die Funktion berechnet zyklisch aus den aktuellen Istpositionen und Istgeschwindigkeiten sowie dem Offset der Maschinenkoordinatensysteme und den achsspezifischen Bremsbeschleunigungen den Abstand der Stillstandspositionen der Maschinenachsen. Ergibt sich dabei ein Abstand kleiner dem parametrisierten Schutzfenster, werden die Maschinenachsen bis zum Stillstand abgebremst. Der über das Schutzfenster vorgegebene Mindestabstand wird dabei nicht unterschritten.

Abstandsüberwachung

Über eine entsprechende Wahl des Offsetvektors kann die Funktion auch dazu verwendet werden, den Abstand der Maschinenachsen auf einen Maximalwert (maximaler Abstandsvektor) zu überwachen.

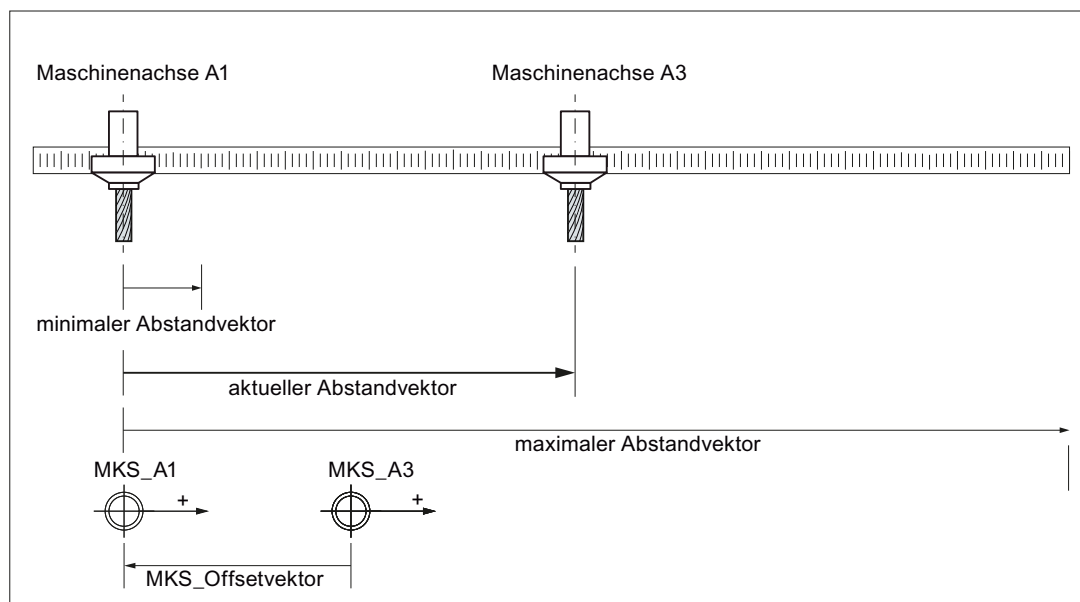


Bild 24-1 Prinzipieller Aufbau

Überwachungsstatus

Der aktuelle Status eines Achspaares kann aus der im NC-Programm optional zu definierenden globalen Anwendervariablen (GUD) `_PROTECT_STATUS` (Seite 900) gelesen werden.

24.3 Inbetriebnahme

24.3.1 Freigabe der Technologiefunktion (Option)

Die Funktion ist eine Option, die über das Lizenzmanagement der Hardware zugeordnet werden muss:

6FC5800-0AN06-0YB0, "RMCC/PROT Kollisionsschutz Achsen"

Zu Testzwecken kann die Funktion durch Setzen des Optionsdatums freigegeben werden:

MD19610 \$ON_TECHNO_EXTENSION_MASK[2], BIT4 = 1

24.3.2 Aktivierung der Technologiefunktion

Aktivierungsregeln

Die Aktivierung der Funktion muss kanalspezifisch für folgende Kanäle der NC erfolgen:

- Unabhängig davon, welchen Kanälen die zu überwachenden Maschinenachsen zugeordnet sind, immer im 1. Kanal der NC
- In allen Kanälen, denen per Maschinendatenparametrierung von der Funktion zu überwachende Maschinenachsen zugeordnet sind
- In allen Kanälen, denen zu einem späteren Zeitpunkt z.B. durch Achstausch, von der Funktion zu überwachenden Maschinenachsen zugeordnet werden

Aktivierung

Die Aktivierung der Funktion erfolgt kanalspezifische über das Maschinendatum:

MD60972 \$MN_CC_ACTIVE_IN_CHAN_PROT[0], Bit n = 1

mit n = 0, 1, 2, ..., entsprechend dem (n+1)-ten Kanal der NC

Siehe auch

TE01: Installation und Aktivierung ladbarer Compile-Zyklen (Seite 689)

24.3.3 Aktivierung der Zusatzfunktionen

Die Aktivierung der Zusatzfunktionen erfolgt achspaar-spezifisch über das Optionsdatum:

MD61535 \$MN_CC_PROTECT_OPTIONS[<a>]

mit a = 0, 1, 2, ... (maximale Anzahl Achspare - 1) entsprechend Achspaar 1, 2, 3, ...

Bit	Wert	Bedeutung
0	1	Achspaar-spezifische Aktivierung / Deaktivierung der Funktion "Achspaar-Kollisionsschutz" über NC/PLC-Nahtstellensignal (Seite 902) Hinweis Nach Aktivierung dieser Zusatzfunktion befindet sich die Schutzfunktion im Überwachungsstatus (Seite 900) == 1 (angewählt, aber noch nicht aktiv)

24.3.4 Definition eines Achspaares

Die Definition eines zu überwachenden Maschinenachspaares erfolgt achspaarsspezifisch im Maschinendatum:

MD61516 \$MN_CC_PROTECT_PAIRS[<a>] = <yyxx>

mit a = 0, 1, 2, ... (maximale Anzahl Achspare - 1) entsprechend Achspaar 1, 2, 3, ...

<yyxx>	Bedeutung
xx	1. und 2. Dezimalstelle → Achsnummer der 1. Maschinenachse
yy	3. und 4. Dezimalstelle → Achsnummer der 2. Maschinenachse

Beispiel

Definition des 1. Achspaares:

- 1. Achse: 4. Maschinenachse
- 2. Achse: 12. Maschinenachse

MD61516 \$MN_CC_PROTECT_PAIRS[0] = 1204

24.3.5 Freifahrrichtung

Über das Maschinendatum wird die Verfahrrichtung zum Freifahren der jeweiligen Maschinenachse eingestellt:

MD61517 \$MN_CC_PROTECT_SAFE_DIR[<a>] = <yyxx>

mit a = 0, 1, 2, ... (maximale Anzahl Achspare - 1) entsprechend Achspaar 1, 2, 3, ...

<yyxx>	Bedeutung
xx	Freifahrrichtung für die 1. Achse des Achspaares
yy	Freifahrrichtung für die 2. Achse des Achspaares
Freifahrrichtung in positiver Verfahrrichtung der Maschinenachse: xx bzw. yy > 0	
Freifahrrichtung in negativer Verfahrrichtung der Maschinenachse: xx bzw. yy = 0	

Hinweis

Änderung der Freifahrrichtung

Eine Änderung der Freifahrrichtung im Maschinendatum MD61517

\$MN_CC_PROTECT_SAFE_DIR[<Achspaar>] darf nur vorgenommen werden, wenn die Schutzfunktion für das Achspaar nicht aktiv ist (MD61516

\$MN_CC_PROTECT_PAIRS[<Achspaar>] == 0).

24.3.6 Offset der Maschinenkoordinatensysteme

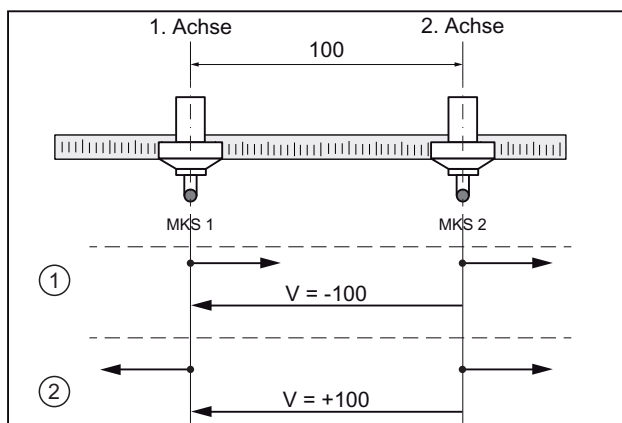
Liegen die Maschinenachsen des Achspaares in verschiedenen Maschinenkoordinatensystemen, muss der entsprechende Verschiebungsvektor in folgendem Maschinendatum angegeben werden:

MD61518 \$MN_CC_PROTECT_OFFSET[<a>] = <Verschiebungsvektor>

mit a = 0, 1, 2, ... (maximale Anzahl Achspaare - 1) entsprechend Achspaar 1, 2, 3, ...

Der Verschiebungsvektor ist als Vektor vom Ursprung des Maschinenkoordinatensystems der 2. Achse des Achspaars zum Ursprung des Maschinenkoordinatensystems der 1. Achse, bezogen auf das Maschinenkoordinatensystem der 1. Achse, anzugeben.

Liegen beide Maschinenachsen im selben Maschinenkoordinatensystem, ist als Verschiebungsvektor der Wert 0 anzugeben.



① Fall 1: Gleichsinnige MKS-Orientierung

② Fall 2: Gegensinnige MKS-Orientierung

V Verschiebungsvektor

Hinweis

Änderung des Verschiebungsvektors

Eine Änderung des Verschiebungsvektors im Maschinendatum MD61518 \$MN_CC_PROTECT_OFFSET[<Achspaar>] darf nur vorgenommen werden, wenn die Schutzfunktion für das Achspaar nicht aktiv ist (MD61516 \$MN_CC_PROTECT_PAIRS[<Achspaar>] == 0).

24.3.7

Schutzfenster

Über das Maschinendatum wird das Schutzfenster bzw. der Mindestabstand festgelegt, den die Achsen des Achspaars nicht unterschreiten dürfen:

MD61519 \$MN_CC_PROTECT_WINDOW[<a>] = <Mindestabstand>

mit a = 0, 1, 2, ... (maximale Anzahl Achspaare - 1) entsprechend Achspaar 1, 2, 3, ...

Bei Annäherung des Abstands an den Mindestabstand werden die Achsen mit der funktionsspezifischen Beschleunigung (Seite 900) abgebremst.

Über die Schutzfenster-Erweiterung (Seite 898) kann das Schutzfenster, z.B. in einem NC-Programm, dynamisch erweitert werden.

Hinweis

Änderung des Schutzfensters

Eine Änderung des Schutzfensters im Maschinendatum MD61519
 $\$MN_CC_PROTECT_WINDOW[<Achspaar>]$ darf auch vorgenommen werden, wenn die Schutzfunktion für das Achspaar **aktiv** ist (MD61516
 $\$MN_CC_PROTECT_PAIRS[<Achspaar>] \neq 0$).

24.3.8 Orientierung

Über das Maschinendatum wird die Orientierung der Achsen des Achspaares zueinander angegeben:

MD61532 $\$MN_CC_PROTECT_DIR_IS_REVERSE[<a>] = <Wert>$

mit $a = 0, 1, 2, \dots$ (maximale Anzahl Achspare - 1) entsprechend Achspaar 1, 2, 3, ...

<Wert>	Bedeutung
0	Geichsinnige Orientierung
1	Gegensinnige Orientierung

Hinweis

Änderung der Orientierung

Eine Änderung der Orientierung im Maschinendatum MD61532
 $\$MN_CC_PROTECT_DIR_IS_REVERSE[<Achspaar>]$ darf nur vorgenommen werden, wenn die Schutzfunktion für das Achspaar nicht aktiv ist (MD61516
 $\$MN_CC_PROTECT_PAIRS[<Achspaar>] == 0$).

24.3.9 Schutzfenster-Erweiterung

Über das Maschinendatum kann das Schutzfenster (Seite 897) dynamisch **vergrößert** werden:

MD61533 $\$MN_CC_PROTECT_WINDOW_EXTENSION[<a>] = <Erweiterung>$

mit $a = 0, 1, 2, \dots$ (maximale Anzahl Achspare - 1) entsprechend Achspaar 1, 2, 3, ...

Das wirksame Schutzfenster eines Achspaares ergibt sich somit zu:

Wirksames Schutzfenster[<Achspaar>] =

MD61519 $\$MN_CC_PROTECT_WINDOW[<Achspaar>] +$

MD61533 $\$MN_CC_PROTECT_WINDOW_EXTENSION[<Achspaar>]$

Eine Verkleinerung des Schutzfenster durch Eingabe eines negativen Wertes ist **nicht** möglich.

Hinweis

Änderung der Schutzfenster-Erweiterung

Eine Änderung der Schutzfenster-Erweiterung im Maschinendatum MD61533 \$MN_CC_PROTECT_WINDOW_EXTENSION[<Achspaar>] darf auch bei aktiver Schutzfunktion, z.B. aus dem NC-Programm heraus, vorgenommen und mit Auslösen von "Maschinendaten wirksam setzen" aktiviert werden.

24.3.10 Aktivierung der Schutzfunktion

Statische Aktivierung über Maschinendaten

Die statische Aktivierung der Schutzfunktion für ein Achspaar erfolgt, sobald folgende Voraussetzungen erfüllt sind:

- Beide Maschinenachsen des Achspaares sind referenziert
- Für das Achspaar sind gültige Maschinenachsen parametrier:
MD61516 \$MN_CC_PROTECT_PAIRS[<Achspaar>] = <gültiges Maschinenachspaar>
- Die Zusatzfunktion "Achspaarsspezifische Aktivierung / Deaktivierung über achsspezifische NC/PLC-Nahtstellensignale" ist nicht aktiv:
MD61535 \$MN_CC_PROTECT_OPTIONS[<Achspaar>], Bit 0 = 0

Dynamische Aktivierung über NC/PLC-Nahtstellensignal

Die dynamische Aktivierung der Schutzfunktion für ein Achspaar erfolgt, sobald folgende Voraussetzungen erfüllt sind:

- Beide Maschinenachsen des Achspaares sind referenziert
- Für das Achspaar sind gültige Maschinenachsen parametrier:
MD61516 \$MN_CC_PROTECT_PAIRS[<Achspaar>] = <gültiges Maschinenachspaar>
- Die Zusatzfunktion "Achspaarsspezifische Aktivierung / Deaktivierung über achsspezifische NC/PLC-Nahtstellensignale" ist aktiv:
MD61535 \$MN_CC_PROTECT_OPTIONS[<Achspaar>], Bit 0 = 1
- Das achsspezifische NC/PLC-Nahtstellensignal zur Aktivierung der Schutzfunktion ist für eine der beiden Maschinenachsen des Achspaares gesetzt:
DB31,DBX24.3 == 1

Abstand kleiner als das Schutzfenster

Ist zum Zeitpunkt, zu dem die Schutzfunktion aktiv wird, der Abstand der beiden Maschinenachsen kleiner dem parametrierten Mindestabstand des Schutzfenster, müssen die Maschinenachsen vom Maschinenbediener freigefahren werden. Die Steuerung erlaubt in diesem Zustand nur Verfahrbewegungen in der parametrierten Freifahrtrichtung (Seite 896) der Maschinenachsen.

Überwachungsstatus

Der aktuelle Überwachungsstatus eines Achspaares kann über die globale Anwendervariable `_PROTECT_STATUS` (Seite 900) gelesen werden.

24.3.11 Achsspezifische Beschleunigung

Die Beschleunigung, mit der durch die Schutzfunktion bei kritischen Annäherung die beiden Maschinenachsen des Achspaares abgebremst werden, wird eingestellt über:

MD63514 `$MA_CC_PROTECT_ACCEL[<Achse>] = <Beschleunigung>`

mit `<Achse>`: Maschinenachsname z.B. AX1, AX2, ...

Hinweis

Ohne Ruckbegrenzung

Die in MD63514 `$MA_CC_PROTECT_ACCEL` eingestellte Bremsbeschleunigung wirkt ohne Ruckbegrenzung.

Vorrang der funktionspezifischen Beschleunigung

Zur Berechnung des Bremszeitpunktes wird von der Schutzfunktion ausschließlich die funktionspezifische Beschleunigung der Maschinenachsen aus MD63514 `$MA_CC_PROTECT_ACCEL` verwendet. Die aktuelle Beschleunigung der Maschinenachse im Kanal wird von der Schutzfunktion nicht berücksichtigt

Hinweis

Bahnbezug

Werden von der Schutzfunktion überwachte Maschinenachsen von einem Kanal im Bahnbezug mit anderen Achsen verfahren, geht dieser Bahnbezug verloren, sobald der Achsverbund aufgrund der Schutzfunktion abgebremst wird. Die von der Schutzfunktion überwachten Maschinenachsen werden dabei mit ihren funktionspezifischen Beschleunigungen aus Maschinendatum MD63514 `$MA_CC_PROTECT_ACCEL` abgebremst. Die übrigen Achsen des Achsverbundes werden mit der aktuellen Bahnbeschleunigung des Kanals abgebremst.

24.3.12 Überwachungsstatus (GUD)

Der aktuelle Status eines Achspaares wird über die globale Anwendervariable `_PROTECT_STATUS` angezeigt.

Die Variable ist standartmäßig nicht vorhanden. Bei Bedarf muss sie in der Definitionsdatei GUD.DEF definiert werden.

Definition

```
DEF NCK INT _PROTECT_STATUS[ <Anzahl parametrierter Achspaares> ]
```


mit <Anzahl parametrierter Achspaare> = 1, 2, 3, ... (maximale Anzahl Achspaare)

Wertebereich

Wert	Bedeutung: Die Überwachung des Achspaars ist ...
0	nicht aktiv
1	angewählt, aber noch nicht aktiv
2	aktiv, die Achsen werden aktuell nicht gebremst
3	aktiv, die Achsen werden aktuell <u>gebremst</u>
4	abgewählt, aber noch aktiv

24.3.13 PLC-Schnittstelle: Achsspezifische Bremsvorgänge

Über das Maschinendatum kann innerhalb des allgemeinen Systemvariablenfeldes \$A_DBD ein Doppelwort (vier Byte) eine achsspezifische Bremschnittstelle festgelegt werden. Über die Bremschnittstelle wird bei einer kritischen Annäherung der Achsen des Achspaars das aktuelle Abbremsen der Maschinenachsen angezeigt.

MD61534 \$MN_CC_PROTECT_A_DBD_INDEX = <Wert>

<Wert>	Bedeutung
-1	Deaktivierung der Ausgabe.
≥ 0	Index der Bremschnittstelle innerhalb des Systemvariablenfeldes: \$A_DBD[<Index>] mit Index = 0, 4, 8, 12, ...

Hinweis

Doppelwort-Index

Der Anfangsindex kann im Maschinendatum byteweise (0, 1, 2, ...) angegeben werden. Da auf die Systemvariable \$A_DBD von der PLC aus doppelwortweise zugegriffen wird, wird ein Anfangsindex der nicht auf einer Doppelwortgrenze (0, 4, 8, ...) liegt, auf die nächste Doppelwortgrenze (0, 4, 8, 12, ...) **abgerundet**: $\text{Index} = (\text{Index DIV } 4) * 4$

Bremschnittstelle

Jedem Bit der Bremschnittstelle ist eine Maschinenachse zugeordnet:

Bit n: (n+1)-te Maschinenachse, mit n = 0, 1, 2, ...

Bit	31	30	29	28	...	3	2	1	0
MA ¹⁾	---	31	30	29	...	4	3	2	1
1) Maschinenachsnummer									

Bit n	Bedeutung
0	Die Maschinenachse (n+1) wird nicht gebremst
1	Die Maschinenachse (n+1) wird gebremst

Literatur

Für das Lesen der Bremsschnittstelle vom PLC-Anwenderprogramm aus steht der Baustein FC21, Funktion 3 zur Verfügung.

Funktionshandbuch Grundfunktionen; Kapitel "PLC-Grundprogramm für SINUMERIK 840D sl" > "Bausteinbeschreibungen" > "FC21: Transfer Datenaustausch NC/PLC" > "Funktion 3, 4: Schneller Datenaustausch PLC-NC"

24.3.14 PLC-Schnittstelle: Achspaar-spezifische Aktivierung der Schutzfunktion

Wenn die Zusatzfunktion "Aktivierung / Deaktivierung der Funktion "Achspaar-Kollisionsschutz" über NC/PLC-Nahtstellensignal (Seite 895)" aktiv ist, kann mit dem achsspezifischen Nahtstellensignal die Schutzfunktion für das Achspaar vom PLC-Anwenderprogramm aus ein- und ausgeschaltet werden:

DB31,DBX24.3 (Kollisionsschutz einschalten)

Einschalten

Die Schutzfunktion ist eingeschaltet, wenn das achsspezifische Nahtstellensignal für eine der beiden Maschinenachsen des Achspaares gesetzt ist.

Ausschalten

Die Schutzfunktion ist ausgeschaltet, wenn das achsspezifische Nahtstellensignal für beiden Maschinenachsen des Achspaares zurückgesetzt ist.

24.4 Randbedingungen

24.4.1 Achsen

Gleicher Achstyp

Beide Maschinenachsen eines Achspaares müssen vom **gleichen** Achstyp sein:

- Linearachse:
 - MD30300 \$MA_IS_ROT_AX = 0
 - MD30310 \$MA_ROT_IS_MODULO = 0
- Rundachse:
 - MD30300 \$MA_IS_ROT_AX = 1
 - MD30310 \$MA_ROT_IS_MODULO = 0

Modulo Rundachsen

Keine Maschinenachsen eines Achspaares darf eine **Modulo-Rundachse** sein:

- MD30300 \$MA_IS_ROT_AX = 1 (Rundachse)
- MD30310 \$MA_ROT_IS_MODULO = 1 (Fehler: Modulo-Rundachse !)

24.4.2 Achscontainer

Ändert sich die Zuordnung der zu überwachenden Maschinenachsen dynamisch im Ablauf des Fertigungsprozesses, z.B. bei Verwendung von Achscontainern, muss die Funktion vor der Änderung, z.B. der Achscontainer-Drehung, deaktiviert, dann umparametriert und wieder aktiviert werden.

Beispiel

Die Schutzfunktion soll die logischen Maschinenachsen 1 und 13 überwachen. Diese beziehen sich auf die Slots 1 und 2 des Achscontainers CT1. Die zugehörigen realen Maschinenachsen sind AX1 und AX13.

Bei der Achscontainer-Drehung wird der Achscontainer um einen Schritt weitergeschaltet, was zu einem Tausch der realen Maschinenachsen führt.

Die Freifahrtrichtung der Maschinenachse AX13 liegt in positiver Verfahrrichtung. Die Freifahrtrichtung der Maschinenachse AX1 liegt in negativer Verfahrrichtung.

Parametrierung der NC

Logische Maschinenachsen: Achsnummer 1 und 13

- MD10002 \$MN_AXCONF_LOGIC_MACHAX_TAB [0] = "CT1_SL1" (log. Masch.Achse 1)
- MD10002 \$MN_AXCONF_LOGIC_MACHAX_TAB [12] = "CT1_SL2" (log. Masch.Achse 13)

Achscontainer CT1, Slot 1 und Slot 2

- MD12750 \$MN_AXCT_NAME_TAB[0] = "CT1"
- MD12701 \$MN_AXCT_AXCONF_ASSIGN_TAB1[0] = "AX1" (Slot 1)
- MD12701 \$MN_AXCT_AXCONF_ASSIGN_TAB1[1] = "AX13" (Slot 2)

Reale Maschinenachsen: Maschinenachsennamen AX1 und AX13

- MD10000 \$MN_AXCONF_MACHAX_NAME_TAB[x] = "AX1"
- MD10000 \$MN_AXCONF_MACHAX_NAME_TAB[y] = "AX13"

Parametrierung der Funktion "Achspaar-Kollisionsschutz" vor der Achscontainer-Drehung

- MD61516 \$MN_CC_PROTECT_PAIRS[0] = 01 13
- MD61517 \$MN_CC_PROTECT_SAVE_DIR[0] = 01 00

Achscontainer-Drehung durchführen

1. Deaktivierung der Schutzfunktion
MD61516 \$MN_CC_PROTECT_PAIRS[0] = 00 00
2. Zur Übernahme der Maschinendatenänderung Reset im 1. Kanal der NC auslösen.
MD60972 \$MN_CC_ACTIVE_IN_CHAN_PROT[0], BITx, ...
3. Achscontainer-Drehung ausführen
AXCTSWED(CT1)

Neuparametrierung der Funktion "Achspaar-Kollisionsschutz" nach der Achscontainer-Drehung:

- MD61516 \$MN_CC_PROTECT_PAIRS[0] = 13 01
- MD61517 \$MN_CC_PROTECT_SAFE_DIR[0] = 01 00

oder

- MD61516 \$MN_CC_PROTECT_PAIRS[0] = 01 13
- MD61517 \$MN_CC_PROTECT_SAFE_DIR[0] = 00 01

Zur Übernahme der Maschinendatenänderung "Reset" im 1. Kanal der NC auslösen.

24.4.3 Link-Achsen

Sind die Achsen eines Achspaares Link-Achsen, d.h. über die Funktion "NCU-Link" werden die Sollwerte der Maschinenachsen von Kanälen verschiedener NCUs erzeugt, können die Achsen gegeneinander weder überwacht noch geschützt werden.

24.4.4 Interpolatorische Kopplungen

Annahme

1. Eine Maschinenachse ist Bestandteil einer interpolatorischen Kopplung z. B. :
 - generische Kopplung (CP)
 - Mitschleppen (TRAIL)
 - Leitwertkopplung (LEAD)
 - elektronisches Getriebe (EG)
 - Synchronspindel (COUP)
2. Die Maschinenachse wird **nicht** im **ersten** Kanal der NC verfahren.
3. Die Maschinenachse wird durch die Funktion "Achspaar-Kollisionsschutz" überwacht.

Auswirkung

Wird die Maschinenachse nicht im ersten Kanal der NC verfahren, stehen die von der interpolatorischen Kopplung erzeugten Anteile für den Positions- und Geschwindigkeitssollwert erst nach einer **Totzeit** von **einem Interpolatortakt** der Funktion "Achspaar-Kollisionsschutz" zur Verfügung. Die Überwachung der Maschinenachsen erfolgt

dadurch um diese Anteile versetzt. Der Betrag der Anteile ist dabei abhängig vom Interpolatortakt und der aktuellen Geschwindigkeit und Beschleunigung der Maschinenachse.

24.5 Beispiele

24.5.1 Kollisionsschutz

Das Bild zeigt die Anordnung der 3 Maschinenachsen und die Verschiebung und Orientierung der Maschinenkoordinatensysteme (MKS).

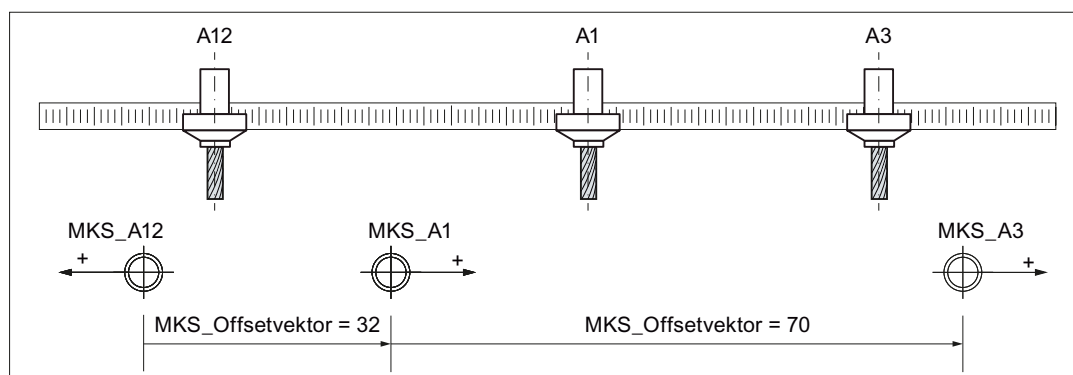


Bild 24-2 Kollisionsschutz für 2 Achspare

Parametrierung: Schutzfunktion 1

Achspaar: 1. Maschinenachse A3, 2. Maschinenachse A1

- MD61516 \$MN_CC_PROTECT_PAIRS[0] = 01 03

Freifahrrichtung: A1 in negativer Richtung, A3 in positiver Richtung

- MD61517 \$MN_CC_PROTECT_SAFE_DIR[0] = 00 01

Offsetvektor von Maschinenkoordinatensystem MKS_A1 nach MKS_A3 bezogen auf MKS_A3

- MD61518 \$MN_CC_PROTECT_OFFSET[0] = 70.0

Schutzfenster beispielhaft 10.0 mm

- MD61519 \$MN_CC_PROTECT_WINDOW[0] = 10.0

Orientierung Maschinenkoordinatensysteme zueinander: gleichsinnig

- MD61532 \$MN_CC_PROTECT_DIR_IS_REVERSE[0] = 0

Schutzfenster-Erweiterung: keine

- MD61533 \$MN_CC_PROTECT_WINDOW_EXTENSION[0] = 0.0

Parametrierung: Schutzfunktion 2

Achspaar: 1. Maschinenachse A1, 2. Maschinenachse A12

- MD61516 \$MN_CC_PROTECT_PAIRS[1] = 12 01

Freifahrtrichtung: A12 in positiver Richtung, A1 in positiver Richtung

- MD61517 \$MN_CC_PROTECT_SAFE_DIR[1] = 01 01

Offsetvektor von Maschinenkoordinatensysteme MKS_A12 nach MKS_A1 bezogen auf MKS_A1

- MD61518 \$MN_CC_PROTECT_OFFSET[1] = 32.0

Schutzfenster beispielhaft 5.0 mm

- MD61519 \$MN_CC_PROTECT_WINDOW[1] = 5.0

Orientierung Maschinenkoordinatensysteme zueinander: gegensinnig

- MD61532 \$MN_CC_PROTECT_DIR_IS_REVERSE[1] = 1

Schutzfenster-Erweiterung: um 5.0 mm auf insgesamt 10.0 mm

- MD61533 \$MN_CC_PROTECT_WINDOW_EXTENSION[1] = 5.0

24.5.2 Kollisionsschutz und Abstandsbegrenzung

Das Bild zeigt die Anordnung der beiden Maschinenachsen, die Verschiebung und Orientierung der Maschinenkoordinatensysteme (MKS) sowie den minimalen und maximalen Abstandsvektor.

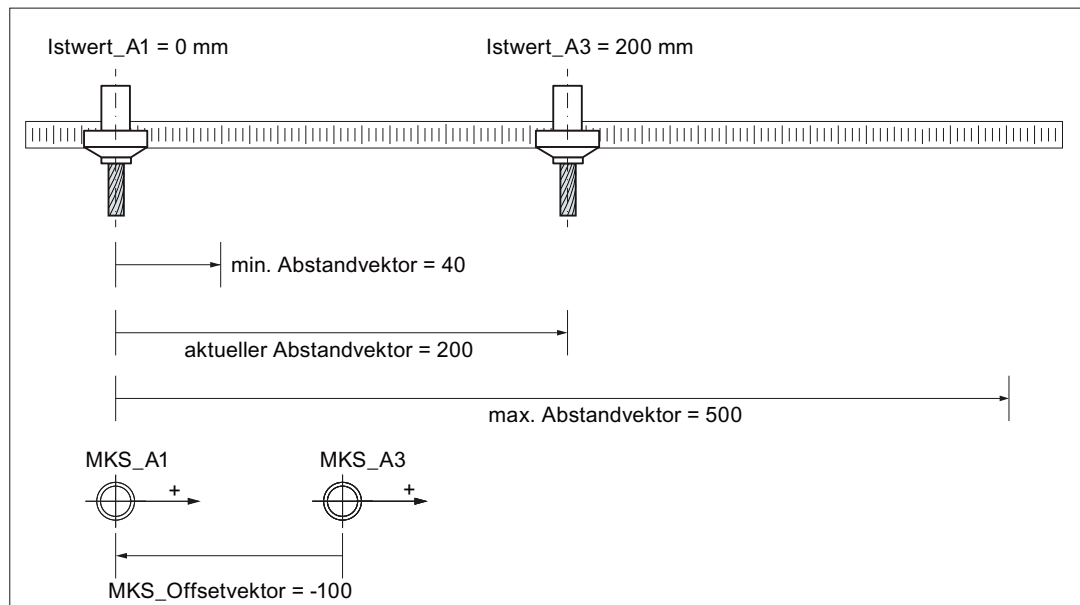


Bild 24-3 Kollisionsschutz und Abstandsbegrenzung für ein Achspaar

Parametrierung: Schutzfunktion 1 - Kollisionsschutz

Achspaar: 1. Maschinenachse A1, 2. Maschinenachse A3

- MD61516 \$MN_CC_PROTECT_PAIRS[0] = 03 01

Freifahrrichtung: A1 in negativer Richtung, A3 in positiver Richtung

- MD61517 \$MN_CC_PROTECT_SAFE_DIR[0] = 01 00

Offsetvektor von Maschinenkoordinatensystem MKS_A3 nach MKS_A1 bezogen auf MKS_A1

- MD61518 \$MN_CC_PROTECT_OFFSET[0] = -100.0

Schutzfenster beispielhaft 40.0 mm

- MD61519 \$MN_CC_PROTECT_WINDOW[0] = 40.0

Orientierung Maschinenkoordinatensysteme zueinander: gleichsinnig

- MD61532 \$MN_CC_PROTECT_DIR_IS_REVERSE[0] = 0

Schutzfenster-Erweiterung: keine

- MD61533 \$MN_CC_PROTECT_WINDOW_EXTENSION[0] = 0.0

Parametrierung: Schutzfunktion 2 - Abstandsbegrenzung

Achspaar: 1. Maschinenachse A1, 2. Maschinenachse A3

- MD61516 \$MN_CC_PROTECT_PAIRS[1] = 03 01

Freifahrrichtung: A1 in positiver Richtung, A3 in negativer Richtung

- MD61517 \$MN_CC_PROTECT_SAFE_DIR[1] = 00 01

Offsetvektor = "Offsetvektor Maschinenkoordinatensystem MKS_A3 nach MKS_A1 bezogen auf MKS_A1" - "maximaler Abstandsvektor bezogen auf MKS_A1"

Hinweis**Maximaler Abstandsvektor**

Der maximale Abstandsvektor von der 1. Maschinenachse zu 2. Maschinenachse ist der Vektor vom Ursprung des Maschinenkoordinatensystems der 1. Maschinenachse zur maximal erlaubten Position der 2. Maschinenachse bezogen auf das Maschinenkoordinatensystems der 1. Maschinenachse.

- MD61518 \$MN_CC_PROTECT_OFFSET[1] = -100.0 - 500.0 = 400.0

Schutzfenster beispielhaft 20.0 mm

- MD61519 \$MN_CC_PROTECT_WINDOW[1] = 20.0

Orientierung Maschinenkoordinatensysteme zueinander: gleichsinnig

- MD61532 \$MN_CC_PROTECT_DIR_IS_REVERSE[1] = 0

Schutzfenster-Erweiterung: keine

- MD61533 \$MN_CC_PROTECT_WINDOW_EXTENSION[1] = 0.0

Steht die Maschinenachse A1 auf 0, wird mit den oben gemachten Einstellungen der Verfahrbereich der Maschinenachse A3 auf den Bereich -60.0 bis 380.0 bezogen auf MKS_A3 beschränkt.

24.6 Datenlisten

24.6.1 Optionsdaten

Nummer	Bezeichner: \$ON_	Beschreibung
19610	TECHNO_EXTENSION_MASK[6]	Freigabe der Technologiefunktion über BIT4 = 1

24.6.2 Maschinendaten

24.6.2.1 NC-spezifische Maschinendaten

Nummer	Bezeichner: \$MN_	Beschreibung
60972	CC_ACTIVE_IN_CHAN_PROT[0]	Kanalspezifische Aktivierung der Technologiefunktion
61516	CC_PROTECT_PAIRS[<a>]	Achspaarspezifische Definition der beiden Maschinenachsen
61517	CC_PROTECT_SAFE_DIR[<a>]	Achspaarspezifische Definition der Freifahrrichtung
61518	CC_PROTECT_OFFSET[<a>]	Achspaarspezifische Definition der Verschiebung der beiden Maschinenkoordinatensysteme
61519	CC_PROTECT_WINDOW[<a>]	Achspaarspezifische Definition der Schutzfenster bzw. Mindestabstand
61532	CC_PROTECT_DIR_IS_REVERSE[<a>]	Achspaarspezifische Definition der Orientierung der Maschinenkoordinatensysteme der beiden Maschinenachsen zueinander
61533	CC_PROTECT_WINDOW_EXTENSION[<a>]	Achspaarspezifische Definition der Schutzfenster-Erweiterung
61534	CC_PROTECT_A_DBD_INDEX	Index der Bremsschnittstelle innerhalb des Systemvariablenfeldes \$A_DBD
61535	CC_PROTECT_OPTIONS[<a>]	Achspaarspezifische Aktivierung der Zusatzfunktionen

24.6.2.2 Achs/Spindel-spezifische Maschinendaten

Nummer	Bezeichner: \$MA_	Beschreibung
61514	CC_PROTECT_ACCEL	Schutzfunktions-spezifische Bremsbeschleunigung

24.6.3 Signale

24.6.3.1 Signale an Achse/Spindel

Signalname	SINUMERIK 840D sl	SINUMERIK 828D
Kollisionsschutz einschalten	DB31,DBX24.3	-

24.6.4 Anwenderdaten

24.6.4.1 Globale Anwenderdaten (GUD)

Bezeichner	Beschreibung
_PROTECT_STATUS[n]	Status der Schutzfunktion (optional)

V2: Vorverarbeitung

25.1 Kurzbeschreibung

Vorverarbeitung

Die in den Verzeichnissen für Standard- und Anwenderzyklen befindlichen Programme können zur schnellen Abarbeitung vorverarbeitet werden.

Die Vorverarbeitung wird über Maschinendatum aktiviert.

Die Standard- und Anwenderzyklen werden bei Power On vorverarbeitet, d. h. das Teileprogramm wird in einen bearbeitungsoptimalen binären Zwischencode steuerungsintern übersetzt (kompiliert).

Alle Programmfehler, die mit Korrektursatz korrigiert werden können, werden bereits zum Zeitpunkt der Vorverarbeitung erkannt. Zusätzlich wird bei Verwendung von Sprüngen und Kontrollstrukturen überprüft, ob die Sprungziele vorhanden sind und ob die Schachtelung von Kontrollstrukturen korrekt ist.

Es steht die volle Steuerungsfunktionalität zur Verfügung:

- Override-Beeinflussung
- Reaktionen auf Daten und Signale, die von der PLC oder Bedienung vorgegeben werden
- Aktuelle Satzanzeige
- Die Programme können im Einzelsatz (SBL1 und SBL2) bearbeitet werden. Satzsuchlauf ist möglich. Das Kompilat kann nicht archiviert werden, es ist dem Anwender verborgen und wird bei Power On jeweils neu erzeugt.

Die Vorverarbeitung ist geeignet für:

- Laufzeitoptimierung von Teileprogrammen mit Hochsprachenanteilen (Sprüngen, Kontrollstrukturen, Bewegungssynchronaktionen)
- Rechenintensive Teileprogramme (z. B. Abspannzyklen)
- Zeitkritische Stellen (z. B. die Programmfortsetzung nach Vorlaufstopp bei schnellem Restweglöschen oder Rückhub oder im Werkzeugwechselzyklus) werden schneller bearbeitet.

Allgemeines

Die Vorverarbeitung von Standard- und Anwenderzyklen ist möglich. Damit kann die Bearbeitungszeit von Teileprogrammen ohne Einschränkung der Steuerungsfunktionalität reduziert werden.

Die Standard- und Anwenderzyklen werden vorverarbeitet, bei entsprechender Setzung des Maschinendatums:

MD10700 \$MN_PREPROCESSING_LEVEL (Programmvorverarbeitungsstufe)

Die Vorverarbeitung erfolgt programmspezifisch. Die Mischung von vorverarbeiteten und im ASCII-Format interpretierten Teileprogrammen ist möglich. Die Vorverarbeitung dient zur Verkürzung von Nebenzeiten.

Für die Vorverarbeitung von Zyklen wird Speicherplatz benötigt. Zur besseren Speicherausnutzung haben Sie zwei Möglichkeiten:

- Mit dem Befehl `DISPLOF` (Anzeige aus) wird das ausführbare Programm verkürzt.
- Das MD10700 `$MN_PREPROCESSING_LEVEL` wurde um Bit 2 und 3 erweitert. Dadurch kann die Zyklen-Vorverarbeitung gezielt für einzelne Verzeichnisse angewählt werden (z. B. Anwenderzyklen).

Das MD10700 `$MN_PREPROCESSING_LEVEL` wurde erweitert um Bit 4. Damit kann die Vorverarbeitung der Anwenderzyklen aus dem Directory `_N_CMA_DIR` angewählt werden.

Das MD10700 `$MN_PREPROCESSING_LEVEL` wurde erweitert um Bit 5. Damit können einzelne Anwenderzyklen gezielt vorverarbeitet werden, welche den Befehl `PREPRO` nach der `PROC`-Anweisung haben.

Vorübersetzte Zyklen werden standardmäßig im dynamischen NC-Speicher abgelegt. Das MD10700 `$MN_PREPROCESSING_LEVEL` wurde erweitert um Bit 6. Damit kann festgelegt werden, dass die jetzt im dynamischen NC-Speicher abgelegten kompilierten Programme, die dort nicht mehr genug Platz haben, im statischen NC-Speicher abgelegt werden dürfen.

Funktionalität

Die in den Verzeichnissen für Standard- und Anwenderzyklen befindlichen Programme werden bei Power On vorverarbeitet, d. h. das Teileprogramm wird in einen bearbeitungsoptimalen binären Zwischencode übersetzt (kompiliert). Beim Aufruf wird dann das Kompilat bearbeitet.

Laufzeitoptimierung

Die Vorverarbeitung ist vor allem zur Laufzeitoptimierung von Teileprogrammen mit Hochsprachenanteilen (Sprüngen, Kontrollstrukturen, Bewegungssynchronaktionen) geeignet.

Während bei der standardmäßig aktiven Interpretation des ASCII-Teileprogramms Sprünge und Kontrollstrukturen durch Suchen in allen Sätzen (Satzanfang) aufgelöst werden, erfolgt beim vorverarbeiteten Teileprogramm der Sprung direkt auf den Zielsatz.

Somit sind die Laufzeitunterschiede zwischen Sprüngen und Kontrollstrukturen aufgehoben.

Beispiel für Laufzeit einer Vorverarbeitung:

Laufzeitreduzierung um 30 % bei aktivem Kompressor

```
DEF INT ZAEHLER
Ziel: G1 G91 COMPON
G1 X0.001 Y0.001 Z0.001 F100000
ZAEHLER=ZAEHLER +1
ZAEHLER=ZAEHLER -1
ZAEHLER=ZAEHLER +1
IF ZAEHLER<= 100000 GOTOB ZIEL
```

Rechenintensive Programme sowie Programme mit symbolischen Namen werden schneller bearbeitet.

Laufzeitkritische Stellen (z. B. die Fortsetzung der Bearbeitung nach Restweglöschen oder Vorlaufstopp in Zyklen) können schneller bearbeitet werden.

Wenn die Interruptroutine als vorverarbeiteter Zyklus vorliegt, kann die Bearbeitung nach der Programmunterbrechung schneller fortgesetzt werden.

25.2 Programmhandling

Aktivierung/Deaktivierung

Die Vorverarbeitung von Zyklen erfolgt bei Power On, wenn das folgende Maschinendatum gesetzt ist:

MD10700 \$MN_PREPROCESSING_LEVEL, Bit1 (Programmvorverarbeitungsstufe)

Bit	Wert	Bedeutung
0		Keine Vorverarbeitung
	0	Aufrufbeschreibung von Zyklen ist nicht standardmäßig bekannt. Zyklen müssen wie normale Unterprogramme vor dem Zyklenaufwurf als Extern erklärt werden. Das ist eine sinnvolle Einstellung, wenn keine Zyklen mit Aufrufparametern verwendet werden.
	1	Im Steuerungshochlauf wird die Aufrufbeschreibung der Zyklen gebildet. Alle Anwenderzyklen (Directory _N_CUS_DIR) und Siemenszyklen (Directory _N_CST_DIR) mit Übergabeparametern können ohne Extern-Erklärung aufgerufen werden. Änderungen an der Zyklen-Aufrufchnittstelle werden erst mit Power On wirksam. Es sind folgende Maschinendaten einzustellen: MD18170 \$MN_MM_NUM_MAX_FUNC_NAMES (Anzahl von Zusatzaktionen) MD18180 \$MN_MM_NUM_MAX_FUNC_PARAM (Anzahl von zusätzlichen Parametern für Zyklen)
1	1	Im Steuerungshochlauf werden alle Zyklen in ein bearbeitungsoptimales Kompilat vorverarbeitet. Alle Anwender-Zyklen (Directory _N_CUS_DIR) und Standard-Zyklen (Directory _N_CST_DIR) werden schnell abgearbeitet. Änderungen an den Zyklen-Programmen werden erst beim nächsten Power On wirksam.
2	1	Im Steuerungshochlauf werden die Standardzyklen aus dem Verzeichnis _N_CST_DIR in ein bearbeitungsoptimales Kompilat vorverarbeitet.
3	1	Im Steuerungshochlauf werden die Anwenderzyklen aus dem Verzeichnis _N_CUS_DIR in ein bearbeitungsoptimales Kompilat vorverarbeitet.
4	1	Vorverarbeitung der Anwenderzyklen aus dem Directory _N_CMA_DIR

5	1	Vorverarbeitung der Anwenderzyklen mit dem Befehl <code>PREPRO</code> in der PROC-Anweisungszeile. Nicht gekennzeichnete Dateien der durch Bit 1-4 bezeichneten Verzeichnisse werden nicht vorverarbeitet. Ist das Bit 0, dann erfolgt die Steuerung des Vorverarbeitens ausschließlich nach den Vorgaben der Bits 0-4.
6	0	Das Kompilat wird im dynamischen NC-Speicher abgelegt, solange noch freier Speicher vorhanden ist. Reicht dieser nicht, so wird die Vorverarbeitung abgebrochen. Bemessung des dynamischen NC-Speichers mit dem Maschinendatum: MD18351 \$MN_MM_DRAM_FILE_MEM_SIZE.

Die durch das Kompilat im dynamischen NC-Speicher belegten Bereiche sind für den Anwender sichtbar.

Kombinationen der Bits sind zulässig.

Kompilieren

Die in den Directories Standard-Zyklen: `_N_CST_DIR`, `_N_CMA_DIR` und Anwender-Zyklen: `_N_CUS_DIR` befindlichen Unterprogramme (Extension `_SPF`) und ggf. die mit `PREPRO` gekennzeichneten Unterprogramme werden kompiliert. Das Kompilat hat den Namen des Original-Zyklus mit Extension `_CYC`.

Hinweis

Programmänderungen an vorkompilierten Programmen werden erst nach dem nächsten Power On wirksam!

Zugriffsrecht

Das vorverarbeitete Programm ist nur ausführbar, nicht les- und schreibbar. Das Kompilat kann nicht verändert oder archiviert werden. Die Original-Zyklen `_SPF` Dateien bleiben erhalten.

Bei Änderung des ASCII-Zyklus wird das Kompilat nicht verändert, d. h. Änderungen werden erst nach dem nächsten Power On wirksam.

Speicherbedarf

Der Speicherbedarf für kompilierte Zyklen ist etwa Faktor 2 zusätzlich zu dem ASCII-Teilprogramm.

Der Speicherbedarf für die in den Teilprogrammen definierte Variable wird über die folgenden bestehenden Maschinendaten festgelegt:

MD28020 \$MC_MM_NUM_LUD_NAMES_TOTAL (Anzahl der lokalen Anwendervariablen)

MD28010 \$MC_MM_NUM_REORG_LUD_MODULES (Anzahl der Bausteine für lokale Anwendervariablen bei REORG)

MD28040 \$MC_MM_LUD_VALUES_MEM (Speichergröße für lokale Anwendervariablen)

MD18242 \$MC_MM_MAX_SIZE_OF_LUD_VALUE (Speicherblockgröße für LUD-/GUD-Werte)

Literatur:

Funktionshandbuch Erweiterungsfunktionen; Speicherkonfiguration (S7)

Der Speicherbedarf zum Zeitpunkt der Vorverarbeitung ist so groß, als würde das vorverarbeitete Programm in der ersten Unterprogrammebene aufgerufen.

Zum Zeitpunkt der Vorverarbeitung bei Power On wird für jedes Sprungziel/Label sowie für jedes Kontrollstrukturelement ein Name wie für eine Variable gezählt und muss in dem folgenden Maschinendatum berücksichtigt werden:

MD28020 \$MC_MM_NUM_LUD_NAMES_TOTAL (Anzahl der lokalen Anwendervariablen)

Beispiel:

Programmcode	Kommentar
PROC NAMES	; 1 Name
DEF INT VARIABLE, FELD[2]	; 2 Namen
ANFANG:	; 1 Name, nur für Vorverarbeitung
FOR VARIABLE = 1 TO 9	; 1 Name, nur für Vorverarbeitung
G1 F10 X=VARIABLE*10-56/86EX4+4*SIN(VARIABLE/3)	
ENDFOR	; 1 Name, nur für Vorverarbeitung
M17	

Um dieses Programm normal abzuarbeiten, muss das folgende Maschinendatum mindestens 3 Namen vereinbaren:

MD28020 \$MC_MM_NUM_LUD_NAMES_TOTAL

Um dieses Programm bei Power On zu kompilieren, sind 6 Namen notwendig.

Vorverarbeitete Programme/Zyklen werden im dynamischen NC-Speicher abgelegt. Der Platzbedarf je Programm ist unverändert wie oben skizziert zu überschlagen. Abstimmung auf die Platzbelegung im statischen NC-Speicher ist nur dann erforderlich, wenn Bit 6 = 1 im folgenden Maschinendatum gesetzt ist:

MD10700 \$MN_PREPROCESSING_LEVEL (Programmverarbeitungsstufe)

In diesem Falle werden die Kompilate der Programme, die im dynamischen NC-Speicher keinen Platz mehr vorfinden, im statischen NC-Speicher untergebracht.

Beispiele für entsprechende Maschineneinstellungen finden Sie unter "Beispiele" im Kapitel "Vorverarbeitung im dynamischen NC-Speicher".

25.3 Programmaufruf

Übersicht

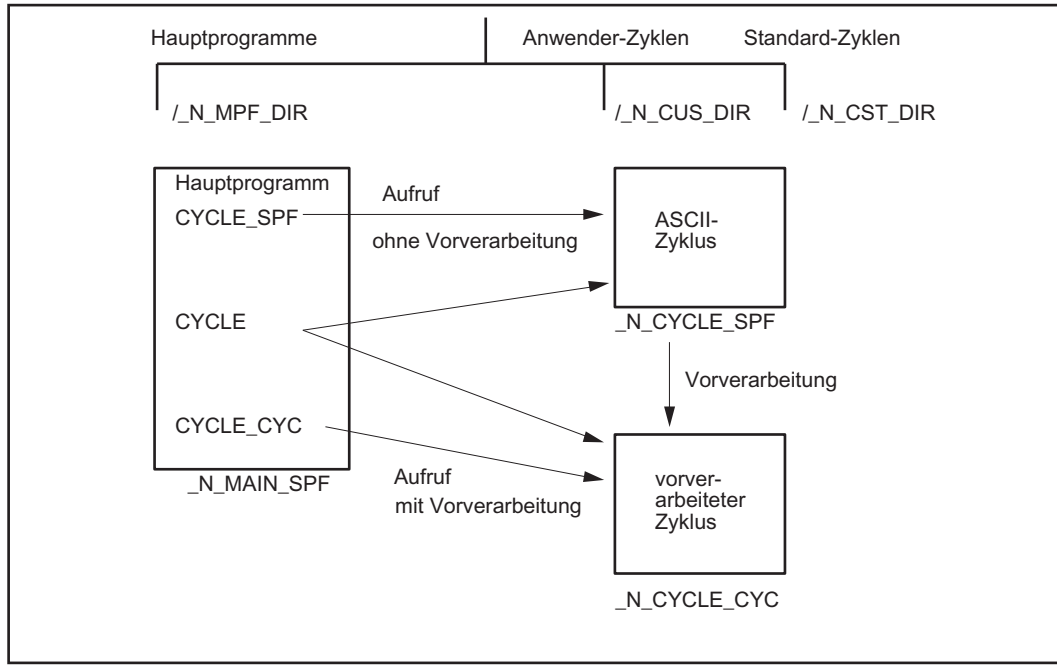


Bild 25-1 Erzeugung und Aufruf vorverarbeiteter Zyklen ohne Parameter

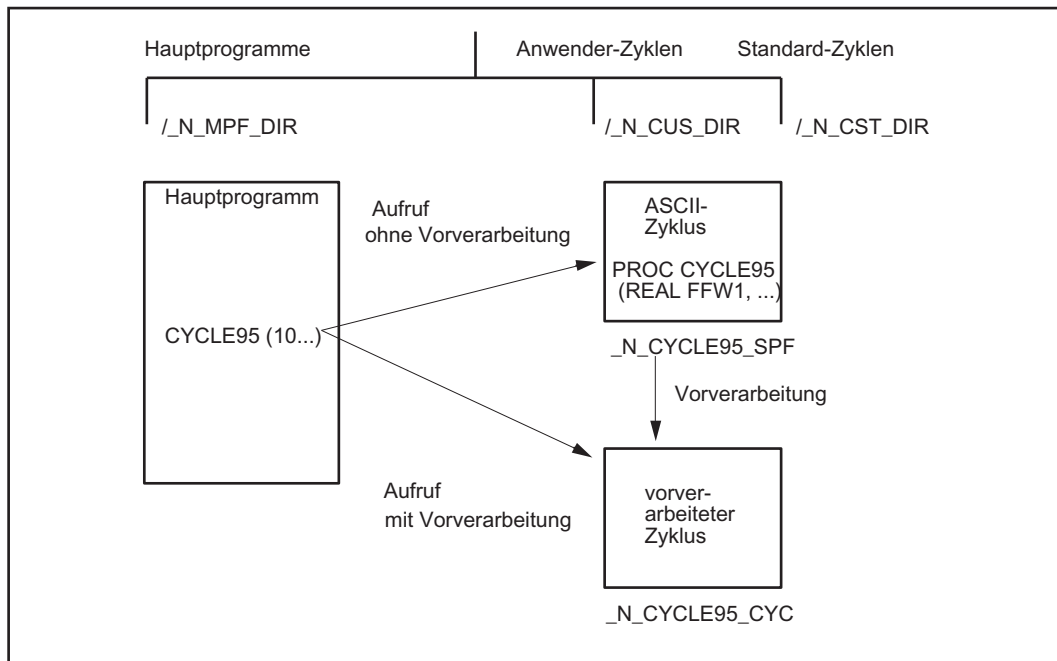


Bild 25-2 Erzeugung und Aufruf vorverarbeiteter Zyklen mit Parameter

Aufruf

- Kompilierter Zyklus: Der Aufruf des kompilierten Zyklus unterscheidet sich nicht vom normalen Unterprogrammaufruf.
Beispiel: CYCLE
- Vorverarbeitung ist aktiviert: Anstelle des ASCII-Zyklus wird der kompilierte Zyklus aufgerufen.
 - Erfolgt der Unterprogrammaufruf explizit mit der Extension `_SPF`, so wird der ASCII-Zyklus aufgerufen, auch wenn ein Kompilat vorhanden ist.
Beispiel: `CYCLE_SPF`; Aufruf des ASCII-Zyklus
 - Erfolgt der Unterprogrammaufruf explizit mit der Extension `_CYC`, so wird der vorverarbeitete Zyklus, sofern vorhanden, aufgerufen. Ist kein Kompilat vorhanden erfolgt Fehlermeldung.
Beispiel: `CYCLE_CYC`; Aufruf des vorkompilierten Zyklus
 - Wird bei aktiviertem Bit 5 eine Datei, die nicht mit `PREPRO` gekennzeichnet ist, explizit mit der Extension `_CYC` aufgerufen, so erfolgt eine Fehlermeldung mit Alarm 14011.
- Wenn ein Unterprogramm aufgerufen wird ohne explizite Extension, so wird zuerst versucht, das kompilierte Programm zu laden, wenn dies nicht möglich ist (wenn nicht mit `PREPRO` gekennzeichnet), so wird versucht das SPF-Programm zu laden.
- Der Wechsel in den externen Sprachmodus durch `G291` wird mit Alarm abgelehnt. Beim Aufruf eines vorkompilierten Zyklus wird explizit in den Siemens-Sprachmodus gewechselt.
- Beim Unterprogrammaufruf wird überprüft, ob das Kompilat älter ist als der Zyklus. Wenn dies der Fall ist, so wird das Kompilat gelöscht und ein Alarm abgesetzt, so dass der Anwender die Zyklen neu vorverarbeiten lassen muss.

Hinweis

Der Aufruf von Zyklen mit der Extension `_SPF` oder `_CYC` ist nur für Zyklen ohne Parameter zulässig

In Zyklen die vorverarbeitet werden, dürfen keine PUDs verwendet werden. Die PUDs werden im aufrufenden Hauptprogramm angelegt. Zum Übersetzungszeitpunkt nach Power On sind diese Daten in den Zyklen nicht bekannt.

In der aktuellen Programmanzeige ist erkennbar, ob der aktuelle ASCII-Zyklus oder das Kompilat aufgerufen wurde (Extension `_SPF` oder `_CYC`).

Aufrufbedingung

Wenn Vorverarbeitung aktiviert ist, müssen alle Zyklen in den Zyklendirectories kompiliert sein. Nicht kompilierte Zyklen in `_N_CUS_DIR` und `_N_CST_DIR`, die z. B. erst nach Power On eingespielt wurden, können nur mit expliziter Angabe Extension `_SPF` aufgerufen werden.

Bei aktiver Vorverarbeitung und Bit 5 werden alle Programme, die nicht mit der Proc-Anweisung `PREPRO` beginnen, nicht vorübersetzt.

Syntax-Check

Alle Programmfehler, die mit Korrektursatz korrigiert werden können, werden bereits zum Zeitpunkt der Vorverarbeitung erkannt. Zusätzlich wird bei Verwendung von Sprüngen und Kontrollstrukturen überprüft, ob die Sprungziele vorhanden sind und ob die Schachtelung von Kontrollstrukturen korrekt ist.

Sprungziele/Labels müssen im Programm eindeutig sein.

Nach Korrektur der bei der Vorverarbeitung erkannten Fehler muss die Vorverarbeitung durch NC Power On erneut gestartet werden.

25.4 Randbedingungen

Verfügbarkeit der Funktion "Vorverarbeitung"

Die Funktion ist eine Option ("Programmvorverarbeitung"), die über das Lizenzmanagement der Hardware zugeordnet werden muss.

Sprachumfang

Es steht der volle Sprachumfang der NC-Sprache im Teileprogramm zur Verfügung.

Die Verrechnung gemessener Prozessgrößen und die Reaktion auf Signale aus dem Prozess und aus anderen Kanälen (Override, Restweglöschen, Bewegungssynchronaktionen, Kanalkoordinierung, Interruptbearbeitung usw.) ist uneingeschränkt möglich.

Achsname

Das Kompilieren von Teileprogrammen erfolgt kanalunabhängig. Deshalb müssen die über die folgenden Maschinendaten eingestellten Geometrie- und Kanalnamen in allen Kanälen **gleich** sein, wenn sie in den **vorkompilierten Zyklen direkt verwendet werden**:

MD20060 \$MC_AXCONF_GEOAX_NAME_TAB (Geometrieachsname im Kanal)

MD20080 \$MC_AXCONF_CHANAX_NAME_TAB (Kanalachsname im Kanal)

Typischerweise werden in Bearbeitungszyklen Achsnamen nicht direkt verwendet, da Zyklen folgendermaßen geschrieben werden:

- kanalunabhängig und
- unabhängig von den an der Maschine definierten Achsnamen.

Die zu verfahrenen Achsen werden indirekt über Maschinendaten angesprochen oder als Parameter übergeben:

- Indirekte Achsprogrammierung:
 - IF \$AA_IM[AXNAME(\$MC_AXCONF_CHANAX_NAME_TAB[4])] > 5
; Dieser Zweig wird durchlaufen, wenn der Istwert der 5. Kanalachse
; bezogen auf das Maschinenkoordinatensystem größer als 5 ist.
 - G1 AX[AXNAME(\$MC-AXCONF-GEOAX-NAME-TAB[0])] = 10
F1000 G90.
; Verfahre die 1. Geometrieachse auf den Wert 10.
ENDIF
- Übergabe der zu verfahrenen Achse aus dem Hauptprogramm:
 - Zyklendefinition
PROC BOHRE(AXIS BOHRACHSE)
WHILE \$AA_IW[BOHRACHSE] > -10
G1 G91 F250 AX[BOHRACHSE] = -1
ENDWHILE
 - Aufruf aus dem Hauptprogramm
BOHRE(Z)

25.5 Beispiele

25.5.1 Vorverarbeitung einzelner Dateien

Programmcode	Kommentar
PROC UP1 PREPRO	; Vorverarbeitung, wenn Bit 5 = 1
	; in PREPROCESSING_LEVEL
N1000 DEF INT ZAEHLER	
N1010 ZIEL: G1 G91 COMPON	
N1020 G1 X0.001 Y0.001 Z0.001 F100000	
N1030 ZAEHLER=ZAEHLER+1	
N1040 ZAEHLER=ZAEHLER-1	
N1050 ZAEHLER=ZAEHLER+1	
N1060 IF ZAEHLER <=10 GOTOB ZIEL	
N1070 M30	

Programmcode	Kommentar
PROC UP2	
N2000 DEF INT VARIABLE, FELD[2]	
N2010 IF \$AN_NCK_Version < 3.4	
N2020 SETAL(61000)	
N2030 ENDIF	
N2040 ANFANG:	
N2050 FOR VARIABLE = 1 TO 5	
N2060 G1 F1000 X=VARIABLE*10-56/86EX4+4*SIN(VARIABLE/3)	
N2070 ENDFOR	
N2080 M17	
PROC MAIN	
N10 G0 X0 Y0 Z0	
N20 UP1	
N30 G0 X10 Y10 Z10	
N40 UP2	
N50 G0 X100 Y100	
N60 UP3	
N70 G0 X10 Y10	
N80 M30	

Beispielkonstellationen:

a) Bit 5 = 1

MD10700 \$MN_PREPROCESSING_LEVEL=45 ; Bit 0, 2, 3, 5

Unterprogramm UP1 wird vorübersetzt, die Aufrufbeschreibung wird gebildet.

Unterprogramm UP2 wird nicht vorübersetzt, die Aufrufbeschreibung wird aber gebildet.

b) Bit 5 = 0

MD10700 \$MN_PREPROCESSING_LEVEL=13 ; Bit 0, 2, 3,

Beide Unterprogramme werden vorübersetzt, die Aufrufbeschreibung wird gebildet.

c) Beispiel für ein **ungültiges** Unterprogramm bei aktiviertem Kompilieren:

Programmcode	Kommentar
PROC SUB1 PREPRO	
G291	; ← Alarm beim Kompilieren, G291 nicht möglich
G0 X0 Y0 Z0	
M17	

25.5.2 Vorverarbeitung im dynamischen NC-Speicher

Maschinendaten für Vorverarbeitung nur im dynamischen NC-Speicher mit selektiver Auswahl:

Programmcode	Kommentar
	; Bit 5 =1 Selektive Programmauswahl
	; Bit 6 =0 Kein Ausweichen auf
	; statischen NC-Speicher, wenn
	; dynamischer NC-Speicher voll
N30 \$MN_MM_DRAM_FILE_MEM_SIZE = 800	; Platz vorhalten
N40 \$MN_PREPROCESSING_LEVEL = 63	; Bit 0-5 =1
M17	

Maschinendaten für Vorverarbeitung im dynamischen NC-Speicher mit Option, den statischen NC-Speicher zu nutzen und mit selektiver Auswahl:

Programmcode	Kommentar
	; Bit 5 =1 Selektive Programmauswahl
	; Bit 6 =1 Ausweichen auf statischen
	; NC-Speicher, wenn dynamischer NC-
	; Speicher voll
N30 \$MN_MM_DRAM_FILE_MEM_SIZE = 800	; Platz vorhalten
N40 \$MN_PREPROCESSING_LEVEL = 127	; Bit 0-6 = 1
M17	

25.6 Datenlisten

25.6.1 Maschinendaten

25.6.1.1 Allgemeine Maschinendaten

Nummer	Bezeichner: \$MN_	Beschreibung
10700	PREPROCESSING_LEVEL	Programmvorverarbeitungsstufe
18242	MM_MAX_SIZE_OF_LUD_VALUE	Maximale Feldgröße der LUD-Variablen

25.6.1.2 Kanal-spezifische Maschinendaten

Nummer	Bezeichner: \$MC_	Beschreibung
28010	MM_NUM_REORG_LUD_MODULES	Anzahl der Bausteine für lokale Anwendervariablen bei REORG (DRAM)
28020	MM_NUM_LUD_NAMES_PER_PROG	Anzahl der lokalen Anwendervariablen (DRAM)
28040	MM_LUD_VALUES_MEM	Speichergröße für lokale Anwendervariablen (DRAM)

W5: 3D-Werkzeugradiuskorrektur - nur 840D sl

26.1 Funktion

26.1.1 Einleitung

3D-Umfangsfräsen und 3D-Stirnfräsen

Die 3D-Werkzeugradiuskorrektur (3D-WRK) dient zur Bearbeitung von Konturen mit Werkzeugen, deren Orientierung unabhängig von der Werkzeugbahn und der Werkzeugform beeinflusst werden kann.

SINUMERIK stellt die 3D-WRK in verschiedenen Varianten zur Verfügung, die in Kombination mit folgenden Fertigungsverfahren zum Einsatz kommen:

- Umfangsfräsen (Seite 924) bei Strukturbauteilen
- Stirnfräsen (Seite 930) bei Freiformflächen

Werkzeugorientierung

Bei der 3D-WRK sind die beiden Fälle eines Werkzeugs mit raumfester Orientierung und eines Werkzeugs mit veränderlicher Orientierung zu unterscheiden.

Die Werkzeugradiuskorrektur mit raumfestem Werkzeug beim Umfangsfräsen wird als 2½D-WRK bezeichnet. Dies ist die herkömmliche Werkzeugradiuskorrektur. Umfangsfräsen mit einer über diesen Leistungsumfang hinausgehenden Funktionalität ist bei der Korrektur mit raumfestem Werkzeug nicht möglich. Beim Umfangsfräsen gibt es also nur dann eine 3D-WRK, wenn die Werkzeugorientierung veränderbar ist. D. h., es werden zusätzlich zu den 3 Freiheitsgraden zur Positionierung des Werkzeugs (in der Regel 3 Linearachsen) zwei weitere Freiheitsgrade (2 Rundachsen) zur Einstellung der Werkzeugorientierung benötigt (5-Achsbearbeitung).

Beim Stirnfräsen kann die Orientierung des Werkzeugs konstant bleiben oder veränderlich sein. Die Vorteile der veränderbaren Orientierung sind:

- bessere Approximation der Endkontur
- höhere Spanleistung
- größere Freiheit bei Auswahl der Werkzeugform
- größere Bandbreite bearbeitbarer Oberflächen (Hinterschnitte)

Hinweis

5D-WRK

Die 3D-WRK wird mitunter auch als 5D-WRK bezeichnet, da in diesem Fall 5 Freiheitsgrade für die Festlegung der Lage des Werkzeugs im Raum relativ zum Werkstück zur Verfügung stehen.

Parametrierung

Die für die 2D-WRK eingestellten Maschinen- und Settingdaten sind auch für die 3D-WRK wirksam.

Literatur:

Funktionshandbuch Grundfunktionen; Kapitel "W1: Werkzeugkorrektur"

Daneben gibt es spezielle, nur für die 3D-WRK relevante Systemdaten. Zur Einstellung dieser Daten siehe Kapitel "Parametrierung (Seite 939)".

Aktivierung/Deaktivierung

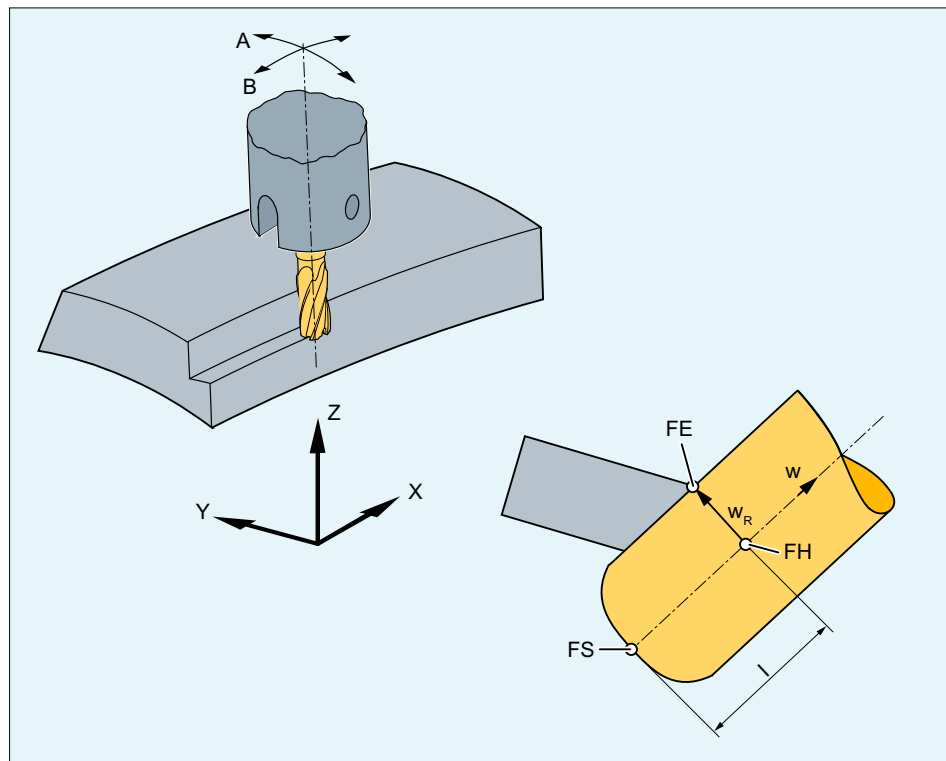
Die Auswahl der 3D-WRK-Variante erfolgt im NC-Programm über G-Befehle der Gruppe 22 (Werkzeugkorrekturtyp). Die eigentliche Aktivierung und die Deaktivierung erfolgen über die G-Befehle der Gruppe 7 (Werkzeugradiuskorrektur).

Siehe Kapitel "Programmierung (Seite 940)".

26.1.2 Umfangsfräsen

Das 3D-Umfangsfräsen dient zur Bearbeitung von Konturen mit Werkzeugen, deren Orientierung bei der Bearbeitung nicht konstant ist. Die Bearbeitung erfolgt dabei mit der Mantelfläche des Werkzeugs. Ist das Werkzeug zylindrisch, werden bei dieser Bearbeitung Regelflächen (z. B. Kegel, Zylinder, etc.) erzeugt. Es können jedoch auch andere Fräserformen (z. B. auch Formfräser) eingesetzt werden. Die für diese Bearbeitungsart notwendige Werkzeugradiuskorrektur ist damit eine Erweiterung der konventionellen 2½D-WRK.

Die Verhältnisse beim 3D-Umfangsfräsen sind in folgendem Bild dargestellt:



FE: Fräserbearbeitungspunkt

FS: Fräuserspitze

FH: Fräserhilfspunkt

l: Eintauchtiefe

w: Werkzeugvektor

w_r : Vektor vom Fräserhilfspunkt zum Fräserbearbeitungspunkt mit Länge des Schafradius R

Bild 26-1 Umfangsfräsen

Eintauchtiefe

Die Eintauchtiefe des Fräses ist der Abstand des Fräserhilfspunkts von der Werkzeugspitze.

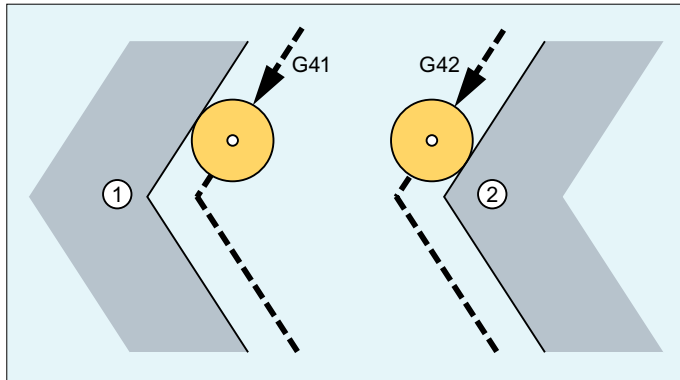
Der Fräserhilfspunkt ist die senkrechte Projektion des Fräserbearbeitungspunkts auf der programmierten Bahn auf die Werkzeuglängsachse.

Mit der Eintauchtiefe wird somit die Lage des Bearbeitungspunkts auf der Mantelfläche des Werkzeugs eingestellt.

Über die Adresse ISD (InSertion Depth) kann beim Umfangsfräsen und **aktiver** 3D-Werkzeugradiuskorrektur die Eintauchtiefe des Werkzeugs geändert werden (siehe Kapitel "Programmierung (Seite 940)").

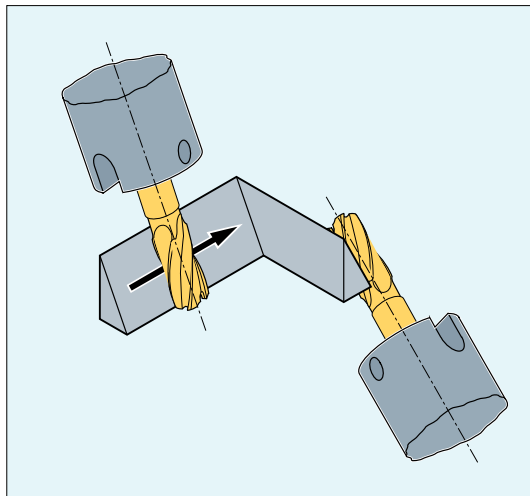
26.1.2.1 Ecken beim Umfangfräsen

Außen- und Innenecken werden getrennt behandelt. Die Bezeichnung Innen- oder Außenecke ist abhängig von der Werkzeugorientierung:



- ① Innenecke
- ② Außenecke

Bei Orientierungsänderungen an einer Ecke kann der Fall auftreten, dass sich der Eckentyp während der Bearbeitung ändert:



Tritt dieser Fall auf, wird die Bearbeitung mit der Alarmmeldung 10770 abgebrochen.

26.1.2.2 Verhalten an Außenecken

An Außenecken werden beim Umfangsfräsen mit 3D-WRK analog zu den Verhältnissen bei der 2½D-WRK die G-Befehle der Gruppe 18 (Eckenverhalten Werkzeugkorrektur) ausgewertet:

- G450: Übergangskreis (Werkzeug umfährt Werkstückecken auf einer Kreisbahn)
Außenecken werden als Kreise mit dem Radius 0 behandelt, wobei die Kreisebene von der Endtangente des ersten und der Starttangente des zweiten Satzes aufgespannt wird. Damit ist auch eine Orientierungsänderung beim Satzübergang möglich. An einer Außenecke wird somit immer ein Satz eingefügt.
Im Gegensatz zur Lösung bei der 2½D-WRK ist das eingefügte Konturelement an einer Außenecke immer ein Kreis mit dem Radius 0, auf den die Werkzeugradiuskorrektur so wirkt wie auf jede andere programmierte Bahn auch. Es ist nicht möglich, an Stelle der Kreise Kegelschnitte einzufügen. Die Adresse DISC hat deshalb in diesem Fall keine Bedeutung und wird nicht ausgewertet.
- G451: Schnittpunkt der Äquidistanten (Werkzeug schneidet in der Werkstückecke frei)
Der Schnittpunkt wird bestimmt, indem die Offsetkurven der beiden beteiligten Sätze verlängert werden und deren Schnittpunkt in der Ebene senkrecht zur Werkzeugorientierung an der Ecke bestimmt wird.
Wird ein Schnittpunkt in der Ebene senkrecht zum Werkzeug gefunden, bedeutet das nicht, dass sich die Kurven auch im Raum schneiden. Vielmehr werden die Kurven in Richtung der Werkzeuglängsachse betrachtet, die in der Regel einen bestimmten Abstand voneinander haben. Die Positionsdifferenz wird über die gesamte Satzlänge in Werkzeugrichtung herausgefahren.
Die Behandlung dieses Offsets in Werkzeugrichtung an Außenecken entspricht der an Innenecken.

Bei nahezu tangentialen Übergängen ist das Verhalten auch bei aktivem G450 wie bei G451 (Grenzwinkel ist über MD20210 \$MC_CUTCOM_CORNER_LIMIT einstellbar). Umgekehrt wird auch bei aktivem G451 ein Kreis eingefügt (Verhalten wie bei G450), wenn kein Schnittpunkt existiert bzw. wenn der Winkel an der Ecke einen bestimmten Wert übersteigt (MD20230 \$MC_CUTCOM_CURVE_INSERT_LIMIT).

Verhalten bei Orientierungsänderungen

Das Schnittpunktverfahren (G451) wird nicht verwendet, wenn zwischen den beteiligten Verfahrätzen mindestens ein Satz eingefügt ist, der eine Änderung der Werkzeugorientierung enthält. In einem solchen Fall wird an der Ecke immer ein Kreis eingefügt.

Über die G-Befehle der Gruppe 27 (Werkzeugkorrektur bei Orientierungsänderung an Außenecken) kann festgelegt werden, ob Orientierungsänderungen, die zwischen den beiden die Ecke bildenden Verfahrätzen programmiert wurden, vor Beginn des eingefügten Kreissatzes oder gleichzeitig mit diesem ausgeführt werden:

- ORIC: Orientierungsänderungen an Außenecken werden dem einzufügenden Kreissatz überlagert.
- ORID: Orientierungsänderungen werden vor dem Kreissatz ausgeführt.

Sind zwischen zwei Verfahrätzen zwei oder mehr Sätze mit Orientierungsänderungen (z. B. A2=... B2=... C2=...) programmiert und ist ORIC aktiv, so wird der einzufügende Kreissatz entsprechend dem Betrag der einzelnen Winkeländerungen auf diese Zwischensätze aufgeteilt. Weitere programmierte Zwischensätze ohne Verfahr- und Orientierungsbewegungen werden an den programmierten Stellen ausgeführt.

26.1 Funktion

Ist ORID aktiv, werden am Ende des ersten der beiden Verfahrssätze alle eingefügten Sätze (sowohl solche mit als auch solche ohne Orientierungsbewegung) ausgeführt. Für die Offsetberechnung wird dabei die Tangente im Endpunkt des ersten Verfahrssatzes verwendet. Der Kreissatz mit konstanter Orientierung wird unmittelbar vor dem zweiten Verfahrssatz eingefügt.

Siehe auch "Beispiel 1: Orientierungsänderung an Außenecke beim 3D-Umfangsfräsen (Seite 956)".

Zwischensätze ohne relevante Verfahrinformationen

Zwischensätze ohne relevante Verfahrinformationen (weder die Werkzeugorientierung noch die Position der Geometrieachsen werden verändert) sind zulässig. Das Schnittpunktverfahren wird auf die benachbarten Sätze so angewendet, als ob diese Zwischensätze nicht vorhanden wären. Ebenso sind in Zwischensätzen Bewegungen in Werkzeugrichtung zulässig.

26.1.2.3 Verhalten an Innenecken

Bei der 3D-WRK werden für die Schnittpunktberechnung immer nur benachbarte Verfahrssätze betrachtet.

Wird die Orientierung an der Satzgrenze nicht verändert, so muss die Kontur nur in der Ebene senkrecht zur Werkzeugachse betrachtet werden. Der Werkzeugquerschnitt ist dann ein Kreis, der die beiden Konturen berührt. Die geometrischen Verhältnisse in dieser Ebene sind identisch mit denen bei der 2¹/₂D-WRK.

Verhalten bei Orientierungsänderungen

Wird die Orientierung am Satzübergang geändert, wird das Werkzeug in der Innenecke so abgewälzt, dass es zu jedem Zeitpunkt die beiden Sätze, die die Innenecke bilden, berührt.

Bahnstücke müssen hinreichend lang sein, damit bei einer Orientierungsänderung an einer Innenecke die Berührungspunkte des Werkzeugs nicht über die Satzgrenze hinweg auf andere Sätze wandern.

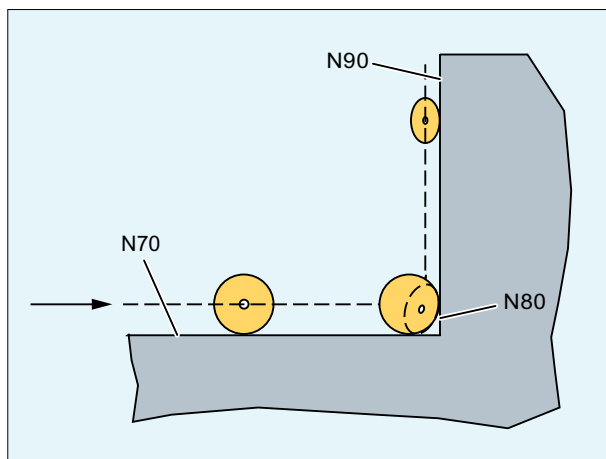


Bild 26-2 Orientierungsänderung an Innenecke

Bei einer Änderung der Orientierung in einem Satz, der mit einem anderen Satz eine Innenecke bildet, muss von dem programmierten Zusammenhang zwischen Bahnposition und zugehöriger Orientierung abgewichen werden, da die Bahnendposition nicht erreicht wird, die Orientierung aber ihren Endwert erreichen muss. Dieses Verhalten ist völlig analog zum Verhalten von Synchronachsen bei der 2¹/₂D-WRK.

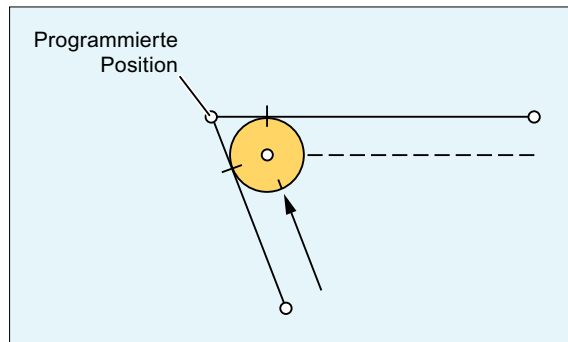


Bild 26-3 Bahnendposition und Orientierungsänderung an Innenecken

Siehe auch "Beispiel 2: Orientierungsänderung an Innenecke beim 3D-Umfangsfräsen (Seite 958)".

Änderung der Eintauchtiefe

Im allgemeinen Fall liegen die Konturelemente, die eine Innenecke bilden, nicht in der Ebene senkrecht zum Werkzeug, so dass die Berührungspunkte der beiden Sätze am Werkzeug unterschiedliche Abstände zur Werkzeugspitze haben.

Daraus folgt: Die Eintauchtiefe (ISD) ändert sich an einer Innenecke sprunghaft vom 1. zum 2. Satz.

Damit diese Tiefendifferenz nicht sprunghaft auftritt, wird sie bei der Interpolation auf die beteiligten Sätze kontinuierlich verteilt. Die Ausgleichsbewegung für die Tiefe erfolgt in der aktuellen Werkzeugrichtung.

Mit dieser Lösung treten bei zylindrischen Werkzeugen keine Konturverletzungen auf, wenn das Werkzeug so lang ist, dass der Fräseingriffspunkt auf dem Fräsermantel den Bereich, in dem eine Bearbeitung möglich ist, nicht verlässt.

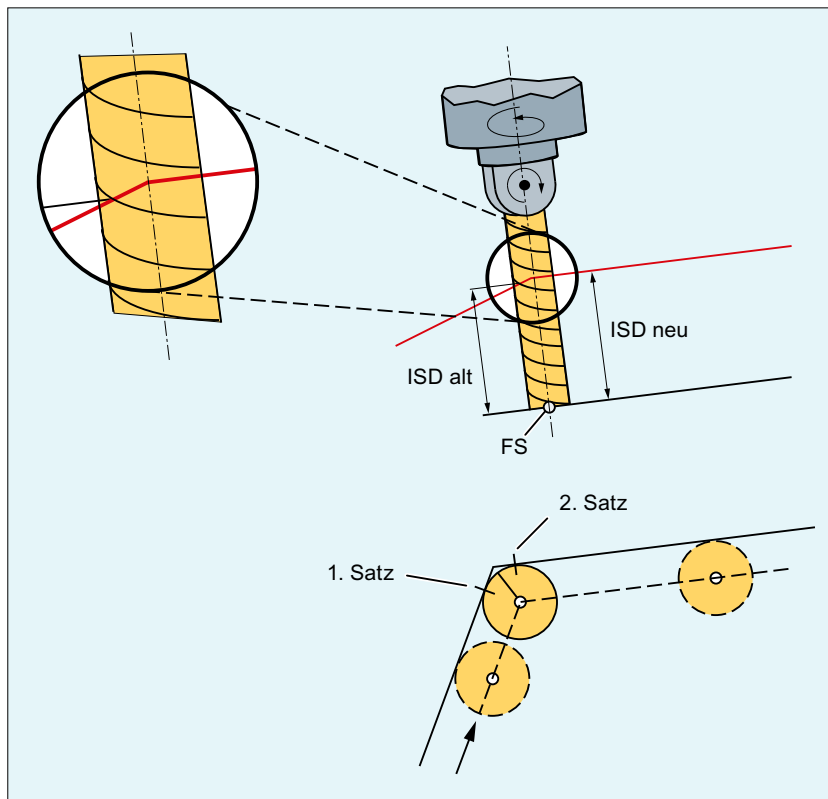


Bild 26-4 Änderung der Eintauchtiefe

26.1.2.4 Überwachung der Bahnkrümmung

Die Bahnkrümmung wird bei unveränderlicher Orientierung in der folgenden Weise überwacht:

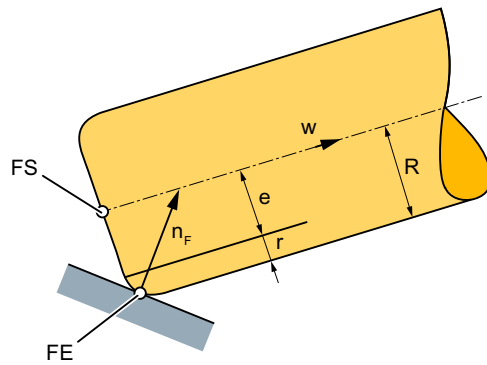
1. Die Kontur in jedem Satz wird auf die Ebene projiziert, die orthogonal zur Werkzeugorientierung liegt.
2. Dann wird die Krümmung der Projektion analysiert. Bereiche mit zu hoher Krümmung (Werkzeug passt nicht in die Kurve) werden ausgeschnitten.
3. Die restlichen Offsetkonturen überschneiden sich und bilden so die globale Offsetkontur.

26.1.3 Stirnfräsen

Das Stirnfräsen dient zur Bearbeitung beliebig gekrümmter Oberflächen. Dabei sind Werkzeuglängsachse und Flächennormalenvektor mehr oder weniger parallel, während beim 3D-Umfangsfräsen die Werkzeuglängsachse und der Flächennormalenvektor der zu bearbeitenden Fläche aufeinander senkrecht stehen.

Für das Stirnfräsen ist die Information über die zu bearbeitende Fläche zwingend erforderlich, d. h. die Beschreibung der linienhaften Bahn im Raum allein reicht nicht aus. Für die Werkzeugkorrektur (der Begriff "Werkzeugradiuskorrektur" ist hier nicht mehr angemessen) beim Stirnfräsen muss außerdem die Werkzeugform bekannt sein.

Die Verhältnisse beim Stirnfräsen sind im folgenden Bild dargestellt:

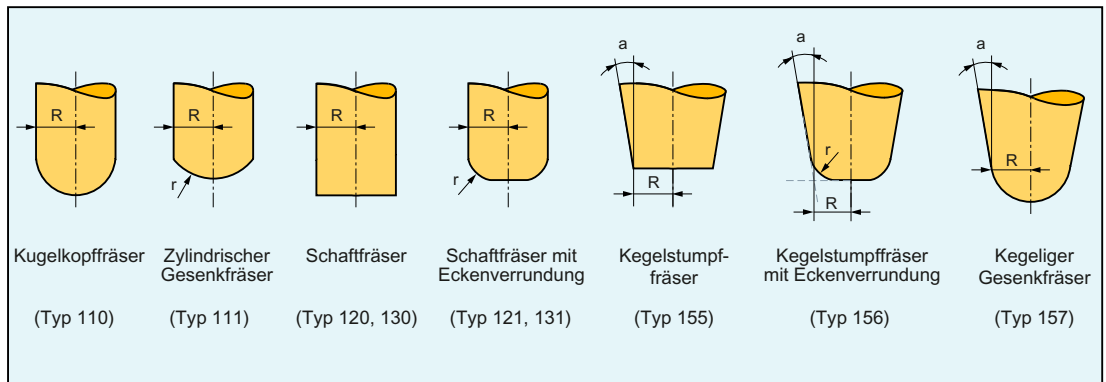


- FE: Fräserbearbeitungspunkt
- FS: Fräterspitze
- n_F : Flächennormalenvektor
- w: Werkzeugvektor
- R: Schaftradius
- r: Eckenradius
- e: Differenz zwischen Schaftradius R und Eckenradius r

Bild 26-5 Stirnfräsen mit einem Torusfräser

26.1.3.1 Werkzeugformen und Werkzeugdaten für Stirnfräsen

Im Folgenden sind die für Stirnfräsen möglichen Werkzeugformen und relevanten Werkzeugdaten zusammengestellt. Die Form des Werkzeugschafts wird nicht berücksichtigt. Deshalb sind z. B. die Werkzeugtypen 120 und 155 in ihrer Wirkung identisch.



- R = Schaftradius (Werkzeugradius)
- r = Eckenradius
- a = Winkel zwischen Werkzeuglängsachse und oberem Ende der Torusfläche

Fräsertyp	Typ-Nr.	R	r	a
Kugelkopffräser	110	> 0	-	-
Zylindrischer Gesenkfräser	111	> 0	> R	-
Schaftfräser, Winkelkopffräser	120, 130	> 0	-	-
Schaftfräser, Winkelkopffräser mit Eckenverrundung	121, 131	> r	> 0	-

26.1 Funktion

Fräsertyp	Typ-Nr.	R	r	a
Kegelstumpffräser	155	> 0	-	> 0
Kegelstumpffräser mit Eckenverrundung	156	> 0	> 0	> 0
Kegeliger Gesenkfräser	157	> 0	-	> 0

- : wird nicht ausgewertet

Werkzeugdaten	Werkzeugparameter	
Werkzeugmaße	Geometrie	Verschleiß
R	\$TC_DP6	\$TC_DP15
r	\$TC_DP7	\$TC_DP16
a	\$TC_DP11	\$TC_DP20

Hinweis

Wird im NC-Programm eine andere als in der Abbildung gezeigte Typ-Nummer angegeben, verwendet das System automatisch den Werkzeugtyp 110 (Kugelkopffräser).

Hinweis

Bei Verletzung der Grenzwerte für die Werkzeugdaten wird ein Alarm ausgegeben.

Geometrie- und Verschleißwerte eines Werkzeugdatums wirken additiv, d. h. die tatsächlich zur Korrekturberechnung verwendete Werkzeugform wird immer durch die Summenwerte beschrieben.

In der Funktion CUT3DFD (Differenzwerkzeug) wird nur der Verschleißwert als Korrekturwert betrachtet. Die Geometriewerte sind in diesem Fall nur zur Beschreibung der Werkzeugform notwendig.

Werkzeuglängenkorrektur

Als Bezugspunkt für die Längenkorrektur gilt die Werkzeugspitze (Schnittpunkt zwischen Werkzeuglängsachse und Werkstückoberfläche).

Werkzeugwechsel

Ein neues Werkzeug mit veränderten Abmessungen (R, r, a) oder anderer Form darf nur bei der erstmaligen Aktivierung der Werkzeugkorrektur (d. h. beim Übergang von G40 nach G41 oder G42) bzw. bei bereits aktiver Korrektur nur bei erneuter Programmierung von G41 bzw. G42 angegeben werden. Alle anderen Werkzeugdaten, z. B. Werkzeuglängen, bleiben von dieser Regel unberücksichtigt, sodass solche Werkzeuge auch ohne erneutes G41 bzw. G42 eingewechselt werden können.

26.1.3.2 Stirnfräsen mit Angabe des Flächennormalenvektors

Für das 3-D-Stirnfräsen sind erweiterte Möglichkeiten zur Orientierungsprogrammierung vorgesehen.

Um die Werkzeugkorrektur für das Stirnfräsen berechnen zu können, reicht die Angabe der Bahn (d. h. einer Linie im Raum) nicht aus. Es muss zusätzlich die zu bearbeitende Fläche

bekannt sein. Die notwendige Information über die Fläche wird der Steuerung mit dem Flächennormalenvektor zur Verfügung gestellt.

Der Flächennormalenvektor am Satzanfang wird mit A4, B4, C4, der am Satzende mit A5, B5, C5 programmiert. Komponenten der Flächennormalenvektoren, die nicht programmiert sind, werden auf null gesetzt. Die Länge eines derart programmierten Vektors hat keine Bedeutung. Ein Vektor der Länge Null (alle drei Komponenten Null) wird ignoriert, d. h., eine vorher programmierte Richtung bleibt gültig, es wird kein Alarm erzeugt.

Wird in einem Satz nur der Startvektor programmiert (A4, B4, C4), so bedeutet das, dass der programmierte Flächennormalenvektor während des ganzen Satzes konstant bleibt. Wird nur der Endvektor programmiert (A5, B5, C5), so wird vom Endwert des vorherigen Satzes mittels Großkreisinterpolation zum programmierten Endwert interpoliert. Sind sowohl Start- als auch Endvektor programmiert, so wird zwischen beiden Richtungen ebenfalls mittels Großkreisinterpolation interpoliert. Die Tatsache, dass der Startvektor in einem Satz neu programmiert werden darf, bedeutet, dass sich die Richtung des Flächennormalenvektors an einem Satzübergang unstetig ändern kann. Unstetige Übergänge des Flächennormalenvektors treten immer dann auf, wenn die an einem Satzübergang beteiligten Flächen (Ebenen) nicht tangential ineinander übergehen, d. h., wenn sie eine Kante bilden.

Ein einmal programmierter Flächennormalenvektor bleibt so lange gültig, bis ein neuer Vektor programmiert wird. Bei Grundstellung wird der Flächennormalenvektor gleich dem Vektor in z-Richtung gesetzt. Diese Grundstellungsrichtung ist unabhängig von der aktiven Ebene (G17 - G19). Ist ORIWKS aktiv, beziehen sich Flächennormalenvektoren auf den aktiven Frame, d. h., bei einer Framedrehung werden diese mitgedreht. Dies gilt sowohl für programmierte Orientierungen als auch für solche, die aus der aktiven Ebene abgeleitet wurden. Ist ORIWKS aktiv, werden bei Framewechsel die Flächennormalenvektoren mitgeführt. Beim Umschalten von ORIWKS nach ORIMKS wird eine durch Framedrehungen modifizierte Orientierung nicht wieder rückgängig gemacht.

Es muss beachtet werden, dass programmierte und intern verwendete Flächennormalenvektoren voneinander abweichen können. Dies ist immer dann der Fall, wenn der programmierte Flächennormalenvektor nicht senkrecht auf der Bahntangente steht. Es wird dann ein neuer Flächennormalenvektor gebildet, der in der von Bahntangente und programmiertem Flächennormalenvektor aufgespannten Ebene liegt, der jedoch auf dem Bahntangentenvektor senkrecht steht. Diese Orthogonalisierung ist erforderlich, da Bahntangentenvektor und Flächennormalenvektor bei einer realen Fläche immer aufeinander senkrecht stehen (müssen). Da die beiden Größen jedoch unabhängig voneinander programmiert werden können, können sie Informationen enthalten, die sich gegenseitig widersprechen. Die Orthogonalisierung bewirkt, dass in diesem Fall der im Bahntangentenvektor enthaltenen Information der Vorzug vor der im Flächennormalenvektor enthaltenen Information gegeben wird. Es wird ein Alarm ausgegeben, wenn der Winkel zwischen Bahntangentenvektor und programmiertem Flächennormalenvektor den Grenzwert unterschreitet, der im folgenden Maschinendatum festgelegt ist:

MD21084 \$MC_CUTCOM_PLANE_PATH_LIMIT (Minimaler Winkel zwischen Flächenormalvektor und Bahntangentenvektor)

Wird ein Satz verkürzt (Innenecke), so wird auch der Interpolationsbereich des Flächennormalenvektors entsprechend reduziert, d. h. anders als bei Interpolationsgrößen wie z. B. der Position einer zusätzlichen Synchronachse wird der Endwert des Flächennormalenvektors dann nicht erreicht.

Neben den bekannten Möglichkeiten zur Orientierungsprogrammierung kann die Werkzeugorientierung mit den beiden Adressen LEAD (Voreil- oder Sturzwinkel) und TILT (Seitwärtswinkel) auch auf den Flächennormalenvektor und Bahntangentenvektor bezogen

werden. Die Interpretation der Winkelangaben ist dabei abhängig von der Einstellung in MD21094 \$MC_ORIPATH_MODE (siehe Kapitel "Bahnrelative Orientierung (ORIPATH, ORIPATHS, ORIOTC) (Seite 101)"). Die Angabe der Winkel relativ zur Flächennormalen ist lediglich eine erweiterte Möglichkeit der Orientierungsprogrammierung am Satzende. Sie impliziert nicht, dass Voreil- und Seitwärtswinkel ihre programmierten Werte bereits vor Erreichen des Bahnendpunkts erreichen.

Aus Bahntangente, Flächennormalenvektor, Voreilwinkel und Seitwärtswinkel am Satzende wird die resultierende Werkzeugorientierung ermittelt. Diese wird am Satzende immer erreicht, insbesondere auch dann, wenn der Satz (an einer Innenecke) verkürzt wird. Ist das entfallende Bahnstück keine Gerade in einer Ebene, weichen im neuen Bahnendpunkt Voreil- und Seitwärtswinkel im Allgemeinen von ihren programmierten Werten ab, denn bei gleicher absoluter Orientierung des Werkzeugs wie im ursprünglichen Bahnendpunkt ist die Orientierung relativ zum Flächennormalenvektor bzw. Bahntangentenvektor verändert.

26.1.3.3 Korrektur auf der Bahn

Beim Stirnfräsen muss der Fall gesondert betrachtet werden, dass der Bearbeitungspunkt auf der Werkzeuoberfläche springt. Dies kann bei einem Torusfräser immer dann der Fall sein, wenn Flächennormalenvektor n_F und Werkzeugvektor w kollinear werden (d. h., das Werkzeug steht exakt senkrecht auf der Oberfläche), da dieser Richtung auf dem Werkzeug nicht ein einzelner Punkt entspricht, sondern die gesamte Kreisfläche an der Werkzeugstirnseite. Der Berührungspunkt ist deshalb bei dieser Orientierung nicht definiert. Ein Bahnpunkt, in dem Werkzeuglängsachse und Flächennormale parallel sind, wird im Folgenden als singularer Punkt oder kurz als Singularität bezeichnet.

Dieser Fall ist auch praktisch von Bedeutung, z. B. dann, wenn mit einem senkrecht stehenden Werkzeug eine konvexe Oberfläche bearbeitet werden soll (z. B. Stirnfräsen mit konstanter Orientierung), deren Flächennormale ebenfalls senkrecht werden kann (z. B. eine Halbkugel). Der Bearbeitungspunkt an der Kontur bleibt dabei fix, die Maschine muss jedoch bewegt werden, um den Bearbeitungspunkt von einer Werkzeugseite zur anderen bringen.

Das beschriebene Problem ist lediglich ein Grenzfall (Voreilwinkel $\beta = 0$ und Seitwärtswinkel $\gamma = 0$). Wird der Voreilwinkel $\beta = 0$ und der Seitwärtswinkel γ hat einen kleinen Wert, so muss das Werkzeug sehr schnell (im Grenzfall sprungartig) bewegt werden, um den durch die Fräsbedingungen erzwungenen Bearbeitungspunkt nahe der die Stirnfläche bildenden Kreislinie einzuhalten (siehe folgendes Bild).

Es wird im Grundsatz durch die Glättung der Flächennormalen und die Glättung der Kontur gelöst.

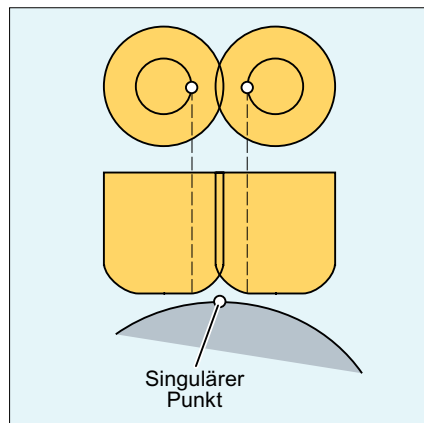


Bild 26-6 Wechsel des Bearbeitungspunkts auf der Werkzeugoberfläche in der Umgebung eines Punkts, in dem Flächennormalenvektor und Werkzeugorientierung parallel sind

Singularitäten können nicht nur in isolierten Punkten auftreten, sondern auch über ganze Kurven. Dieser Fall tritt z. B. (aber nicht ausschließlich) dann auf, wenn es sich bei der zu interpolierenden Kurve um eine ebene Kurve (Kurve, deren Schmiegeebene konstant ist) handelt und das Werkzeug konstant parallel zum Binormalenvektor (senkrecht zur Schmiegeebene) ausgerichtet ist. Ein einfaches Beispiel ist ein Kreisbogen in der X-Y-Ebene, der mit einem Werkzeug bearbeitet wird, das parallel zur Z-Achse ausgerichtet ist. Bei Bahnen dieses Typs reduziert sich die Werkzeugkorrektur auf eine Werkzeuglängenkorrektur. D. h., das Werkzeug wird so geführt, dass seine Spitze auf der programmierten Bahn liegt.

Beim sprunghaften Übergang zwischen singulären und nicht singulären Kurven müssen wie bei der Behandlung einzelner Punkte Linearsätze eingefügt werden, damit der Bearbeitungspunkt auf dem Werkzeug von der Werkzeugspitze zur Peripherie wandern kann (an Außenecken bzw. konvexen Flächen) bzw. die Bahnen müssen verkürzt werden, um Konturverletzungen zu vermeiden (an Innenecken bzw. konkaven Flächen).

Deswegen ist es notwendig, sowohl die Glättung der Konturen (Funktion "Top Surface") als auch die Glättung der Flächennormalen einzuschalten, damit die Übergänge stetig verlaufen.

26.1.3.4 Ecken beim Stirnfräsen

Zwei Flächen, die nicht tangential ineinander übergehen, bilden eine Kante. Die auf den Flächen definierten Bahnen bilden eine Ecke. Diese Ecke ist ein Punkt der Kante.

Der Eckentyp (Innen- oder Außenecke) wird durch die Flächennormalen der beteiligten Flächen und die auf ihnen definierten Bahnen festgelegt.

An einer Kante kann der Fall auftreten, dass die Flächennormalen der beiden beteiligten Flächen in entgegengesetzte Richtungen der Gesamtflächen zeigen (die Vorderseite der einen Fläche wird mit der Rückseite der zweiten Fläche fortgesetzt), siehe auch das folgende Bild. Derartige Übergänge sind nicht erlaubt und werden mit einem Alarm abgelehnt.

Das Skalarprodukt aus Flächennormalenvektor und (gegebenenfalls veränderlicher) Werkzeugorientierung an einer Ecke oder Bahn muss in jedem Punkt positiv sein, d. h. eine Bearbeitung von der Rückseite der Oberfläche ist nicht zulässig. Ist diese Bedingung nicht erfüllt, wird ein Alarm ausgegeben. Die grundsätzlich zulässigen Gültigkeitsbereiche der Werkzeugorientierung bei Innen- und Außenecken sind im folgenden Bild veranschaulicht. Diese Bereiche werden weiter durch die Bedingung eingeschränkt, dass der Winkel zwischen den zu bearbeitenden Flächen und der "steilsten" Mantellinie der Werkzeugoberfläche einen

über ein Maschinendatum festgelegten Wert nicht unterschreiten darf. Die "steilste" Mantellinie ist dabei eine Linie, die um den Winkel α gegen die Werkzeuglängsachse geneigt ist (bei zylindrischen Werkzeugen hat diese Linie die Richtung der Werkzeuglängsachse). Diese Einschränkung ist notwendig, damit der Berührungspunkt auf dem Werkzeug den zulässigen Bereich nicht verlässt.

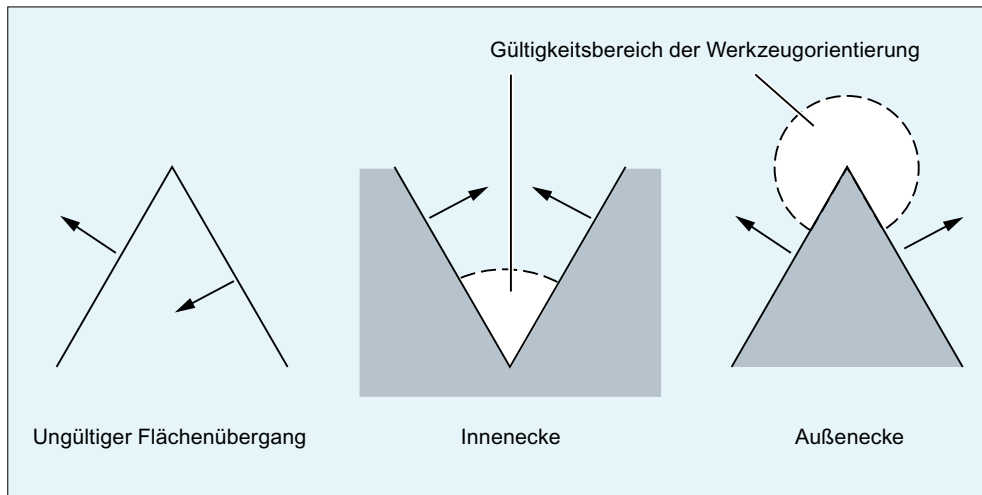


Bild 26-7 Ecken beim Stirnfräsen

Zwischen zwei Sätzen, die eine Bahn enthalten, können Sätze eingefügt sein, die keine Bewegung enthalten (z. B. Hilfsfunktionsausgaben) und/oder die Achsbewegungen der nicht an der Bahn beteiligten Geometrieachsen enthalten. Insbesondere können in solchen Sätzen auch Orientierungsänderungen programmiert sein. Eine Ausnahme bildet dabei das Aktivieren und das Deaktivieren der 3D-Werkzeugradiuskorrektur: Zwischen dem Aktivierungssatz und dem ersten korrigierten Satz bzw. zwischen dem letzten korrigierten Satz und dem Deaktivierungssatz sind Zwischensätze mit Orientierungsänderungen nicht zulässig. Andere Zwischensätze sind jedoch erlaubt.

26.1.3.5 Verhalten an Außenecken

Außenecken werden beim Stirnfräsen als Kreise mit dem Radius 0 behandelt, wobei die Kreisebene von der Endtangente des ersten und der Starttangente des zweiten Satzes aufgespannt wird. Damit ist auch eine Orientierungsänderung beim Satzübergang möglich. An einer Außenecke wird somit immer ein Kreis als Konturelement eingefügt. Das Schnittpunktverfahren steht beim Stirnfräsen nicht zur Verfügung.

Verhalten bei Orientierungsänderungen

Über die G-Befehle der Gruppe 27 (Werkzeugkorrektur bei Orientierungsänderung an Außenecken) kann festgelegt werden, ob Orientierungsänderungen, die zwischen den beiden die Ecke bildenden Verfahrssätzen programmiert wurden, vor Beginn des eingefügten Kreissatzes oder gleichzeitig mit diesem ausgeführt werden:

- ORIC: Orientierungsänderungen an Außenecken werden dem einzufügenden Kreissatz überlagert.
- ORID: Orientierungsänderungen werden vor dem Kreissatz ausgeführt.

Sind zwischen zwei Verfahrssätzen zwei oder mehr Sätze mit Orientierungsänderungen (z. B. A2=... B2=... C2=...) programmiert und ist ORIC aktiv, so wird der einzufügende Kreissatz entsprechend dem Betrag der einzelnen Winkeländerungen auf diese Zwischensätze aufgeteilt. Weitere programmierte Zwischensätze ohne Verfahr- und Orientierungsbewegungen werden an den programmierten Stellen ausgeführt.

Ist ORID aktiv, werden am Ende des ersten der beiden Verfahrssätze alle eingefügten Sätze (sowohl solche mit als auch solche ohne Orientierungsbewegung) ausgeführt. Für die Offsetberechnung wird dabei die Tangente im Endpunkt des ersten Verfahrssatzes verwendet. Der Kreissatz mit konstanter Orientierung wird unmittelbar vor dem zweiten Verfahrssatz eingefügt.

Siehe auch "Beispiel 1: Orientierungsänderung an Außenecke beim 3D-Umfangsfräsen (Seite 956)".

26.1.3.6 Verhalten an Innenecken

An einer Innenecke muss die Lage des Werkzeugs gefunden werden, in der dieses die beiden beteiligten Flächen berührt, wobei die Berührungspunkte auf den auf den beiden Flächen definierten Bahnen liegen sollen. Diese Aufgabe ist im Allgemeinen nicht exakt lösbar, denn führt man das Werkzeug auf der Bahn der ersten Fläche, so wird es die zweite Fläche im Allgemeinen in einem Punkt berühren, der nicht auf der Bahn liegt.

Das Werkzeug wird deshalb nicht auf der Bahn der ersten Fläche geführt, sondern es wird in geeigneter Weise so von der Bahn abgewichen, dass in der Position, in der das Werkzeug beide Flächen berührt, die Abweichung der Berührungspunkte von den jeweiligen Konturen minimal wird, siehe auch das folgende Bild.

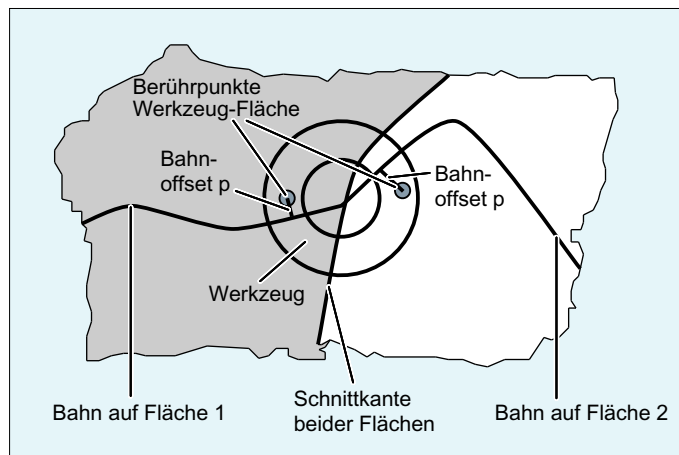


Bild 26-8 Innenecke beim Stirnfräsen (Ansicht in Richtung der Werkzeuglängsachse)

Hinweis

Die Abweichung der Berührungspunkte von der programmierten Kontur wird in der Regel klein sein, da der im Bild zum besseren Verständnis dargestellte Fall, dass der Bearbeitungspunkt an einer Innenecke die Fräserseite "wechselt" (die Winkeldifferenz Ψ um die Werkzeuglängsachse zwischen den beiden Berührungspunkten auf der Werkzeugoberfläche hat ungefähr den Wert 180°), eher die Ausnahme sein wird (siehe auch folgendes Bild, rechts). Im Normalfall wird Ψ nahezu konstant bleiben, so dass sich ein relativ kleiner Abstand der Berührungspunkte auf der Werkzeugoberfläche ergibt (siehe auch folgendes Bild, links).

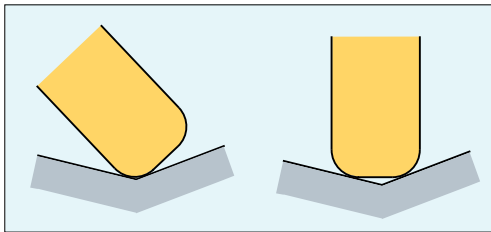


Bild 26-9 Bearbeitung an Innenecken

Die Differenz zwischen programmiertem Punkt auf der Bahn und tatsächlich anzufahrendem Punkt (der Bahnoffset p) wird linear über die gesamte Satzlänge herausgefahren. Differenzen, die sich aus Innenecken an Satzanfang und Satzende ergeben, werden überlagert. Die aktuelle Differenz in einem Bahnpunkt ist dabei immer senkrecht zu Bahn und liegt in der durch den Flächennormalenvektor definierten Oberfläche.

Ist die Werkzeugorientierung an einer Innenecke nicht konstant, wird die Orientierungsänderung analog zur der im Kapitel "Verhalten an Innenecken" für das 3D-Umfangsfräsen dargestellten Art und Weise ausgeführt, d. h. das Werkzeug wird in der Ecke so abgewälzt, dass am Satzanfang, am Satzende und an zwei Punkten bei $\frac{1}{3}$ und bei $\frac{2}{3}$ der Orientierungsänderung die Bedingung erfüllt ist, dass diese die beiden begrenzenden Flächen berührt. Zwischen diesen 4 Punkten wird mit einem Polynom 3. Grades interpoliert.

Eine veränderliche Werkzeugorientierung in einem Satz, der wegen einer Innenecke verkürzt wird, wird ebenfalls in der im Kapitel "Verhalten an Innenecken" für das 3D-Umfangsfräsen (Seite 928) dargestellten Art und Weise behandelt, d. h. die gesamte Orientierungsänderung wird im verkürzten Satz ausgeführt. Damit ändert sich auch der funktionale Zusammenhang zwischen Bahntangente, Flächennormale und Werkzeugorientierung. Dies hat zur Folge, dass in dem verkürzten Satz neue, vorher nicht vorhandene Singularitäten oder unzulässige Seitwärtswinkel (in Punkten, die nahezu singular sind) auftreten können. Tritt bei der Behandlung einer Innenecke eine derartige Situation auf, wird die Bearbeitung mit einem Alarm abgebrochen. Eine Satzaufteilung an den singulären Punkten wird nicht vorgenommen, da die dann erforderlichen Ausgleichbewegungen häufig zu Konturverletzungen führen würden, und der Wechsel der Bearbeitungsseite am Werkzeug vom Anwender in der Regel nicht beabsichtigt und auch nicht vorhersehbar ist. Der Alarm wird bei der Untersuchung einer Innenecke auch dann ausgegeben, wenn die Singularität im zweiten der beiden Sätze auftritt, ohne dass der Übergang zum Folgesatz betrachtet wird. Es wird somit nicht erkannt, wenn ein solcher Satz auch mit seinem Folgesatz eine Innenecke bildet und die Singularität durch die zweite Verkürzung wieder verschwinden würde.

Abhilfe: NC-Programm neu/anders erzeugen, z. B. mit einer kleineren Toleranz.

Der Flächennormalenvektor n_F wird durch die Verkürzung eines Satzes nicht beeinflusst. Das bedeutet, dass anders als bei der Werkzeugorientierung die eventuell auszuführende Orientierungsänderung dieses Vektors nicht auf das verkürzte Verfahrintervall abgebildet wird. Dies ist notwendig, weil andernfalls eine andere als die programmierte Fläche bearbeitet würde. Anders als bei der Werkzeugorientierung ist auch eine sprunghafte Änderung der Flächennormalen an einem Satzübergang unproblematisch, da ihr keine Achsbewegungen entsprechen.

Beim Stirnfräsen muss ebenso wie beim 3D-Umfangsfräsen (siehe Kapitel "Verhalten an Innenecken (Seite 928)") sichergestellt sein, dass in den beiden Verfahrsätzen, die eine Innenecke bilden, Berührungspunkte gefunden werden. Es erfolgt keine Betrachtung über mehrere Verfahrsätze hinweg (d. h. es gibt keine Flaschenhalserkennung), CDON/CDOF wird nicht ausgewertet. Wird kein Berührungspunkt gefunden, wird die Bearbeitung mit einem Alarm (Kollisionsgefahr) abgebrochen.

26.1.3.7 Überwachung der Bahnkrümmung

Die Bahnkrümmung wird nur fürs Umfangsfräsen überwacht. Es wird daher für CUT3DF*-Funktionen in der Regel nicht erkannt, wenn versucht wird, eine konkave Fläche zu bearbeiten, die so stark gekrümmt ist, dass eine Bearbeitung mit dem verwendeten Werkzeug nicht möglich ist. Eine Ausnahme können solche Sätze bilden, die auf Grund einer erkannten Singularität aufgeteilt werden, so dass der Übergang zwischen den beiden daraus entstehenden Teilsätzen wie eine Innenecke behandelt wird. Abgesehen von diesen Sonderfällen liegt es in der Verantwortung des Anwenders, nur solche Werkzeuge zu verwenden, mit denen überall auf der Kontur eine Bearbeitung ohne Konturverletzung möglich ist. Dabei könnte hilfreich sein, die Funktion CUT3DFD zu verwenden, wobei das Differenzwerkzeug sich minimal vom Originalwerkzeug unterscheidet.

26.2 Inbetriebnahme

26.2.1 Parametrierung

Minimaler Winkel zwischen Bahntangente und WZ-Orientierung

Bei der 3D-WRK darf der Winkel zwischen der Bahntangente und der Werkzeugorientierung einen bestimmten Grenzwinkel nicht unterschreiten.

Dieser Winkel wird festgelegt im Maschinendatum:

MD21080 \$MC_CUTCOM_PARALLEL_ORI_LIMIT

Je geringer der Wert von MD21080, umso höher ist im Allgemeinen der Rechenaufwand, der benötigt wird, um die Einhaltung der genannten Bedingungen zu überprüfen. Ausnahmen gelten für Linearsätze mit konstanter Orientierung.

Minimaler Winkel zwischen Flächennormalenvektor und WZ-Orientierung

Das folgende Maschinendatum gibt beim 3D-Stirnfräsen den Winkel an, den Flächennormalenvektor und Werkzeugorientierung in jedem Punkt der Bahn mindestens bilden müssen, wenn mit einem Seitwärtswinkel ungleich null gearbeitet wird und das Werkzeug kein Kugelfräser ist:

MD21082 \$MC_CUTCOM_PLANE_ORI_LIMIT

Beim Unterschreiten dieses Werts wird die Bearbeitung mit einem Alarm abgebrochen.

Je geringer der Wert von MD21082, umso höher ist im Allgemeinen der Rechenaufwand, der benötigt wird, um die Einhaltung der genannten Bedingung zu überprüfen.

MD21082 ist nicht wirksam in Linearsätzen mit konstanter Orientierung. In diesem Fall sind beliebig kleine Winkel zugelassen, auch wenn der Seitwärtswinkel ungleich null ist.

Minimaler Winkel zwischen Flächennormalenvektor und Bahntangentenvektor

Das folgende Maschinendatum gibt beim 3D-Stirnfräsen den Winkel an, den Flächennormalenvektor und Bahntangentenvektor in jedem Punkt der Bahn mindestens bilden müssen:

MD21084 \$MC_CUTCOM_PLANE_PATH_LIMIT

Beim Unterschreiten dieses Werts wird die Bearbeitung mit einem Alarm abgebrochen.

Je geringer der Wert von MD21084, umso höher ist im Allgemeinen der Rechenaufwand, der benötigt wird, um die Einhaltung der genannten Bedingung zu überprüfen.

Interpolation der Flächennormalen über Polynome

Für die Verwendung der "Glättung der Flächennormalen beim 3D-Stirnfräsen" muss die Funktion "Interpolation der Flächennormalen über Polynome" freigeschaltet sein:

MD28291 \$MC_MM_SMOOTH_SURFACE_NORMALS = TRUE

Ist MD28291 nicht gesetzt, wird die Bearbeitung mit einem Alarm abgebrochen.

26.3 Programmierung

26.3.1 3D-Werkzeugradiuskorrektur für das 3D-Umfangsfräsen anwählen (CUT3DC, CUT3DCD, ISD)

Die 3D-Werkzeugradiuskorrektur (3D-WRK) für das 3D-Umfangsfräsen ohne Berücksichtigung von Begrenzungsflächen wird mit dem modal wirksamen G-Befehl CUT3DC bzw. CUT3DCD angewählt.

Die eigentliche Aktivierung erfolgt mit G41 bzw. G42. Ausgeschaltet wird die Werkzeugradiuskorrektur mit G40.

Syntax

```
G41/G42 ORIC/ORID ISD=... CUT3DC/CUT3DCD CDOF2 X... Y... Z...
...
G40 X... Y... Z...
```

Bedeutung

CUT3DC:	3D-WRK für das Umfangsfräsen (nur bei aktiver 5-Achs-Transformation)	
CUT3DCD:	Auf ein Differenzwerkzeug bezogene 3D-WRK für das Umfangsfräsen (nur bei aktiver 5-Achs-Transformation) Die Radiusdifferenz wird durch den WZ-Parameter \$TC_DP15 festgelegt.	
G41/G42 X... Y... Z... :	Werkzeugradiuskorrektur einschalten	
	G41:	Werkzeugradiuskorrektur links von der Kontur
	G42:	Werkzeugradiuskorrektur rechts von der Kontur
	Hinweis: Das Einschalten muss in einem Linearsatz (G0/G1) erfolgen.	
CDOF2:	Kollisionsüberwachung für 3D-Umfangsfräsen ausschalten	
ORIC/ORID:	Über die G-Befehle ORIC und ORID wird das Verhalten bei Orientierungsänderungen an Außenecken festgelegt.	
	ORIC:	Orientierungsänderungen an Außenecken werden dem einzufügenden Kreissatz überlagert.
	ORID:	Orientierungsänderungen an Außenecken werden vor dem einzufügenden Kreissatz ausgeführt.
ISD=<Value>:	Mit der Adresse ISD kann beim Umfangsfräsen und aktiver 3D-Werkzeugradiuskorrektur die Eintauchtiefe des Werkzeugs geändert werden.	
	<Value>:	Länge der Eintauchtiefe
G40 X... Y... Z... :	Werkzeugradiuskorrektur ausschalten Hinweis: Das Ausschalten muss in einem Linearsatz (G0/G1) mit Geometrieachsbewegungen erfolgen.	

Hinweis

Die G-Befehle zur Anwahl der 3D-WRK werden im Anfahrsatz ausgewertet, d. h. typischerweise in dem Satz, der G41 oder G42 enthält.

G41 bzw. G42 kann auch in Sätzen ohne Verfahrbewegung in den für die Korrektur relevanten Geometrieachsen programmiert sein. In diesem Fall ist der Anfahrsatz der erste auf einen solchen Satz folgende Verfahrersatz.

Ein Wechsel der 3D-WRK-Variante bei aktiver Werkzeugradiuskorrektur wird ohne Alarm ignoriert.

Beispiel

Programmcode	Kommentar
	; Definition des Werkzeugs D1:
\$TC_DP1[1,1]=120	; Typ (Schaftfräser)
\$TC_DP3[1,1]=20	; Längenkorrekturvektor
\$TC_DP6[1,1]=8	; Radius
N10 X0 Y0 Z0 T1 D1 F12000	; Anwahl des Werkzeugs.
N20 TRAORI(1)	; Einschalten der Transformation.
N30 G42 ORIC ISD=10 CUT3DC G64 X30	; Aktivieren des 3D-Umfangsfräsens, ; Orientierungsänderungen an Außenecken kontinuierlich, ; Eintauchtiefe: 10mm
N40 ORIWKS A30 B15	; Orientierungsänderung an einer Ecke durch Angabe von Achspositi- onen.
N50 Y20 A3=1 C3=1	; Verfahrssatz mit Orientierungsänderung, ; Angabe der Orientierung mit Richtungsvektor.
N60 X50 Y30	; Verfahrssatz mit konstanter Orientierung.
N70 Y50 A3=0.5 B3=1 C3=5	; Verfahrssatz mit Orientierungsänderung.
N80 M63	; Satz ohne Verfahrinformation.
N90 X0 ISD=20	; Verfahrssatz mit Änderung der Eintauchtiefe.
N100 G40 Y0	; Deaktivieren der Werkzeugradiuskorrektur.
N110 M30	

Weitere Informationen

Bahn und Orientierung

Die hier benutzte Variante des Umfangsfräsens ist durch die Vorgabe einer Bahn (Leitlinie) und der zugehörigen Orientierung realisiert. Bei dieser Art der Bearbeitung ist auf der Bahn die Werkzeugform ohne Bedeutung. Entscheidend ist allein der Radius am Werkzeugeingriffspunkt.

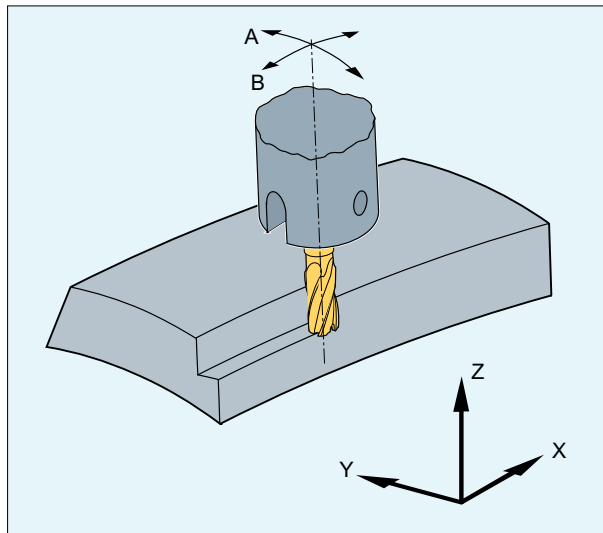


Bild 26-10 Umfangsfräsen

Anfahrverhalten

Das Anfahrverhalten ist bei den 3D-Varianten der Werkzeugradiuskorrektur immer NORM.

Verhalten an Außenecken

An Außenecken werden beim Umfangsfräsen mit 3D-WRK analog zu den Verhältnissen bei der 2½D-WRK die G-Befehle der Gruppe 18 (Eckenverhalten Werkzeugkorrektur) ausgewertet:

- G450: Übergangskreis (Werkzeug umfährt Werkstückecken auf einer Kreisbahn)
Im Gegensatz zur Lösung bei der 2½D-WRK ist das eingefügte Konturelement an einer Außenecke immer ein Kreis mit dem Radius 0, auf den die Werkzeugradiuskorrektur so wirkt wie auf jede andere programmierte Bahn auch. Es ist nicht möglich, an Stelle der Kreise Kegelschnitte einzufügen. Die Adresse DISC hat deshalb in diesem Fall keine Bedeutung und wird nicht ausgewertet.
- G451: Schnittpunkt der Äquidistanten (Werkzeug schneidet in der Werkstückecke frei)
Der Schnittpunkt wird bestimmt, indem die Offsetkurven der beiden beteiligten Sätze verlängert werden und deren Schnittpunkt in der Ebene senkrecht zur Werkzeugorientierung an der Ecke bestimmt wird.

Das Schnittpunktverfahren (G451) wird nicht verwendet, wenn zwischen den beteiligten Verfahransätzen mindestens ein Satz eingefügt ist, der eine Änderung der Werkzeugorientierung enthält. In einem solchen Fall wird an der Ecke immer ein Kreis eingefügt.

Verhalten bei Orientierungsänderungen an Außenecken

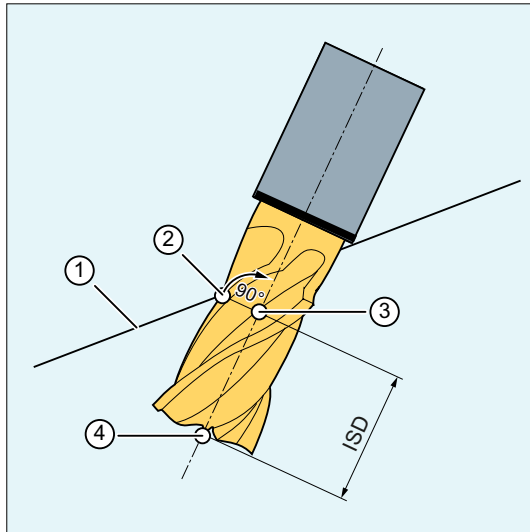
Über die G-Befehle ORIC und ORID kann festgelegt werden, ob Orientierungsänderungen, die zwischen den beiden die Ecke bildenden Verfahransätzen programmiert wurden, vor Beginn des eingefügten Kreissatzes (ORID) oder gleichzeitig mit diesem (ORIC) ausgeführt werden.

Eintauchtiefe

Die Eintauchtiefe des Fräasers ist der Abstand des Fräserhilfspunkts von der Werkzeugspitze.

Der Fräserhilfspunkt ist die senkrechte Projektion des Fräserbearbeitungspunkts auf der programmierten Bahn auf die Werkzeuglängsachse.

Mit der Eintauchtiefe wird somit die Lage des Bearbeitungspunkts auf der Mantelfläche des Werkzeugs eingestellt.



- ① Programmierte Bahn
- ② Fräserbearbeitungspunkt
- ③ Fräserhilfspunkt
- ④ Fräuserspitze
- ISD Eintauchtiefe (InSertion Depth)

Bild 26-11 Eintauchtiefe

Werkzeugradiuskorrektur bezogen auf ein Differenzwerkzeug

Die auf ein Differenzwerkzeug bezogene 3D-WRK für das Umfangsfräsen wird durch den Befehl CUT3DCD aktiviert. Sie ist anzuwenden, wenn sich die programmierte Kontur auf die Mittelpunktsbahn eines Normwerkzeugs bezieht und die Bearbeitung mit einem davon abweichenden Werkzeug erfolgt. Bei der Berechnung der 3D-Werkzeugradiuskorrektur werden dann nur der Verschleißwert des Radius des aktiven Werkzeugs (\$TC_DP15) und die gegebenenfalls programmierten Werkzeugkorrekturoffsets OFFN und TOFFR eingerechnet. Der Grundradius (\$TC_DP6) des aktiven Werkzeugs wird **nicht** eingerechnet.

Taschenfräsen mit schrägen Seitenwänden für Umfangsfräsen mit CUT3DC

Bei dieser 3D-Werkzeugradiuskorrektur wird eine Abweichung des Fräserradius kompensiert, indem in Richtung der Flächennormalen der zu bearbeitenden Fläche zugestellt wird. Dabei bleibt die Ebene, in der die Stirnseite des Fräasers liegt unverändert, wenn die Eintauchtiefe ISD gleich geblieben ist. Ein Fräser mit z. B. kleinerem Radius gegenüber einem Normwerkzeug würde dann den Taschenboden, der auch die Begrenzungsfläche darstellt, nicht erreichen. Für eine automatische Zustellung des Werkzeugs muss der Steuerung diese Begrenzungsfläche bekannt sein, siehe Kapitel "3D-Umfangsfräsen unter Berücksichtigung einer Begrenzungsfläche (CUT3DCC, CUT3DCCD) (Seite 951)".

Advanced Surface / Top Surface

Hinweis

Bei Verwendung der Werkzeugradiuskorrektur CUT3DCD in Kombination mit der lizenzpflichtigen Option "Advanced Surface" oder "Top Surface" sind die Einstellempfehlungen bezüglich "Advanced Surface" / "Top Surface" zu beachten!

Zur Überprüfung der eingestellten Daten stehen über das SIOS-Portal spezielle Prüfprogramme zur Verfügung:

- Prüfprogramme für Advanced Surface (<https://support.industry.siemens.com/cs/ww/de/view/78956392>)
- Prüfprogramme für Top Surface (<https://support.industry.siemens.com/cs/ww/de/view/109738423>)

26.3.2 3D-Werkzeugradiuskorrektur für das 3D-Stirnfräsen anwählen (CUT3DF, CUT3DFS, CUT3DFF, CUT3DFD)

Die 3D-Werkzeugradiuskorrektur (3D-WRK) für das 3D-Stirnfräsen wird mit dem modal wirksamen G-Befehl CUT3DF, CUT3DFS, CUT3DFF bzw. CUT3DFD angewählt.

Die eigentliche Aktivierung erfolgt mit G41 bzw. G42.

Um die Werkzeugradiuskorrektur berechnen zu können, ist beim 3D-Stirnfräsen die Definition der Flächennormale der zu bearbeitenden Ebene erforderlich. Diese muss im Satz mit G41 bzw. G42 über die Adressen A4, B4, C4 und A5, B5, C5 erfolgen.

Ausgeschaltet wird die Werkzeugradiuskorrektur mit G40.

Syntax

```
G41/G42 ORIC/ORID CUT3DF/CUT3DFS/CUT3DFF/CUT3DFD X... Y... Z... A4=... B4=...
C4=... A5=... B5=... C5=...
...
G40 X... Y... Z...
```

Bedeutung

CUT3DFS:	3D-WRK für das Stirnfräsen mit konstanter Orientierung. Die Werkzeugorientierung ist durch G17 - G19 festgelegt und wird durch Frames nicht beeinflusst.
CUT3DFF:	3D-WRK für das Stirnfräsen mit konstanter Orientierung. Die Werkzeugorientierung ist die durch G17 - G19 festgelegte und gegebenenfalls durch einen Frame gedrehte Richtung.
CUT3DF:	3D-WRK für das Stirnfräsen mit Orientierungsänderung (nur bei aktiver 5-Achs-Transformation)

CUT3DFD:	Auf ein Differenzwerkzeug bezogene 3D-WRK für das Stirnfräsen mit Orientierungsänderung (nur bei aktiver 5-Achs-Transformation) Die Radiusdifferenz wird durch den WZ-Parameter \$TC_DP15 festgelegt. Hinweis: CUT3DFD ist nur in Kombination mit der "Glättung der Flächennormalen beim 3D-Stirnfräsen" möglich. Diese wird aktiviert über den Aufruf der lizenzpflichtigen Funktion "Top Surface" mittels CYCLE832(...).	
G41/G42 X... Y... Z...:	Werkzeugkorrektur einschalten Das Verhalten bei G41 und bei G42 ist beim 3D-Stirnfräsen identisch. Hinweis: Das Einschalten muss in einem Linearsatz (G0/G1) erfolgen.	
A4/5=... B4/5=... C4/5=...:	Definition der Flächennormale der zu bearbeitenden Ebene	
	A4=... B4=... C4=...:	Definition am Satzanfang
	A5=... B5=... C5=...:	Definition am Satzende
ORIC/ORID:	Über die G-Befehle ORIC und ORID wird das Verhalten bei Orientierungsänderungen an Außenecken festgelegt.	
	ORIC:	Orientierungsänderungen an Außenecken werden dem einzufügenden Kreissatz überlagert.
	ORID:	Orientierungsänderungen werden vor dem Kreissatz ausgeführt.
G40 X... Y... Z...:	Werkzeugradiuskorrektur ausschalten Hinweis: Das Ausschalten muss in einem Linearsatz (G0/G1) mit Geometrieachsbewegungen erfolgen.	

Hinweis

G41 bzw. G42 kann auch in Sätzen ohne Verfahrensbewegung in den für die Korrektur relevanten Geometrieachsen programmiert sein. In diesem Fall ist der Anfahrsatz der erste auf einen solchen Satz folgende Verfahrtsatz.

Ein Wechsel der 3D-WRK-Variante bei aktiver Werkzeugradiuskorrektur wird ohne Alarm ignoriert.

Beispiele

Beispiel 1: 3D-Stirnfräsen mi CUT3DF

Programmcode	Kommentar
N10	; Definition des Werkzeugs D1:
N20 \$TC_DP1[1,1]=121	; Werkzeugtyp (Torusfräser)
N30 \$TC_DP3[1,1]=20	; Längenkorrektur
N40 \$TC_DP6[1,1]=5	; Radius
N50 \$TC_DP7[1,1]=3	; Verrundungsradius
N60	

Programmcode	Kommentar
N70	
N80 X0 Y0 Z0 A0 B0 C0 G17 T1 D1 F12000	; Anwahl des Werkzeugs.
N90 TRAORI(1)	; Orientierungstransformation anwählen.
N100 B4=-1 C4=1	; Definition der Ebene.
N110 G41 ORID CUT3DF G64 X10 Y0 Z0	; Werkzeugkorrektur einschalten.
N120 X30	
N130 Y20 A4=1 C4=1	; Außenecke, Neudefinition der Ebene.
N140 B3=1 C3=5	; Orientierungsänderung mit ORID.
N150 B3=1 C3=1	; Orientierungsänderung mit ORID.
N160 X-10 A5=1 C5=2 ORIC	
N170 A3=-2 C3=1	; Orientierungsänderung mit ORIC
N180 A3=-1 C3=1	; Orientierungsänderung mit ORIC
N190 Y-10 A4=-1 C4=3	; Neudefinition der Ebene.
N200 X-20 Y-20 Z10	; Innenecke mit Vorgängersatz.
N210 X-30 Y10 A4=1 C4=1	; Innenecke, Neudefinition der Ebene.
N220 A3=1 B3=0.5 C3=1.7	; Orientierungsänderung mit ORIC.
N230 X-20 Y30 A4=1 B4=-2 C4=3 ORID	
N240 A3 = 0.5 B3=-0.5 C3=1	; Orientierungsänderung.
N250 X0 Y30 C4=1	; Bahnbewegung, neue Ebene, ; Orientierung mit Relativprogrammierung.
N260 BSPLINE X20 Z15	; Splinebeginn, Relativprogrammierung der Orientierung
N270 X30 Y25 Z18	; bleibt während Spline aktiv.
N280 X40 Y20 Z13	
N290 X45 Y0 PW=2 Z8	
N300 Y-20	
N310 G2 ORIMKS A30 B45 I-20 X25 Y-40 Z0	; Helix, Orientierung mit Achsprogrammierung.
N320 G1 X0 A3=-0.123 B3=0.456 C3=2.789 B4=-1 C4=5 B5=-1 C5=2	; Bahnbewegung, Orientierung, nicht konstante Ebene.
N330 X-20 G40	; Ausschalten der Werkzeugradiuskorrektur.
N340 M30	

Beispiel 2: Aus einem CAD-System erzeugtes NC-Programm (Ausschnitt) mit CUT3DFD

Programmcode	Kommentar
N01 G710	
N03 T="12"	
N06 S5305 M03	
N07 G642	
	; Anfahren der Startposition im MKS unter Berücksichtigung der Werkzeuglänge.
G00 G90 X-250.62787 Y-38.37944 A=DC(253.12719) B-12.49543	
G00 G90 Z251.80052	
	; Ende der Positionierung im MKS.

26.3 Programmierung

Programmcode	Kommentar
; TRAORI (1) G500 D1 CYCLE832 (0.01, _TOP_SURFACE_SMOOTH_ON + _ORI_FI- NISH, 1)	; Orientierungstransformation anwählen. ; Aufruf CYCLE832 mit: ; Konturtoleranz = 0,01 mm, ; Bearbeitungsart: Top Surface mit Glättung, ; Schichten mit Eingabe einer Orientierungsto- leranz, ; Orientierungstoleranz = 1 Grad
CUT3DFD N08 G90 G94 N09 G00 X-269.21195 Y128.32027 Z1.18577 A3=-.216361688 B3=.934284397 C3=-.283373051	; Die Sätze N09 bis N10 sind der schnelle Anteil der Anfahrbewegung an das Werkstück mit konstan- ter Orientierung.
N10 G00 X-251.90301 Y53.57752 Z23.85561 N11 G01 X-247.57578 Y34.89183 Z29.52308 F50000.00000	; In den Sätzen N11 bis N21 wird der langsame An- teil der Anfahrbewegung realisiert. Das Werkzeug befindet sich bereits nahe am Werkstück, außer- dem sind jetzt die Formenbau-Einstellungen (z.B. COMPSURF) aus dem CYCLE832 aktiv (wegen aktivem G01). Der Bahnweg dieser sog. Einschwingphase für das Formenbau-Verhalten sollte etwa das 1000- fache der Konturtoleranz betragen (in diesem Bei- spiel also 10 mm).
N12 X-247.69126 Y33.82182 Z24.78219 F1061.00000 N13 X-247.76560 Y33.13299 Z21.73022 N14 X-247.82755 Y32.55897 Z19.18691 N15 X-247.87918 Y32.08062 Z17.06748 N16 X-247.92220 Y31.68200 Z15.30129 N17 X-247.95805 Y31.34981 Z13.82947 A3=-.216361686 B3=.934284391 C3=-.283373071 N18 X-247.98792 Y31.07299 Z12.60295 A3=-.216360662 B3=.934280801 C3=-.283385691 N19 X-248.01282 Y30.84230 Z11.58085 A3=-.216336015 B3=.934194446 C3=-.283689030 N20 X-248.03357 Y30.65006 Z10.72910 A3=-.216233089 B3=.933833626 C3=-.284952647 N21 X-248.05086 Y30.48986 Z10.01931 A5=-.060687572 B5=.974940255 C5=-.214029243 A3=-.215712821 B3=.932005189 C3=-.291263295 N22 G41 X-248.06237 Y30.32400 Z9.36695 A5=-.060431854 B5=.973045457 C5=-.222554556 A3=-.214974689 B3=.929398552 C3=-.300007025 F1061.03295	; Ab N22 ist die Flächennormale erstmals voll- ständig über den gesamten Satz definiert (d.h., Flächennormale am Ende des Vorgängersatzes N21 ist vorhanden, somit die Flächennormale am An- fang des Satzes N22 und die Flächennormale am En- de des Satzes N22). Damit ist die Voraussetzung für das Einschalten der Werkzeugkorrektur mit G41/G42 erfüllt.

Programmcode	Kommentar
N23 X-248.07130 Y30.15119 Z8.71082 A5=-.060165696 B5=.971048883 C5=-.231179920 A3=-.214177198 B3=.926684940 C3=-.308841625	
N24 X-248.07829 Y29.97126 Z8.05094 A5=-.059884286 B5=.968941717 C5=-.239928784 A3=-.213318480 B3=.923853466 C3=-.317789237	
N25 X-248.08317 Y29.78487 Z7.38844 A5=-.059584206 B5=.966718449 C5=-.248807482 A3=-.212397895 B3=.920898045 C3=-.326854594	
N26 X-248.08578 Y29.59254 Z6.72679 A5=-.059263963 B5=.964380907 C5=-.257793037 A3=-.211418355 B3=.917822366 C3=-.336012474	
...	

Hinweis

Beim 3D-Stirnfräsen mit CUT3DFD ist für das Einschalten der Werkzeugkorrektur mit G41/G42 die Definition der Flächennormalen erforderlich. Die Programmierung von G41/G42 ohne Definition der Flächennormalen führt zur Ausgabe eines Alarms.

Weitere Informationen

3D-Stirnfräsen

Für diese Art des 3D-FräSENS wird die zeilenweise Beschreibung der 3D-Bahnen auf der Werkstückoberfläche benötigt. Die Berechnungen werden unter Berücksichtigung der Werkzeugform und Werkzeugabmessungen üblicherweise im CAM durchgeführt. Der Postprozessor schreibt in das Teileprogramm – neben den NC-Sätzen – die Werkzeugorientierungen (bei aktiver 5-Achstransformation) und den G-Befehl für die gewünschte 3D-Werkzeugkorrektur. Hierdurch hat der Maschinenbediener die Möglichkeit – abweichend von dem für die Berechnung der NC-Bahnen verwendeten Werkzeug – geringfügig kleinere Werkzeuge einzusetzen.

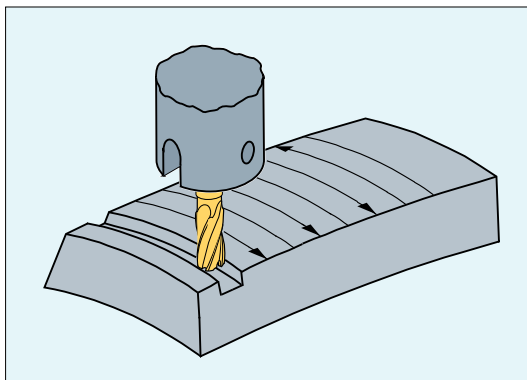


Bild 26-12 Stirnfräsen

Anfahrverhalten

Das Anfahrverhalten ist bei den 3D-Varianten der Werkzeugradiuskorrektur immer NORM.

Verhalten an Außenecken

Außenecken werden beim Stirnfräsen als Kreise mit dem Radius 0 behandelt, wobei die Kreisebene von der Endtangente des ersten und der Starttangente des zweiten Satzes aufgespannt wird. Damit ist auch eine Orientierungsänderung beim Satzübergang möglich. An einer Außenecke wird somit immer ein Kreis als Konturelement eingefügt. Das Schnittpunktverfahren steht beim Stirnfräsen nicht zur Verfügung.

Verhalten bei Orientierungsänderungen an Außenecken

Über die G-Befehle ORIC und ORID kann festgelegt werden, ob Orientierungsänderungen, die zwischen den beiden die Ecke bildenden Verfahrssätzen programmiert wurden, vor Beginn des eingefügten Kreissatzes (ORID) oder gleichzeitig mit diesem (ORIC) ausgeführt werden.

Werkzeugradiuskorrektur bezogen auf ein Differenzwerkzeug

Die auf ein Differenzwerkzeug bezogene 3D-Werkzeugradiuskorrektur wird durch den Befehl CUT3DFD angewählt. Sie ist anzuwenden, wenn sich die programmierte Kontur auf die Mittelpunktsbahn eines Normwerkzeugs bezieht und die Bearbeitung mit einem davon abweichenden Werkzeug erfolgt. Bei der Berechnung der 3D-Werkzeugradiuskorrektur werden dann nur der Verschleißwert des Radius des aktiven Werkzeugs (\$TC_DP15) und die gegebenenfalls programmierten Werkzeugkorrekturoffsets OFFN und TOFFR eingerechnet. Der Grundradius (\$TC_DP6) des aktiven Werkzeugs wird **nicht** eingerechnet.

Das 3D-Stirnfräsen mit CUT3DFD ist nur in Kombination mit der "Glättung der Flächennormalen beim 3D-Stirnfräsen" möglich. Diese wird aktiviert über den Aufruf der lizenzpflichtigen Funktion "Top Surface" mittels CYCLE832(...). Die Aktivierung muss **vor** dem Einschalten der Werkzeugkorrektur mit G41/G42 erfolgen, und zwar nicht direkt vor dem Werkzeugeingriff, sondern schon eine Bahnweglänge vorher, die etwa dem 1000-fachen der Konturtoleranz entspricht (z. B. 1000 x 0,01 mm = 10 mm). Die Deaktivierung muss in umgekehrter Reihenfolge ablaufen: erst Ausschalten der Werkzeugkorrektur mit G40, dann, nach einer Bahnweglänge, die etwa dem 1000-fachen der Konturtoleranz entspricht, die Deaktivierung durch z. B. CUT2D (o. ä.).

Um die "Glättung der Flächennormalen beim 3D-Stirnfräsen" anwenden zu können, ist zusätzlich die Freischaltung der Funktion "Interpolation der Flächennormalen über Polynome" erforderlich:

```
MD28291 $MC_MM_SMOOTH_SURFACE_NORMALS = TRUE
```

Hinweis

Für das 3D-Stirnfräsen mit CUT3DFD in Kombination mit "Top Surface" sind die Einstellempfehlungen bezüglich "Top Surface" zu beachten!

Zur Überprüfung der eingestellten Daten stehen über das SIOS-Portal spezielle Prüfprogramme zur Verfügung (<https://support.industry.siemens.com/cs/ww/de/view/109738423>).

26.3.3 3D-Umfangsfräsen unter Berücksichtigung einer Begrenzungsfläche (CUT3DCC, CUT3DCCD)

Beim 3D-Umfangsfräsen mit kontinuierlicher oder konstanter Veränderung der Werkzeugorientierung wird häufig die Werkzeugmittelpunktsbahn für ein definiertes Normwerkzeug programmiert. Da in der Praxis oft nicht die passenden Normwerkzeuge zur Verfügung stehen, kann ein von einem Normwerkzeug nicht allzu stark abweichendes Werkzeug ($\leq 5\%$) eingesetzt werden.

Mit CUT3DCCD wird für ein reales Differenzwerkzeug eine Begrenzungsfläche berücksichtigt, die das programmierte Normwerkzeug beschreiben würde. Das NC-Programm beschreibt die Mittelpunktsbahn des Normwerkzeugs.

Mit CUT3DCC wird bei Verwendung von zylindrischen Werkzeugen eine Begrenzungsfläche berücksichtigt, die das programmierte Normwerkzeug erreicht hätte. Das NC-Programm beschreibt die Kontur auf der Bearbeitungsfläche.

Der Flächennormalenvektor der Begrenzungsfläche wird wie beim 3D-Stirnfräsen mit A4, B4, C4 und A5, B5, C5 angegeben.

Syntax

```
G41/G42 CUT3DCCD/CUT3DCC CDOF2 X... Y... Z... A4=... B4=... C4=... A5=... B5=...
C5=...
...
G40 X... Y... Z...
```

Bedeutung

CUT3DCCD:	3D-WRK für das Umfangsfräsen unter Berücksichtigung einer Begrenzungsfläche mit Differenzwerkzeug auf der Werkzeugmittelpunktsbahn: Zustellung zur Begrenzungsfläche	
CUT3DCC:	3D-WRK für das Umfangsfräsen unter Berücksichtigung einer Begrenzungsfläche mit 3D-Radiuskorrektur: Kontur an der Bearbeitungsfläche	
G41/G42 X... Y... Z...:	Werkzeugradiuskorrektur einschalten	
	G41:	Werkzeugradiuskorrektur links von der Kontur
	G42:	Werkzeugradiuskorrektur rechts von der Kontur
	Hinweis: Das Einschalten muss in einem Linearsatz (G0/G1) erfolgen.	
CDOF2:	Kollisionsüberwachung für 3D-Umfangsfräsen ausschalten	
A4/5=... B4/5=... C4/5=...:	Definition der Flächennormale der Begrenzungsfläche	
	A4=... B4=... C4=...:	Definition am Satzanfang
	A5=... B5=... C5=...:	Definition am Satzende
G40 X... Y... Z...:	Werkzeugradiuskorrektur ausschalten	
	Hinweis: Das Ausschalten muss in einem Linearsatz (G0/G1) mit Geometriachsbewegungen erfolgen.	

Hinweis

Die G-Befehle zur Anwahl der 3D-WRK werden im Anfahrsatz ausgewertet, d. h. typischerweise in dem Satz, der G41 oder G42 enthält.

G41 bzw. G42 kann auch in Sätzen ohne Verfahrbewegung in den für die Korrektur relevanten Geometrieachsen programmiert sein. In diesem Fall ist der Anfahrsatz der erste auf einen solchen Satz folgende Verfahrsatz.

Ein Wechsel der 3D-WRK-Variante bei aktiver Werkzeugradiuskorrektur wird ohne Alarm ignoriert.

Beispiel

Programmcode	Kommentar
N10 \$TC_DP1[1,1]=120	; Zylinderfräser
N20 \$TC_DP6[1,1]=10	
N30 \$TC_DP15[1,1]=-3	
...	
; Bearbeitung mit Zylinderfräser und CUT3DCCD	
N110 TRAORI	; Einschalten der Transformation.
N120 A4=0 B4=0 C4=1	; Definition der Flächennormale der Begrenzungsfläche am Satzanfang.
N130 X0 Y0 Z0 A0 C0 T1 D1 F20000	
N140 X10 Y0 Z0 G41 CUT3DCCD CDOF2 G64	; Aktivieren des 3D-Umfangsfräsens unter Berücksichtigung der Begrenzungsfläche + Ausschalten der Kollisionsüberwachung.
N150 X20	
N160 X30 A45	; Stumpfer Winkel ==> keine Zustellung
N170 X40 A-45	; Spitzer Winkel ==> Zustellung
N180 X55	
N190 Y10 Z10	; Bewegung in Werkzeugrichtung.
N200 Y20	
N210 C45	; Reine Orientierungsänderung.
N220 Y30 C90	
N230 A5=-1 B5=0 C5=2 Y40	; Änderung der Oberfläche.
N240 Y50 G40	; Deaktivieren der Werkzeugradiuskorrektur.
...	

Weitere Informationen

Werkzeugtyp

Der Werkzeugtyp (Werkzeugparameter \$TC_DP1) wird ausgewertet. Es sind nur Fräswerkzeuge mit zylindrischem Schaft (Zylinder- oder Schafffräser, Torusfräser und als

Grenzfall der zylindrische Gesenkfräser) zugelassen. Das entspricht den Werkzeugtypen 1 - 399 mit Ausnahme der Nummern 111 und 155 bis 157.

Normwerkzeuge mit Eckenverrundung

Die Eckenverrundung des Normwerkzeugs wird durch den Werkzeugparameter \$TC_DP7 beschrieben. Aus dem Werkzeugparameter \$TC_DP16 ergibt sich die Abweichung der Eckenverrundung des realen Werkzeugs gegenüber dem Normwerkzeug.

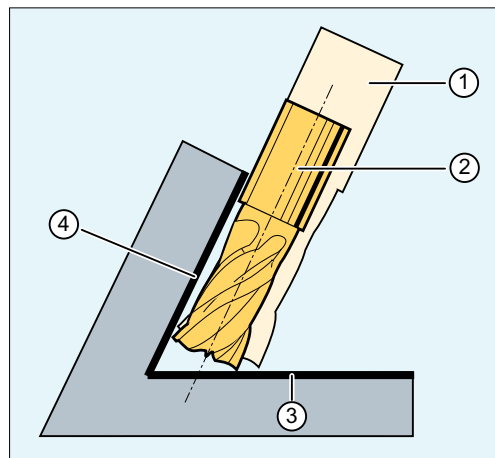
Beispiel: Torusfräser mit verringertem Radius gegenüber dem Normwerkzeug

Werkzeugtyp	Schaffradius (R)	Eckenradius (r)
Normwerkzeug mit Eckenverrundung	$R = \$TC_DP6$	$r = \$TC_DP7$
Reales Werkzeug mit Eckenverrundung Werkzeugtypen 121 und 131 Torusfräser (Schafffräser mit Eckenverrundung)	$R' = \$TC_DP6 + \$TC_DP15 + OFFN$	$r' = \$TC_DP7 + \TC_DP16

In diesem Beispiel sind sowohl \$TC_DP15 + OFFN als auch \$TC_DP16 negativ.

3D-WRK mit CUT3DCCD: Werkzeugmittelpunktsbahn mit Zustellung bis zur Begrenzungsfläche

Wird ein Werkzeug verwendet, welches im Vergleich zum passenden Normwerkzeug einen kleineren Radius aufweist, dann wird ein in Längsrichtung zugestellter Fräser soweit weiter geführt, bis dieser den Taschenboden wieder berührt. Damit wird die Ecke, die von der Bearbeitungs- und der Begrenzungsfläche gebildet wird, so weit ausgeräumt, wie dies das Werkzeug zulässt. Es handelt sich dabei um eine gemischte Bearbeitungsweise aus Umfangs- und Stirnfräsen. Analog zu einem Werkzeug mit verringertem Radius, wird beim Werkzeug mit vergrößertem Radius in die entgegengesetzte Richtung entsprechend zugestellt.



- ① Normwerkzeug
- ② Werkzeug mit kleinerem Radius zugestellt bis zur Begrenzungsfläche
- ③ Begrenzungsfläche
- ④ Bearbeitungsfläche

Gegenüber allen anderen Werkzeugkorrekturen der G-Gruppe 22 hat ein für CUT3DCCD angegebener Werkzeugparameter \$TC_DP6 keine Bedeutung für den Werkzeugradius und beeinflusst die resultierende Korrektur nicht. Der Korrekturoffset ergibt sich aus der Summe des Verschleißwerts des Werkzeugradius (Werkzeugparameter \$TC_DP15) und einem zur Berechnung des senkrechten Offsets zur Begrenzungsfläche programmierten Werkzeugoffset OFFN.

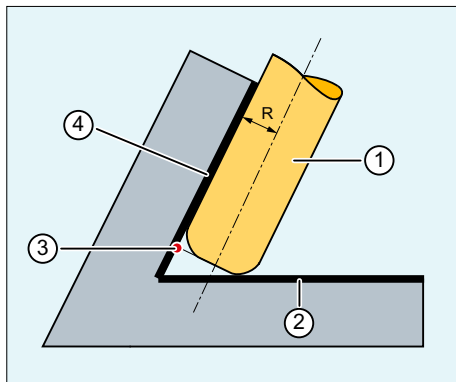
Ob die zu bearbeitende Fläche links oder rechts von der Bahn liegt, kann aus dem erzeugten Teileprogramm nicht entnommen werden. Es wird deshalb von einem positiven Radius und einem negativen Verschleißwert des Originalwerkzeugs ausgegangen. Ein negativer Verschleißwert beschreibt immer ein Werkzeug mit verringertem Durchmesser.

Verwendung von zylindrischen Werkzeugen

Bei der Verwendung von zylindrischen Werkzeugen ist eine Zustellung nur dann erforderlich, wenn die Bearbeitungsfläche und die Begrenzungsfläche einen spitzen Winkel (kleiner als 90 Grad) bilden. Werden Torusfräser (Schafffräser mit Eckverrundung) verwendet, dann erfordert dies sowohl bei spitzen als auch bei stumpfen Winkeln eine Zustellung in Längsrichtung des Werkzeugs.

3D-WRK mit CUT3DCC: Kontur an der Bearbeitungsfläche

Ist CUT3DCC mit einem Torusfräser aktiv, so bezieht sich die programmierte Bahn auf einen fiktiven Zylinderfräser gleichen Durchmessers. Der hieraus resultierende Bahnbezugspunkt ist bei Verwendung eines Torusfräser im folgenden Bild dargestellt.



- ① Torusfräser
- ② Begrenzungsfläche
- ③ Bahnbezugspunkt
- ④ Bearbeitungsfläche
- R Schafradius (Werkzeugradius)

Es ist zulässig, dass der Winkel zwischen Bearbeitungs- und Begrenzungsfläche auch innerhalb eines Satzes von einem spitzen in einem stumpfen Winkel oder umgekehrt übergeht.

Gegenüber dem Normwerkzeug darf das verwendete reale Werkzeug sowohl größer als auch kleiner sein. Dabei darf der resultierende Eckenradius nicht negativ werden und das Vorzeichen des resultierenden Werkzeugradius muss erhalten bleiben.

Bei CUT3DCC bezieht sich das NC-Teileprogramm auf die Kontur an der Bearbeitungsfläche. Es wird hierbei wie bei der herkömmlichen Werkzeugradienkorrektur der Gesamtradius herangezogen, der sich aus folgenden Komponenten zusammensetzt:

- Werkzeugradius (Werkzeugparameter \$TC_DP6)
- Verschleißwert (Werkzeugparameter \$TC_DP15)
- Einem zur Berechnung des senkrechten Offsets zur Begrenzungsfläche programmierten Werkzeugoffset OFFN

Die Lage der Begrenzungsfläche wird aus folgender Differenz bestimmt:

Abmessungen des Normwerkzeugs - Werkzeugradius (WZ-Parameter \$TC_DP6)

Advanced Surface / Top Surface

Hinweis

Bei Verwendung der Werkzeugradiuskorrektur CUT3DCC / CUT3DCCD in Kombination mit der lizenzpflichtigen Funktion "Advanced Surface" oder "Top Surface" sind die Einstellempfehlungen bezüglich "Advanced Surface" / "Top Surface" zu beachten!

Zur Überprüfung der eingestellten Daten stehen über das SIOS-Portal spezielle Prüfprogramme zur Verfügung:

- Prüfprogramme für Advanced Surface (<https://support.industry.siemens.com/cs/ww/de/view/78956392>)
- Prüfprogramme für Top Surface (<https://support.industry.siemens.com/cs/ww/de/view/109738423>)

26.4 Randbedingungen

Randbedingungen für das Stirnfräsen

Funktionen mit höherwertigeren Geometrieinformationen

Funktionen mit höherwertigeren Geometrieinformationen (z. B. Polynom- oder Spline-Interpolation) sind nicht anwendbar.

Top Surface

Das 3D-Stirnfräsen mit CUT3DFD ist nur in Kombination mit der lizenzpflichtigen Funktion "Top Surface" möglich.

Flaschenhalserkennung

Eine "Flaschenhalserkennung" ist nicht möglich und liegt daher in der alleinigen Verantwortung des Anwenders.

Verfahrbewegungen in Orientierungsrichtung des Werkzeugs

Bei Verfahrbewegungen in Orientierungsrichtung des Werkzeugs wird ein Alarm angezeigt und die Bearbeitung gestoppt.

26.5 Beispiele

26.5.1 Beispiel 1: Orientierungsänderung an Außenecke beim 3D-Umfangsfräsen

Orientierungsänderung an Außenecke bei aktivem ORIC

Programmcode	Kommentar
N10 A0 B0 X0 Y0 Z0 F5000	
N20 T1 D1	; Anwahl des Werkzeugs (Radius=5).
N30 TRAORI(1)	; Einschalten der Transformation.
N40 CUT3DC	; Anwahl 3D-WRK für Umfangsfräsen.
N50 ORIC	; Orientierungsänderungen an Außenecken dem einzufügenden Kreissatz überlagern.
N60 G42 X10 Y10	; Einschalten der WRK.
N70 X60	
N80 A3=1 B3=0 C3=1	; Orientierungsänderung an der von N70 und N90 gebildeten Außenecke.
N90 Y60	
N100 X10	
N110 G40 X0 Y0	; Ausschalten der WRK.
N120 M30	

Im Satz N80 werden Kreisbewegung und Orientierungsänderung parallel ausgeführt (ORIC aktiv):

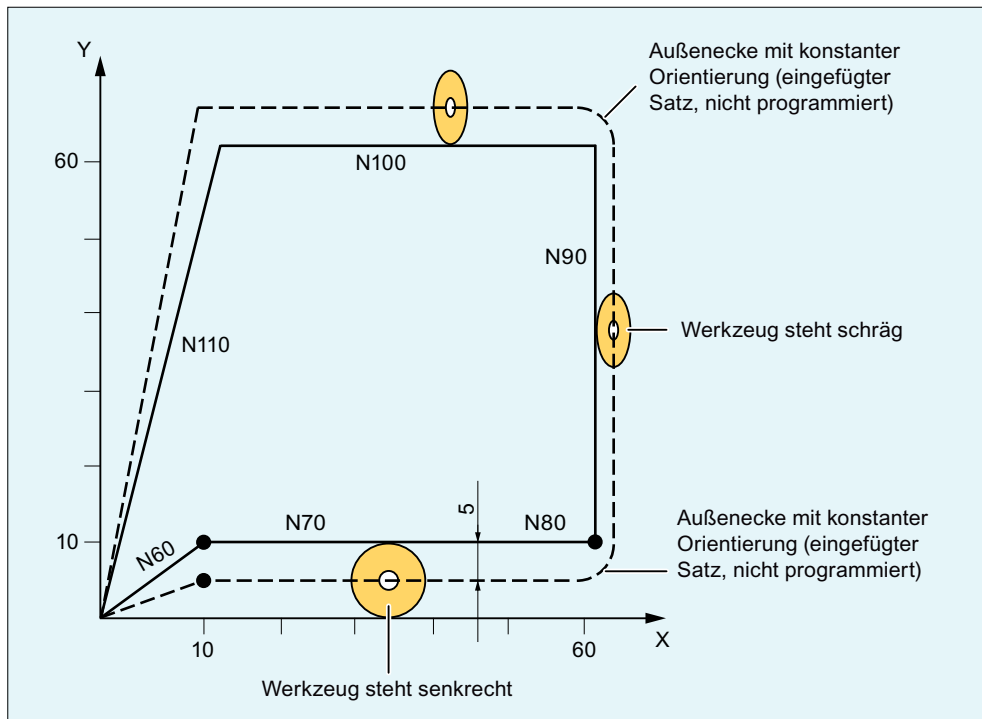


Bild 26-13 ORIC: Orientierungsänderung und Bahnbewegung parallel

Sonderfall:

Zwischensätze ohne Verfah- und Orientierungsbewegungen werden an den programmierten Stellen ausgeführt, z. B. Hilfsfunktionen.

Beispiel:

Programmcode	Kommentar
N70 X60	
N75 M20	; Hilfsfunktionsaufruf
N80 A3=1 B3=0 C3=1	; Orientierungsänderung an der von N70 und N90 gebildeten Außenecke.
N90 Y60	

Die Sätze N75 und N80 werden nach N70 ausgeführt. Danach wird der Kreissatz mit der aktuellen Orientierung ausgefahren.

Orientierungsänderung an Außenecke bei aktivem ORID

Programmcode	Kommentar
N10 A0 B0 X0 Y0 Z0 F5000	
N20 T1 D1	; Anwahl des Werkzeugs (Radius=5).
N30 TRAORI(1)	; Einschalten der Transformation.
N40 CUT3DC	; Anwahl 3D-WRK für Umfangsfräsen.
N50 ORID	; Orientierungsänderungen an Außenecken vor dem einzufügenden Kreissatz ausführen.
N60 G42 X10 Y10	; Einschalten der WRK.
N70 X60	
N80 A3=1 B3=0 C3=1	; Orientierungsänderung an der von N70 und N90 gebildeten Außenecke.
N90 Y60	
N100 X10	
N110 G40 X0 Y0	; Ausschalten der WRK.
N120 M30	

26.5 Beispiele

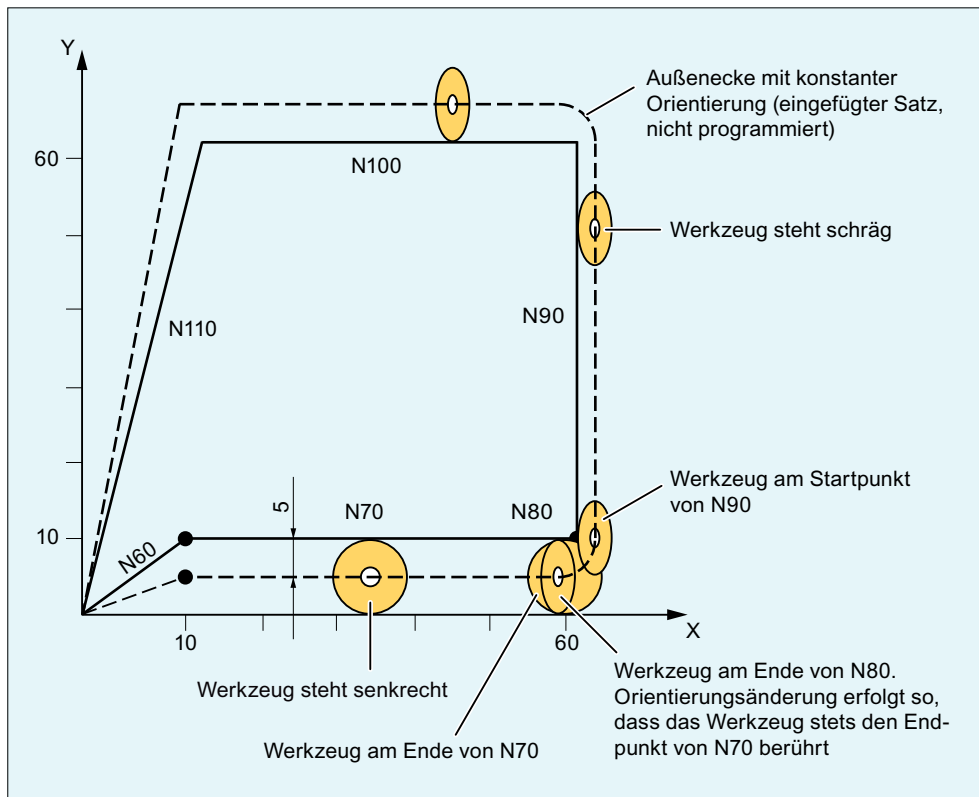
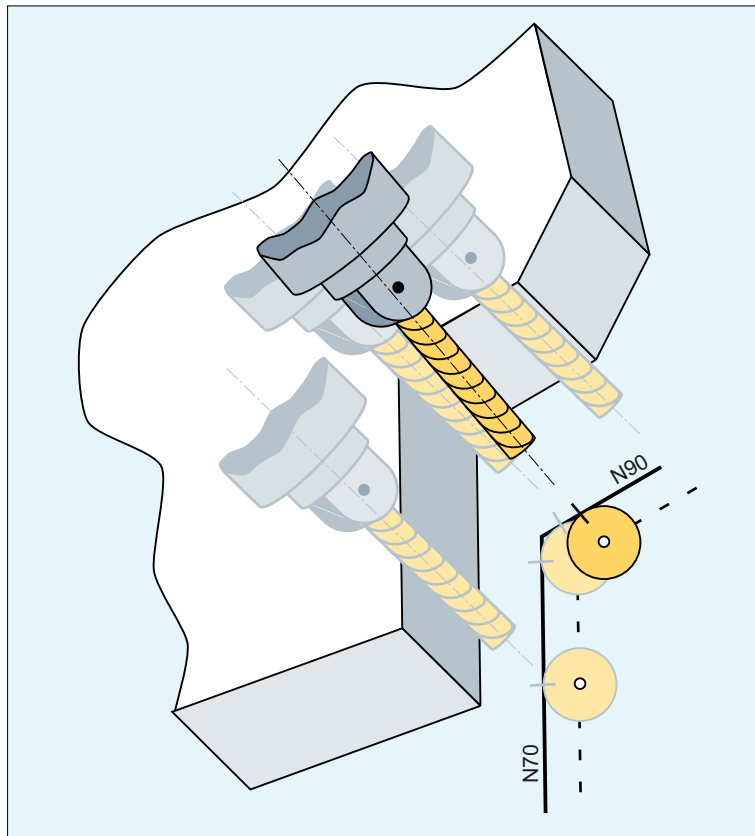


Bild 26-14 ORID: Orientierungsänderung und Bahnbewegung nacheinander

26.5.2 Beispiel 2: Orientierungsänderung an Innenecke beim 3D-Umfangsfräsen

Programmcode	Kommentar
N10 A0 B0 X0 Y0 Z0 F5000	
N20 T1 D1	; anwahl des Werkzeugs (Radius=5).
N30 TRAORI(1)	; Einschalten der Transformation.
N40 CUT3DC	; Anwahl 3D-WRK für Umfangsfräsen.
N50 ORID	
N60 G42 X10 Y10 G451	; Einschalten der WRK.
N70 Y60	
N80 A3=1 B3=0 C3=1	; Orientierungsänderung an der von N70 und N90 gebildeten Innenecke.
N90 X60 Y90	
N100 G40 X... Y...	
...	
N190 CDOF	
N200 M30	



26.6 Datenlisten

26.6.1 Allgemeine Maschinendaten

Nummer	Bezeichner: \$MN_	Beschreibung
18094	MM_NUM_CC_TDA_PARAM	Anzahl der TDA-Daten
18096	MM_NUM_CC_TOA_PARAM	Anzahl der TOA-Daten, die pro Werkzeug angelegt werden und vom CC ausgewertet werden können
18100	MM_NUM_CUTTING_EDGES_IN_TOA	Werkzeugkorrekturen pro TOA-Baustein
18110	MM_NUM_TOA_MODULES	Anzahl der TOA-Bausteine

26.6.2 Kanal-spezifische Maschinendaten

Nummer	Bezeichner: \$MC_	Beschreibung
20110	RESET_MODE_MASK	Festlegung der Steuerungs-Grundstellung nach Hochlauf und RESET/Teileprogrammende
20120	TOOL_RESET_VALUE	Festlegung des Werkzeuges, von dem im Hochlauf und bei RESET bzw. Teileprogrammende in Abhängigkeit von MD 20110 die Werkzeuglängenkorrektur angewählt wird
20130	CUTTING_EDGE_RESET_VALUE	Festlegung der Werkzeugschneide, von der im Hochlauf und bei RESET bzw. Teileprogrammende in Abhängigkeit von MD20110 die Werkzeuglängenkorrektur angewählt wird
20140	TRAFO_RESET_VALUE	Festlegung des Transformationsatzes, der im Hochlauf und bei RESET bzw. Teileprogrammende in Abhängigkeit von MD20110 angewählt wird
20210	CUTCOM_CORNER_LIMIT	Maximalwinkel für Ausgleichssätze bei WRK
20220	CUTCOM_MAX_DISC	Verhalten der WRK an Außenecken
20230	CUTCOM_CURVE_INSERT_LIMIT	Minimalwert für Schnittpunktberechnung bei WRK
20240	CUTCOM_MAXNUM_CHECK_BLOCKS	Sätze für vorausschauende Konturberechnung bei WRK
20250	CUTCOM_MAXNUM_DUMMY_BLOCKS	Satzanzahl ohne Verfahrbewegung bei WRK
20270	CUTTING_EDGE_DEFAULT	Angewählte WZ-Schneide nach WZ-Wechsel
20610	ADD_MOVE_ACCEL_RESERVE	Beschleunigungsreserve für überlagerte Bewegungen
21080	CUTCOM_PARALLEL_ORI_LIMIT	Minimaler Winkel zwischen Bahntangente und WZ-Orientierung bei 3D-WRK
21082	CUTCOM_PLANE_ORI_LIMIT	Minimaler Winkel zwischen Flächennormalenvektor und WZ-Orientierung (beim 3D-Stirnfräsen)
21084	CUTCOM_PLANE_PATH_LIMIT	Minimaler Winkel zwischen Flächennormalenvektor und Bahntangentenvektor (beim 3D-Stirnfräsen)
22550	TOOL_CHANGE_MODE	Neue WZK bei M-Funktion
22560	TOOL_CHANGE_M_CODE	M-Funktion für Werkzeugwechsel
28291	MM_SMOOTH_SURFACE_NORMALS	Interpolation der Flächennormalen über Polynome (beim 3D-Stirnfräsen)

W6: Weglängenauswertung - nur 840D sl

27.1 Kurzbeschreibung

Funktion

Mit der Funktion "Weglängenauswertung" stellt der NC spezifische Daten von Maschinenachsen als System- und BTSS-Variable zur Verfügung, mit deren Hilfe es möglich ist, die Belastung der Maschinenachsen einzuschätzen und somit eine Aussage über den Wartungszustand der Maschine machen zu können.

Aufgezeichnete Daten

Folgende Daten werden aufgezeichnet:

- Gesamtverfahrweg
- Gesamtverfahrzeit
- Gesamte Anzahl der Verfahrvorgänge
- Gesamtverfahrweg bei grossen Achsgeschwindigkeiten
- Gesamtverfahrzeit bei grossen Achsgeschwindigkeiten
- Anzahl der Verfahrvorgänge bei grossen Achsgeschwindigkeiten
- Gesamtsumme des Rucks
- Verfahrzeit der Achse mit Ruck
- Anzahl der Verfahrvorgänge mit Ruck

Auswertung

Mit Aktivierung der Funktion werden die ausgewählten Daten von der Steuerung automatisch mitgeführt und über System- und BTSS-Variable zur Auswertung im Teileprogramm oder Synchronaktionen bzw. über Anwender-spezifische HMI-Funktionen zur Verfügung gestellt.

Bedeutung

Die Daten werden in der Steuerung remanent gespeichert, so dass sie auch nach POWER OFF / ON kontinuierlich weitergeführt werden können. Somit repräsentiert der aktuelle Wert eines Datums die Summe aller Messwerte seit Aktivierung der Funktion.

27.2 Daten

Folgende Daten stehen zu Verfügung:

Systemvariable ¹⁾	BTSS-Variable	Bedeutung
\$AA_TRAVEL_DIST	aaTravelDist	Gesamtverfahrweg: Summe aller Sollpositionsänderungen im MKS ²⁾ in [mm] bzw. [Grad].
\$AA_TRAVEL_TIME	aaTravelTime	Gesamtverfahrzeit: Summe der IPO-Takte von Sollpositionsänderungen im MKS ²⁾ in [s] (Auflösung: 1 IPO-Takt)
\$AA_TRAVEL_COUNT	aaTravelCount	Gesamtanzahl der Verfahrvorgänge: Ein Verfahrvorgang einer Maschinenachse ist durch folgende Zustandsfolge bezogen auf die Sollposition gekennzeichnet: Stillstand > Verfahren > Stillstand
\$AA_TRAVEL_DIST_HS	aaTravelDistHS	Gesamtverfahrweg bei großer Achsgeschwindigkeit ³⁾
\$AA_TRAVEL_TIME_HS	aaTravelTimeHS	Gesamtverfahrzeit bei großer Achsgeschwindigkeit ³⁾
\$AA_TRAVEL_COUNT_HS	aaTravelCountHS	Gesamtanzahl der Verfahrvorgänge bei großer Achsgeschwindigkeit ³⁾
\$AA_JERK_TOT	aaJerkTotal	Gesamtsumme des Rucks der Achse: Summe aller Sollwerte des Rucks in [m/s ³] bzw. [Grad/ s ³].
\$AA_JERK_TIME	aaJerkTime	Gesamtverfahrzeit mit Ruck: Summe der IPO-Takte von Sollwertänderungen des Rucks in [s] (Auflösung: 1 IPO-Takt)
\$AA_JERK_COUNT	aaJerkCount	Gesamtanzahl der Verfahrvorgänge mit Ruck
¹⁾ Die Systemvariablen können aus Teileprogrammen und Synchronaktionen heraus gelesen werden ²⁾ MKS: Maschinenkoordinatensystem ³⁾ Istgeschwindigkeit der Maschinenachse ≥ 80% der parametrierten maximalen Achsgeschwindigkeit (MD32000 MAX_AX_VELO)		

27.3 Parametrierung

27.3.1 Allgemeine Aktivierung

Die allgemeine Aktivierung der Funktion erfolgt über das NC-spezifische Maschinendatum:
MD18860 \$MN_MM_MAINTENANCE_MON (Aktivierung der Aufzeichnung von Wartungsdaten)

27.3.2 Datengruppen

Die Daten sind in Datengruppen zusammengefasst.

Die Aktivierung der Datengruppen erfolgt über das achsspezifische Maschinendatum:

MD33060 \$MA_MAINTENANCE_DATA (Konfiguration der Aufzeichnung von Wartungsdaten)

Bit	Wert	Aktivierung folgender Daten: Systemvariable / BTSS-Variable
0	1	<ul style="list-style-type: none"> • Gesamtverfahrweg: \$AA_TRAVEL_DIST / aaTravelDist • Gesamtverfahrzeit: \$AA_TRAVEL_TIME / aaTravelTime • Gesamtanzahl der Verfahrvorgänge: \$AA_TRAVEL_COUNT / aaTravelCount
1	1	<ul style="list-style-type: none"> • Gesamtverfahrweg bei großen Geschwindigkeiten: \$AA_TRAVEL_DIST_HS / aaTravelDistHS • Gesamtverfahrzeit bei großen Geschwindigkeiten: \$AA_TRAVEL_TIME_HS / aaTravelTimeHS • Gesamtanzahl der Verfahrvorgänge bei großen Geschwindigkeiten: \$AA_TRAVEL_COUNT_HS / aaTravelCountHS
2	1	<ul style="list-style-type: none"> • Gesamtsumme des Rucks: \$AA_JERK_TOT / aaJerkTotal • Verfahrzeit mit Ruck: \$AA_JERK_TIME / aaJerkTime • Gesamtanzahl der Verfahrvorgänge mit Ruck: \$AA_JERK_COUNT / aaJerkCount

27.4 Beispiele

27.4.1 Verfahrweg pro Teileprogramm

An einer Maschine sind drei Geometrieachsen AX1, AX2 und AX3 vorhanden. Für die Geometrieachse AX1 soll der im Teileprogramm gefahrene Gesamtverfahrweg, die Gesamtverfahrzeit und die Anzahl der Verfahrvorgänge ermittelt werden.

Parametrierung

Aktivierung der Gesamtfunktion:

```
MD18860 $MN_MM_MAINTENANCE_MON = TRUE
```

Aktivierung der Gruppe: "Gesamtverfahrweg, Gesamtverfahrzeit und Anzahl der Verfahrvorgänge" für die Geometrieachse AX1:

```
MD33060 $MA_MAINTENANCE_DATA[AX1] = 1
```

Programmierung

Zur Ermittlung der Werte bezogen auf das Teileprogramm muss der aktuelle Wert der Systemvariablen am Anfang des Teileprogramms in einer Rechenvariablen gespeichert werden. Die Differenzbildung erfolgt dann am Ende des Teileprogramms.

Teileprogramm (Auszug):

Programmcode	Kommentar
...	

27.5 Datenlisten

Programmcode	Kommentar
	; Aktuelle Werte:
R10=\$AA_TRAVEL_DIST[AX1]	; Gesamtverfahrweg AX1
R11=\$AA_TRAVEL_TIME[AX1]	; Gesamtverfahrzeit AX1
R12=\$AA_TRAVEL_COUNT[AX1]	; Anzahl Verfahrvorgänge AX1
...	; Verfahrbewegung der Achsen
	; Differenzbildung:
R10=R10-\$AA_TRAVEL_DIST[AX1]	; Verfahrweg AX1 im Teileprogramm
R11=R11-\$AA_TRAVEL_TIME[AX1]	; Verfahrzeit AX1 im Teileprogramm
R12=R12-\$AA_TRAVEL_COUNT[AX1]	; Anzahl Verfahrvorgänge AX1 im Teileprogramm

27.5 Datenlisten

27.5.1 Maschinendaten

27.5.1.1 NC-spezifische Maschinendaten

Nummer	Bezeichner: \$MN_	Beschreibung
18860	MM_MAINTENANCE_MON	Aktivierung der Aufzeichnung von Wartungsdaten

27.5.1.2 Achs-/Spindel-spezifische Maschinendaten

Nummer	Bezeichner: \$MA_	Beschreibung
33060	MAINTENANCE_DATA	Konfiguration der Aufzeichnung von Wartungsdaten

Z3: NC/PLC-Nahtstellensignale

Ab Ausgabestand 05/2017 befindet sich die ausführliche Beschreibung der NC/PLC-Nahtstellensignale im Listenhandbuch NC-Variablen und Nahtstellensignale.

A.1 Liste der Abkürzungen

A	
A	Ausgang
ADI4	Analog Drive Interface for 4 Axes
AC	Adaptive Control
ALM	Active Line Module
ARM	Asynchroner rotatorischer Motor
AS	Automatisierungssystem
ASCII	American Standard Code for Information Interchange: Amerikanische Code-Norm für den Informationsaustausch
ASIC	Application Specific Integrated Circuit: Anwender-Schaltkreis
ASUP	Asynchrones Unterprogramm
AUXFU	Auxiliary Function: Hilfsfunktion
AWL	Anweisungsliste
AWP	Anwenderprogramm

B	
BA	Betriebsart
BAG	Betriebsartengruppe
BCD	Binary Coded Decimals: Im Binärcode verschlüsselte Dezimalzahlen
BERO	Berührungsloser Näherungsschalter
BI	Binector Input
BICO	Binector Connector
BIN	Binary Files: Binärdateien
BIOS	Basic Input Output System
BKS	Basiskoordinatensystem
BO	Binector Output
BTSS	Bedientafelschnittstelle

C	
CAD	Computer-Aided Design
CAM	Computer-Aided Manufacturing
CC	Compile Cycle: Compile-Zyklen
CEC	Cross Error Compensation
CI	Connector Input
CF-Card	Compact Flash-Card

C	
CNC	Computerized Numerical Control: Computerunterstützte numerische Steuerung
CO	Connector Output
CoL	Certificate of License
COM	Communication
CPA	Compiler Projecting Data: Projektierdaten des Compilers
CRT	Cathode Ray Tube: Bildröhre
CSB	Central Service Board: PLC-Baugruppe
CU	Control Unit
CP	Communication Processor
CPU	Central Processing Unit: Zentrale Rechereinheit
CR	Carriage Return
CTS	Clear To Send: Meldung der Sendebereitschaft bei seriellen Daten-Schnittstellen
CUTCOM	Cutter Radius Compensation: Werkzeugradiuskorrektur

D	
DAU	Digital-Analog-Umwandler
DB	Datenbaustein (PLC)
DBB	Datenbaustein-Byte (PLC)
DBD	Datenbaustein-Doppelwort (PLC)
DBW	Datenbaustein-Wort (PLC)
DBX	Datenbaustein-Bit (PLC)
DDE	Dynamic Data Exchange
DDS	Drive Data Set: Antriebsdatensatz
DIN	Deutsche Industrie Norm
DIO	Data Input/Output: Datenübertragungs-Anzeige
DIR	Directory: Verzeichnis
DLL	Dynamic Link Library
DO	Drive Object
DPM	Dual Port Memory
DPR	Dual Port RAM
DRAM	Dynamischer Speicher (ungepuffert)
DRF	Differential Resolver Function: Differential-Drehmelder-Funktion (Handrad)
DRIVE-CLiQ	Drive Component Link with IQ
DRY	Dry Run: Probelaufvorschub
DSB	Decoding Single Block: Dekodierungseinzelsatz
DSC	Dynamic Servo Control / Dynamic Stiffness Control
DW	Datenwort
DWORD	Doppelwort (aktuell 32 Bit)

E	
E	Eingang
EES	Execution from External Storage
E/A	Ein-/Ausgabe
ENC	Encoder: Istwertgeber
EFP	Einfach Peripheriemodul (PLC–E/A–Baugruppe)
EGB	Elektronisch gefährdete Baugruppen/Bauelemente
EMV	Elektromagnetische Verträglichkeit
EN	Europäische Norm
ENC	Encoder: Istwertgeber
EnDat	Geberschnittstelle
EPROM	Erasable Programmable Read Only Memory: Löschbarer, elektrisch programmierbarer nur Lesespeicher
ePS Network Services	Dienste zur internetgestützten Maschinen-Fernwartung
EQN	Typbezeichnung eines Absolutwertgebers mit 2048 Sinussignalen/Umdrehung
ES	Engineering System
ESR	Erweitertes Stillsetzen und Rückziehen
ETC	ETC–Taste ">"; Erweiterung der Softkeyleiste im gleichen Menü

F	
FB	Funktionsbaustein (PLC)
FC	Function Call: Funktionsbaustein (PLC)
FEPROM	Flash–EPROM: Les– und schreibbarer Speicher
FIFO	First In First Out: Speicher, der ohne Adressangabe arbeitet und dessen Daten in derselben Reihenfolge gelesen werden, in der sie gespeichert wurden
FIPO	Feininterpolator
FPU	Floating Point Unit: Gleitpunkteinheit
FRK	Fräsradiuskorrektur
FST	Feed Stop: Vorschub Halt
FUP	Funktionsplan (Programmiermethode für PLC)
FW	Firmware

G	
GC	Global Control (PROFIBUS: Broadcast-Telegramm)
GDIR	Globaler Teileprogrammspeicher
GEO	Geometrie, z.B. Geometrieachse
GIA	Gear Interpolation Data: Getriebeinterpolationsdaten
GND	Signal Ground
GP	Grundprogramm (PLC)
GS	Getriebestufe
GSD	Gerätestammdatei zur Beschreibung eines PROFIBUS Slaves

Anhang

A.1 Liste der Abkürzungen

G	
GSDML	Generic Station Description Markup Language: XML-basierte Beschreibungssprache zur Erstellung einer GSD-Datei
GUD	Global User Data: Globale Anwenderdaten

H	
HEX	Kurzbezeichnung für hexadezimale Zahl
HIFu	Hilfsfunktion
HLA	Hydraulischer Linearantrieb
HMI	Human Machine Interface: SINUMERIK-Bedienoberfläche
HSA	Hauptspindeltrieb
HW	Hardware

I	
IBN	Inbetriebnahme
IKA	Interpolatorische Kompensation
IM	Interface-Modul: Anschaltungsbaugruppe
IMR	Interface-Modul Receive: Anschaltungsbaugruppe für Empfangsbetrieb
IMS	Interface-Modul Send: Anschaltungsbaugruppe für Sendebetrieb
INC	Increment: Schrittmaß
INI	Initializing Data: Initialisierungsdaten
IPO	Interpolator
ISA	International Standard Architecture
ISO	International Standard Organization

J	
JOG	Jogging: Einrichtbetrieb

K	
K_V	Verstärkungsfaktor des Regelkreises
K_P	Proportionalverstärkung
K_U	Übersetzungsverhältnis
KOP	Kontaktplan (Programmiermethode für PLC)

L	
LAI	Logic Machine Axis Image: Logisches Maschinenachsen-Abbild
LAN	Local Area Network
LCD	Liquid-Crystal Display: Flüssigkristallanzeige
LED	Light Emitting Diode: Leuchtdiode
LF	Line Feed

L	
LMS	Lagemesssystem
LR	Lageregler
LSB	Least Significant Bit: Niederwertigstes Bit
LUD	Local User Data: Anwenderdaten (lokal)

M	
MAC	Media Access Control
MAIN	Main program: Hauptprogramm (OB1, PLC)
MB	Megabyte
MCI	Motion Control Interface
MCIS	Motion-Control-Information-System
MCP	Machine Control Panel: Maschinensteuertafel
MD	Maschinendatum bzw. Maschinendaten
MDA	Manual Data Automatic: Handeingabe
MDS	Motor Data Set: Motordatensatz
MELDW	Meldungswort
MKS	Maschinenkoordinatensystem
MM	Motor Module
MPF	Main Program File: Hauptprogramm (NC)
MSTT	Maschinensteuertafel

N	
NC	Numerical Control: Numerische Steuerung mit Satzaufbereitung, Verfahrbereich usw.
NCU	Numerical Control Unit: Hardware-Einheit des NC
NRK	Bezeichnung des Betriebssystems des NC
NST	Nahtstellensignal
NURBS	Non-Uniform Rational B-Spline
NV	Nullpunktverschiebung
NX	Numerical Extension: Achserweiterungsbaugruppe

O	
OB	Organisationsbaustein in der PLC
OEM	Original Equipment Manufacturer
OP	Operation Panel: Bedieneinrichtung
OPI	Operation Panel Interface: Bedientafel-Anschaltung
OPT	Options: Optionen
OLP	Optical Link Plug: Busstecker für Lichtleiter
OSI	Open Systems Interconnection: Normung für Rechnerkommunikation

P	
PAA	Prozessabbild der Ausgänge
PAE	Prozessabbild der Eingänge
PC	Personal Computer
PCIN	Name der SW für den Datenaustausch mit der Steuerung
PCMCIA	Personal Computer Memory Card International Association: Speichersteckkarten-Normierung
PCU	PC Unit: PC-Box (Rechereinheit)
PG	Programmiergerät
PKE	Parameterkennung: Teil eines PKW
PKW	Parameterkennung: Wert (Parametrierteil eines PPO)
PLC	Programmable Logic Control: Anpass-Steuerung
PN	PROFINET
PNO	PROFIBUS-Nutzerorganisation
PO	POWER ON
POE	Programmorganisationseinheit
POS	Position/Positionieren
POSMO A	Positioning Motor Actuator: Positioniermotor
POSMO CA	Positioning Motor Compact AC: Komplette Antriebseinheit mit integrierter Leistungs- und Reglungsbaugruppe sowie Positioniereinheit und Programmspeicher; Wechselstrom-Einspeisung
POSMO CD	Positioning Motor Compact DC: wie CA, jedoch Gleichstromspeisung
POSMO SI	Positioning Motor Servo Integrated: Positioniermotor; Gleichstromspeisung
PPO	Parameter Prozessdaten Objekt ; Zyklisches Datentelegramm bei der Übertragung mit PROFIBUS-DP und Profil "Drehzahlveränderbare Antriebe"
PPU	Panel Processing Unit (zentrale Hardware einer Panel-basierten CNC-Steuerung z.B. SINUMERIK 828D)
PROFIBUS	Process Field Bus: Serieller Datenbus
PRT	Programmtest
PSW	Programmsteuerwort
PTP	Point to Point: Punkt zu Punkt
PUD	Program Global User Data: Programmglobale Anwendervariable
PZD	Prozessdaten: Prozessdatenteil eines PPO

Q	
QFK	Quadrantenfehler Kompensation

R	
RAM	Random Access Memory: Schreib-/Lese-Speicher
REF	Funktion Referenzpunkt anfahren
REPOS	Funktion Repositionieren
RISC	Reduced Instruction Set Computer: Prozessortyp mit kleinem Befehlssatz und schnellem Befehlsdurchsatz

R	
ROV	Rapid Override: Eingangskorrektur
RP	R-Parameter, Rechenparameter, vordefinierte Anwendervariable
RPA	R-Parameter Active: Speicherbereich in NC für R-Parameternummern
RPY	Roll Pitch Yaw: Drehungsart eines Koordinatensystems
RTL	Rapid Traverse Linear Interpolation: Lineare Interpolation bei Eilgangbewegung
RTS	Request To Send: Sendeteil einschalten, Steuersignal von seriellen Daten-Schnittstellen
RTCP	Real Time Control Protocol

S	
SA	Synchronaktion
SBC	Safe Break Control: Sichere Bremsenansteuerung
SBL	Single Block: Einzelsatz
SBR	Subroutine: Unterprogramm (PLC)
SD	Settingdatum bzw. Settingdaten
SDB	System Datenbaustein
SEA	Setting Data Active: Kennzeichnung (Dateityp) für Settingdaten
SERUPRO	Search-Run by Program Test: Satzsuchlauf via Programmtest
SFB	System Funktionsbaustein
SFC	System Function Call
SGE	Sicherheitsgerichteter Eingang
SGA	Sicherheitsgerichteter Ausgang
SH	Sicherer Halt
SIM	Single in Line Module
SK	Softkey
SKP	Skip: Funktion zum Ausblenden eines Teileprogrammsatzes
SLM	Synchroner Linearmotor
SM	Schrittmotor
SMC	Sensor Module Cabinet Mounted
SME	Sensor Module Externally Mounted
SMI	Sensor Module Integrated
SPF	Sub Program File: Unterprogramm (NC)
SPS	Speicherprogrammierbare Steuerung = PLC
SRAM	Statischer Speicher (gepuffert)
SRK	Schneidenradiuskorrektur
SRM	Synchron rotatorischer Motor
SSFK	Spindelsteigungsfehlerkompensation
SSI	Serial Synchron Interface: Serielle synchrone Schnittstelle
SSL	Satzsuchlauf
STW	Steuerwort
SUG	Scheibenumfangsgeschwindigkeit
SW	Software

Anhang

A.1 Liste der Abkürzungen

S	
SYF	System Files: Systemdateien
SYNACT	Synchronized Action: Synchronaktion

T	
TB	Terminal Board (SINAMICS)
TCP	Tool Center Point: Werkzeugspitze
TCP/IP	Transport Control Protocol / Internet Protocol
TCU	Thin Client Unit
TEA	Testing Data Active: Kennung für Maschinendaten
TIA	Totally Integrated Automation
TM	Terminal Module (SINAMICS)
TO	Tool Offset: Werkzeugkorrektur
TOA	Tool Offset Active: Kennzeichnung (Dateityp) für Werkzeugkorrekturen
TRANSMIT	Transform Milling Into Turning: Koordinatentransformation für Fräsbearbeitungen an einer Drehmaschine
TTL	Transistor-Transistor-Logik (Schnittstellen-Typ)
TZ	Technologiezyklus

U	
UFR	User Frame: Nullpunktverschiebung
UP	Unterprogramm
USB	Universal Serial Bus
USV	Unterbrechungsfreie Stromversorgung

V	
VDI	Interne Kommunikationsschnittstelle zwischen NC und PLC
VDI	Verein Deutscher Ingenieure
VDE	Verband Deutscher Elektrotechniker
VI	Voltage Input
VO	Voltage Output
VSA	Vorschubantrieb

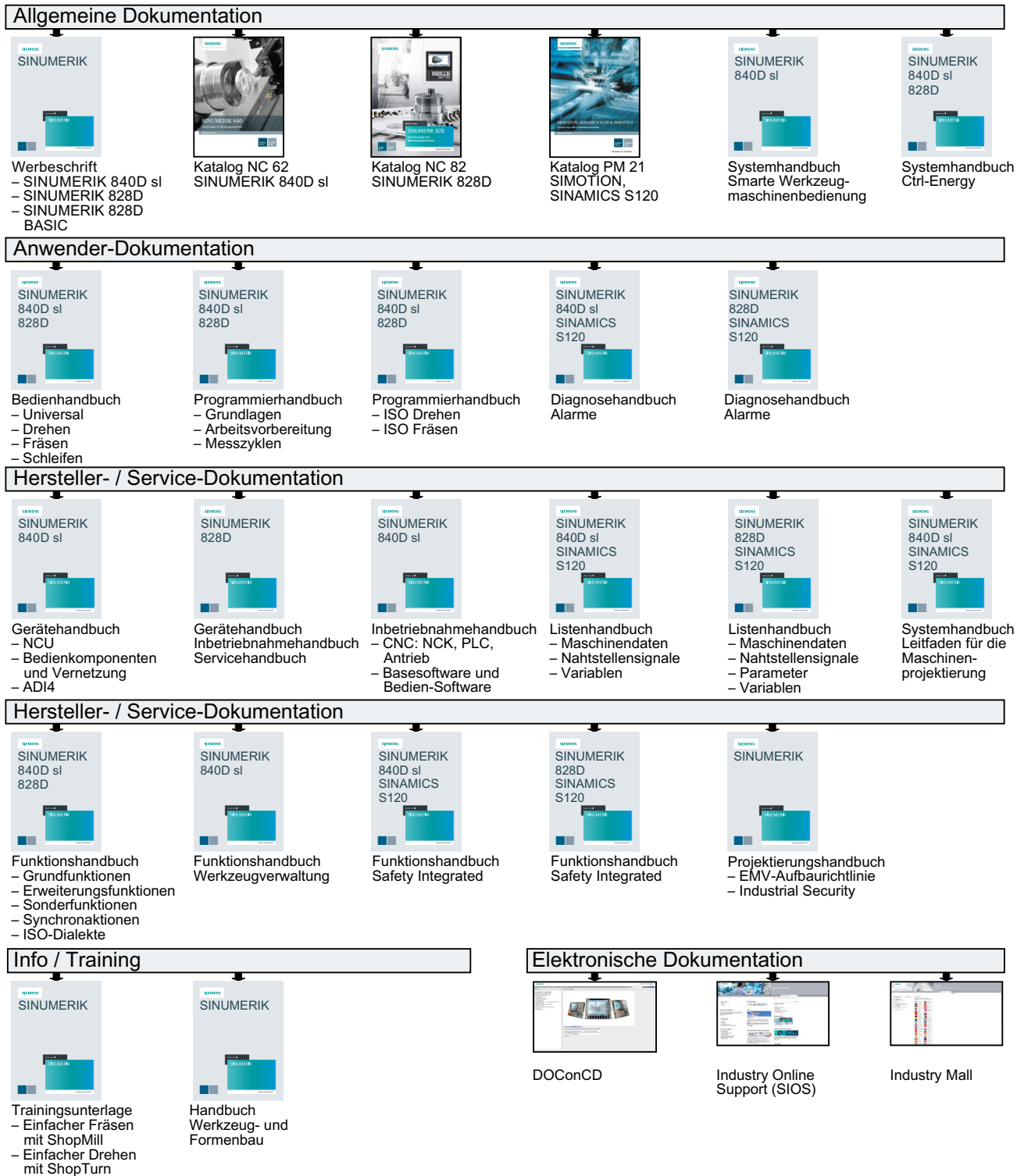
W	
WAB	Funktion Weiches An- und Abfahren
WKS	Werkstückkoordinatensystem
WKZ	Werkzeug
WLK	Werkzeuglängenkorrektur
WOP	Werkstatt-orientierte Programmierung
WPD	Work Piece Directory: Werkstückverzeichnis
WRK	Werkzeug-Radius-Korrektur

W	
WZ	Werkzeug
WZK	Werkzeugkorrektur
WZV	Werkzeugverwaltung
WZW	Werkzeugwechsel

X	
XML	Extensible Markup Language

Z	
ZOA	Zero Offset Active: Kennung für Nullpunktverschiebungen
ZSW	Zustandswort (des Antriebs)

A.2 Dokumentationsübersicht



Glossar

Absolutmaß

Angabe des Bewegungsziels einer Achsbewegung durch ein Maß, das sich auf den Nullpunkt des momentan gültigen Koordinatensystems bezieht. Siehe → Kettenmaß.

Achsadresse

Siehe → Achsname

Achsen

Die CNC-Achsen werden entsprechend ihres Funktionsumfangs abgestuft in:

- Achsen: interpolierende Bahnachsen
- Hilfsachsen: nicht interpolierende Zustell- und Positionierachsen mit achsspezifischem Vorschub. Hilfsachsen sind an der eigentlichen Bearbeitung nicht beteiligt, z. B. Werkzeugzubringer, Werkzeugmagazin.

Achsname

Zur eindeutigen Identifikation müssen alle Kanal- und → Maschinenachsen der Steuerung mit kanal- bzw. steuerungsweit eindeutigen Namen bezeichnet werden. Die → Geometrieachsen werden mit X, Y, Z benannt. Die um die Geometrieachsen drehenden → Rundachsen werden mit A, B, C benannt.

Adresse

Eine Adresse ist die Kennzeichnung für einen bestimmten Operanden oder Operandenbereich, z. B. Eingang, Ausgang usw.

Alarme

Alle → Meldungen und Alarme werden auf der Bedientafel im Klartext mit Datum und Uhrzeit und dem entsprechenden Symbol für das Löschkriterium angezeigt. Die Anzeige erfolgt getrennt nach Alarmen und Meldungen.

1. Alarme und Meldungen im Teileprogramm
Alarme und Meldungen können direkt aus dem Teileprogramm im Klartext zur Anzeige gebracht werden.
2. Alarme und Meldungen von PLC
Alarme- und Meldungen der Maschine können aus dem PLC-Programm im Klartext zur Anzeige gebracht werden. Dazu sind keine zusätzlichen Funktionsbaustein-Pakete notwendig.

Antrieb

Der Antrieb ist diejenige Einheit der CNC, welche die Drehzahl- und Momentenregelung aufgrund der Vorgaben der NC ausführt.

Anwenderdefinierte Variable

Anwender können für beliebige Nutzung im → Teileprogramm oder Datenbaustein (globale Anwenderdaten) anwenderdefinierte Variablen vereinbaren. Eine Definition enthält eine Datentypangabe und den Variablennamen. Siehe → Systemvariable.

Anwenderprogramm

Anwenderprogramme für Automatisierungssysteme S7-300 werden mit der Programmiersprache STEP 7 erstellt. Das Anwenderprogramm ist modular aufgebaut und besteht aus einzelnen Bausteinen.

Die grundlegenden Bausteintypen sind:

- Code-Bausteine
Diese Bausteine enthalten die STEP 7-Befehle.
- Datenbausteine
Diese Bausteine enthalten Konstanten und Variablen für das STEP 7-Programm.

Anwenderspeicher

Alle Programme und Daten wie Teileprogramme, Unterprogramme, Kommentare, Werkzeugkorrekturen, Nullpunktverschiebungen/Frames sowie Kanal- und Programmanwenderdaten können in den gemeinsamen CNC-Anwenderspeicher abgelegt werden.

Arbeitsfeldbegrenzung

Mit der Arbeitsfeldbegrenzung kann der Verfahrbereich der Achsen zusätzlich zu den Endschaltern eingeschränkt werden. Je Achse ist ein Wertepaar zur Beschreibung des geschützten Arbeitsraumes möglich.

Arbeitsraum

Dreidimensionaler Raum, in den die Werkzeugspitze aufgrund der Konstruktion der Werkzeugmaschine hineinfahren kann. Siehe → Schutzraum.

Arbeitsspeicher

Der Arbeitsspeicher ist ein RAM-Speicher in der → CPU, auf den der Prozessor während der Programmbearbeitung auf das Anwenderprogramm zugreift.

Archivieren

Auslesen von Dateien und/oder Verzeichnissen auf ein **externes** Speichergerät.

Asynchrones Unterprogramm

Teileprogramm, das asynchron (unabhängig) zum aktuellen Programmzustand durch ein Interruptsignal (z. B. Signal "schneller NC-Eingang") gestartet werden kann.

Automatik

Betriebsart der Steuerung (Satzfolgebetrieb nach DIN): Betriebsart bei NC-Systemen, in der ein → Teileprogramm angewählt und kontinuierlich abgearbeitet wird.

Bahnachse

Bahnachsen sind alle Bearbeitungsachsen des → Kanals, die vom → Interpolator so geführt werden, dass sie gleichzeitig starten, beschleunigen, stoppen und den Endpunkt erreichen.

Bahngeschwindigkeit

Die maximal programmierbare Bahngeschwindigkeit ist abhängig von der Eingabefineinheit. Bei einer Auflösung von beispielsweise 0,1 mm beträgt die maximal programmierbare Bahngeschwindigkeit 1000 m/min.

Bahnsteuerbetrieb

Ziel des Bahnsteuerbetriebes ist es, ein größeres Abbremsen der → Bahnachsen an den Teileprogramm-Satzgrenzen zu vermeiden und mit möglichst gleicher Bahngeschwindigkeit in den nächsten Satz zu wechseln.

Bahnvorschub

Bahnvorschub wirkt auf → Bahnachsen. Er stellt die geometrische Summe der Vorschübe der beteiligten → Geometrieachsen dar.

Basisachse

Achse, deren Soll- oder Istwert für die Berechnung eines Kompensationswertes herangezogen wird.

Basiskoordinatensystem

Kartesisches Koordinatensystem, wird durch Transformation auf das Maschinenkoordinatensystem abgebildet.

Im → Teileprogramm verwendet der Programmierer Achsnamen des Basiskoordinatensystems. Es besteht, wenn keine → Transformation aktiv ist, parallel zum → Maschinenkoordinatensystem. Der Unterschied zu diesem liegt in den → Achsnamen.

Baudrate

Geschwindigkeit bei der Datenübertragung (Bit/s).

Baustein

Als Bausteine werden alle Dateien bezeichnet, die für die Programmerstellung und Programmverarbeitung benötigt werden.

Bearbeitungskanal

Über eine Kanalstruktur können durch parallele Bewegungsabläufe Nebenzeiten verkürzt werden, z. B. Verfahren eines Ladeportals simultan zur Bearbeitung. Ein CNC-Kanal ist dabei als eigene CNC-Steuerung mit Dekodierung, Satzaufbereitung und Interpolation anzusehen.

Bedienoberfläche

Die Bedienoberfläche (BOF) ist das Anzeigemedium einer CNC-Steuerung in Gestalt eines Bildschirms. Sie ist mit horizontalen und vertikalen Softkeys gestaltet.

Beschleunigung mit Ruckbegrenzung

Zur Erzielung eines optimalen Beschleunigungsverhaltens an der Maschine bei gleichzeitiger Schonung der Mechanik kann im Bearbeitungsprogramm zwischen sprunghafter Beschleunigung und stetiger (ruckfreier) Beschleunigung umgeschaltet werden.

Betriebsart

Ablaufkonzept für den Betrieb einer SINUMERIK-Steuerung. Es sind die Betriebsarten → Jog, → MDA, → Automatik definiert.

Betriebsartengruppe

Technologisch zusammengehörige Achsen und Spindeln können zu einer Betriebsartengruppe (BAG) zusammengefasst werden. Achsen/Spindeln einer BAG können von einem oder mehreren → Kanälen gesteuert werden. Den Kanälen der BAG ist immer die gleiche → Betriebsart zugeordnet.

Bezeichner

Die Wörter nach DIN 66025 werden durch Bezeichner (Namen) für Variable (Rechenvariable, Systemvariable, Anwendervariable), für Unterprogramme, für Schlüsselwörter und Wörter mit mehreren Adressbuchstaben ergänzt. Diese Ergänzungen kommen in der Bedeutung den Wörtern beim Satzaufbau gleich. Bezeichner müssen eindeutig sein. Derselbe Bezeichner darf nicht für verschiedene Objekte verwendet werden.

Booten

Laden des Systemprogramms nach Power On.

C-Achse

Achse, um die eine gesteuerte Drehbewegung und Positionierung mit der Werkstückspindel erfolgt.

CNC

Siehe → NC

Computerized Numerical Control: umfasst die Komponenten → NC, → PLC, HMI, → COM.

CNC

Siehe → NC

Computerized Numerical Control: umfasst die Komponenten → NC, → PLC, HMI, → COM.

COM

Komponente der NC-Steuerung zur Durchführung und Koordination von Kommunikation.

CPU

Central Processing Unit, siehe → Speicherprogrammierbare Steuerung

C-Spline

Der C-Spline ist der bekannteste und am meisten verwendete Spline. Die Übergänge an den Stützpunkten sind tangential- und krümmungstetig. Es werden Polynome 3. Grades verwendet.

Datenbaustein

1. Dateneinheit der → PLC, auf die → HIGHSTEP-Programme zugreifen können.
2. Dateneinheit der → NC: Datenbausteine enthalten Datendefinitionen für globale Anwenderdaten. Die Daten können bei der Definition direkt initialisiert werden.

Datenübertragungsprogramm PCIN

PCIN ist ein Hilfsprogramm zum Senden und Empfangen von CNC-Anwenderdaten über die serielle Schnittstelle, wie z. B. Teileprogramme, Werkzeugkorrekturen etc. Das PCIN-Programm ist unter MS-DOS auf Standard-Industrie-PCs lauffähig.

Datenwort

Zwei Byte große Dateneinheit innerhalb eines → Datenbausteins.

Diagnose

1. Bedienbereich der Steuerung
2. Die Steuerung besitzt sowohl ein Selbstdiagnose-Programm als auch Testhilfen für den Service: Status-, Alarm- und Serviceanzeigen

DRF

Differential Resolver Function: NC-Funktion, die in Verbindung mit einem elektronischen Handrad eine inkrementale Nullpunktverschiebung im Automatik-Betrieb erzeugt.

Editor

Der Editor ermöglicht das Erstellen, Ändern, Ergänzen, Zusammenschieben und Einfügen von Programmen/Texten/Programmsätzen.

Eilgang

Schnellste Verfahrgeschwindigkeit einer Achse. Sie wird z. B. verwendet, wenn das Werkzeug aus einer Ruhestellung an die → Werkstückkontur herangefahren oder von der Werkstückkontur zurückgezogen wird. Die Eilganggeschwindigkeit wird maschinenspezifisch über Maschinendatum eingestellt.

Externe Nullpunktverschiebung

Von der → PLC vorgegebene Nullpunktverschiebung.

Fertigteilkontur

Kontur des fertig bearbeiteten Werkstücks. Siehe → Rohteil.

Festpunkt-Anfahren

Werkzeugmaschinen können feste Punkte wie Werkzeugwechsellpunkt, Beladepunkt, Palettenwechsellpunkt etc. definiert anfahren. Die Koordinaten dieser Punkte sind in der Steuerung hinterlegt. Die Steuerung verfährt die betroffenen Achsen, wenn möglich, im → Eilgang.

Frame

Ein Frame stellt eine Rechenvorschrift dar, die ein kartesisches Koordinatensystem in ein anderes kartesisches Koordinatensystem überführt. Ein Frame enthält die Komponenten → Nullpunktverschiebung, → Rotation, → Skalierung, → Spiegelung.

Führungssachse

Die Führungssachse ist die → Gantry-Achse, die aus Sicht des Bedieners und des Programmierers vorhanden und damit entsprechend wie eine normale NC-Achse beeinflussbar ist.

Genauhalt

Bei programmierter Genauhalt-Anweisung wird die in einem Satz angegebene Position genau und ggf. sehr langsam angefahren. Zur Reduktion der Annäherungszeit werden für Eilgang und Vorschub → Genauhaltsgrenzen definiert.

Genauhaltgrenze

Erreichen alle Bahnachsen ihre Genauhaltgrenze, so verhält sich die Steuerung als habe sie einen Zielpunkt exakt erreicht. Es erfolgt Satzweilerschaltung des → Teileprogramms.

Geometrie

Beschreibung eines → Werkstücks im → Werkstückkoordinatensystem.

Geometrieachse

Die Geometrieachsen bilden das 2- bzw. 3-dimensionale → Werkstückkoordinatensystem in dem in → Teileprogrammen die Geometrie des Werkstücks programmiert wird.

Geradeninterpolation

Das Werkzeug wird auf einer Geraden zum Zielpunkt verfahren und dabei das Werkstück bearbeitet.

Geschwindigkeitsführung

Um bei Verfahrbewegungen um sehr kleine Beträge je Satz eine akzeptable Verfahrgeschwindigkeit erreichen zu können, kann vorausschauende Auswertung über mehrere Sätze (→ Look Ahead) eingestellt werden.

Gewindebohren ohne Ausgleichsfutter

Mit dieser Funktion können Gewinde ohne Ausgleichsfutter gebohrt werden. Durch das interpolierende Verfahren der Spindel als Rundachse und der Bohrachse werden Gewinde exakt auf Endbohrtiefe geschnitten, z. B. Sacklochgewinde (Voraussetzung: Achsbetrieb der Spindel).

Gleichlaufachse

Die Gleichlaufachse ist die → Gantry-Achse, deren Sollposition stets von der Verfahrbewegung der → Führungssachse abgeleitet und damit synchron verfahren wird. Aus Sicht des Bedieners und des Programmierers ist die Gleichlaufachse "nicht vorhanden".

Grenzdrehzahl

Maximale/minimale (Spindel-)Drehzahl: Durch Vorgaben von Maschinendaten, der → PLC oder → Settingdaten kann die maximale Drehzahl einer Spindel begrenzt sein.

Hauptprogramm

Die Bezeichnung Hauptprogramm stammt noch aus der Zeit, als Teileprogramm fest in Haupt- und → Unterprogramme unterteilt waren. Diese feste Einteilung besteht mit der heutigen SINUMERIK NC-Sprache nicht mehr. Prinzipiell kann jedes Teileprogramm im Kanal angewählt und gestartet werden. Es läuft dann in der → Programmebene 0 (Hauptprogramm-Ebene) ab. Im Hauptprogramm können weitere Teileprogramme oder → Zyklen als Unterprogramme aufgerufen werden

Hauptsatz

Durch ":" eingeleiteter Satz, der alle Angaben enthält, um den Arbeitsablauf in einem → Teileprogramm starten zu können.

HIGHSTEP

Zusammenfassung der Programmiermöglichkeiten für die → PLC des Systems AS300/AS400.

Hilfsfunktionen

Mit Hilfsfunktionen können in → Teileprogrammen → Parameter an die → PLC übergeben werden, die dort vom Maschinenhersteller definierte Reaktionen auslösen.

Hochsprache CNC

Die Hochsprache dient zum Schreiben von NC-Programmen, → Synchronaktionen und → Zyklen. Sie bietet: Kontrollstrukturen, → Anwenderdefinierte Variable, → Systemvariable, → Makrotechnik.

HW-Konfig

SIMATIC S7-Tool zum Konfigurieren und Parametrieren von Hardware-Komponenten innerhalb eines S7-Projekts.

Interpolator

Logische Einheit des → NC, die nach Angaben von Zielpositionen im Teileprogramm Zwischenwerte für die in den einzelnen Achsen zu fahrenden Bewegungen bestimmt.

Interpolatorische Kompensation

Über interpolatorische Kompensationen wie → Spindelsteigungsfehler-, Durchhang-, Winkligkeits- und Temperaturkompensation werden mechanische Fehler der Maschine kompensiert.

Interruptroutine

Interruptroutinen sind spezielle → Unterprogramme, die durch Ereignisse (externe Signale) vom Bearbeitungsprozess gestartet werden können. Ein in Abarbeitung befindlicher Teileprogrammsatz wird abgebrochen, die Unterbrechungsposition der Achsen wird automatisch gespeichert.

JOG

Betriebsart der Steuerung (Einrichtebetrieb): In der Betriebsart JOG kann die Maschine eingerichtet werden. Einzelne Achsen und Spindeln können über die Richtungstasten im Tippbetrieb verfahren werden. Weitere Funktionen in der Betriebsart JOG sind das → Referenzpunktfahren, → Repos sowie → Preset (Istwert setzen).

Kanal

Ein Kanal ist dadurch gekennzeichnet, dass er unabhängig von anderen Kanälen ein → Teileprogramm abarbeiten kann. Ein Kanal steuert exklusiv die ihm zugeordneten Achsen und Spindeln. Teileprogrammabläufe verschiedener Kanäle können durch → Synchronisation koordiniert werden.

Kettenmaß

Auch Inkrementmaß: Angabe eines Bewegungsziels einer Achse durch eine zu verfahrenende Wegstrecke und Richtung bezogen auf einen bereits erreichten Punkt. Siehe → Absolutmaß.

Kompensationsachse

Achse, deren Soll- oder Istwert durch den Kompensationswert modifiziert wird.

Kompensationstabelle

Tabelle von Stützpunkten. Sie liefert für ausgewählte Positionen der Basisachse die Kompensationswerte der Kompensationsachse.

Kompensationswert

Differenz zwischen der durch den Messgeber gemessenen Achsposition und der gewünschten, programmierten Achsposition.

Kontur

Umriss des → Werkstücks

Konturüberwachung

Als Maß für die Konturtreue wird der Schleppfehler innerhalb eines definierbaren Toleranzbandes überwacht. Ein unzulässig hoher Schleppfehler kann sich z. B. durch

Überlastung des Antriebs ergeben. In diesem Fall kommt es zu einem Alarm und die Achsen werden stillgesetzt.

Koordinatensystem

Siehe → Maschinenkoordinatensystem, → Werkstückkoordinatensystem

Korrekturspeicher

Datenbereich in der Steuerung, in dem Werkzeugkorrekturdaten hinterlegt sind.

Kreisinterpolation

Das → Werkzeug soll zwischen festgelegten Punkten der Kontur mit einem gegebenen Vorschub auf einem Kreis fahren und dabei das Werkstück bearbeiten.

Krümmung

Die Krümmung k einer Kontur ist das Inverse des Radius r des anschmiegenden Kreises in einem Konturpunkt ($k = 1/r$).

KÜ

Übersetzungsverhältnis

KV

Kreisverstärkungsfaktor, regelungstechnische Größe eines Regelkreises

Ladespeicher

Der Ladespeicher ist bei der CPU 314 der → SPS gleich dem → Arbeitsspeicher.

Linearachse

Die Linearachse ist eine Achse, welche im Gegensatz zur Rundachse eine Gerade beschreibt.

Look Ahead

Mit der Funktion **Look Ahead** wird durch das "Vorausschauen" über eine parametrierbare Anzahl von Verfahrssätzen ein Optimum an Bearbeitungsgeschwindigkeit erzielt.

Losekompensation

Ausgleich einer mechanischen Maschinenlose, z. B. Umkehrlose bei Kugelrollspindeln. Für jede Achse kann die Losekompensation getrennt eingegeben werden.

Makrotechnik

Zusammenfassung einer Menge von Anweisungen unter einem Bezeichner. Der Bezeichner repräsentiert im Programm die Menge der zusammengefassten Anweisungen.

Maschinenachsen

In der Werkzeugmaschine physikalisch existierende Achsen.

Maschinenfestpunkt

Durch die Werkzeugmaschine eindeutig definierter Punkt, z. B. Maschinen-Referenzpunkt.

Maschinenkoordinatensystem

Koordinatensystem, das auf die Achsen der Werkzeugmaschine bezogen ist.

Maschinennullpunkt

Fester Punkt der Werkzeugmaschine, auf den sich alle (abgeleiteten) Messsysteme zurückführen lassen.

Maschinensteuertafel

Bedientafel der Werkzeugmaschine mit den Bedienelementen Tasten, Drehschalter usw. und einfachen Anzeigeelementen wie LEDs. Sie dient der unmittelbaren Beeinflussung der Werkzeugmaschine über die PLC.

Maßangabe metrisch und inch

Im Bearbeitungsprogramm können Positions- und Steigungswerte in inch programmiert werden. Unabhängig von der programmierbaren Maßangabe (*G70/G71*) wird die Steuerung auf ein Grundsystem eingestellt.

Masse

Als Masse gilt die Gesamtheit aller untereinander verbundenen inaktiven Teile eines Betriebsmittels, die auch im Fehlerfall keine gefährliche Berührungsspannung annehmen können.

MDA

Betriebsart der Steuerung: Manual Data Automatic. In der Betriebsart MDA können einzelne Programmsätze oder Satzfolgen ohne Bezug auf ein Haupt- oder Unterprogramm eingegeben und anschließend über die Taste NC-Start sofort ausgeführt werden.

Meldungen

Alle im Teileprogramm programmierten Meldungen und vom System erkannte → Alarme werden auf der Bedientafel im Klartext mit Datum und Uhrzeit und dem entsprechenden Symbol für das Löschkriterium angezeigt. Die Anzeige erfolgt getrennt nach Alarmen und Meldungen.

Metrisches Messsystem

Genormtes System von Einheiten: für Längen z. B. mm (Millimeter), m (Meter).

NC

Numerical Control Komponente der → CNC, die → Teileprogramme abarbeitet und die Bewegungsvorgänge der Werkzeugmaschine koordiniert.

Nebensatz

Durch "N" eingeleiteter Satz mit Informationen für einen Arbeitsschritt, z. B. eine Positionsangabe.

Netz

Ein Netz ist die Verbindung von mehreren S7-300 und weiteren Endgeräten, z. B. einem PG, über → Verbindungskabel. Über das Netz erfolgt ein Datenaustausch zwischen den angeschlossenen Geräten.

NRK

Numeric Robotic Kernel (Betriebssystem des → NC)

Nullpunktverschiebung

Vorgabe eines neuen Bezugspunkts für ein Koordinatensystem durch Bezug auf einen bestehenden Nullpunkt und ein → Frame.

1. Einstellbar
Es steht eine projektierbare Anzahl von einstellbaren Nullpunktverschiebungen für jede CNC-Achse zur Verfügung. Die über G-Befehle anwählbaren Verschiebungen sind alternativ wirksam.
2. Extern
Zusätzlich zu allen Verschiebungen, die die Lage des Werkstücknullpunkts festlegen, kann eine externe Nullpunktverschiebung durch Handrad (DRF-Verschiebung) oder von der PLC überlagert werden.
3. Programmierbar
Mit der Anweisung TRANS sind für alle Bahn- und Positionierachsen Nullpunktverschiebungen programmierbar.

NURBS

Die steuerungsinterne Bewegungsführung und Bahninterpolation wird auf Basis von NURBS (**N**on **U**niform **R**ational **B**-**S**plines) durchgeführt. Damit steht steuerungsintern für alle Interpolationen ein einheitliches Verfahren zur Verfügung.

OEM

Für Maschinenhersteller, die ihre eigene Bedienoberfläche erstellen oder technologiespezifische Funktionen in die Steuerung einbringen wollen, sind Freiräume für individuelle Lösungen (OEM-Applikationen) vorgesehen.

Orientierter Spindelhalt

Halt der Werkstückspindel in vorgegebener Winkellage, z. B. um an bestimmter Stelle eine Zusatzbearbeitung vorzunehmen.

Override

Manuelle bzw. programmierbare Eingriffsmöglichkeit, die es dem Bediener gestattet, programmierte Vorschübe oder Drehzahlen zu überlagern, um sie einem bestimmten Werkstück oder Werkstoff anzupassen.

Peripheriebaugruppe

Peripheriebaugruppen stellen die Verbindung zwischen CPU und Prozess her.

Peripheriebaugruppen sind:

- → Digital-Ein-/Ausgabebaugruppen
- → Analog-Ein-/Ausgabebaugruppen
- → Simulatorbaugruppen

PLC

Programmable Logic Control: → Speicherprogrammierbare Steuerung. Komponente der → NC: Anpass-Steuerung zur Bearbeitung der Kontroll-Logik der Werkzeugmaschine.

PLC-Programmierung

Die PLC wird mit der Software **STEP 7** programmiert. Die Programmiersoftware STEP 7 basiert auf dem Standardbetriebssystem **WINDOWS** und enthält die Funktionen der STEP 5 - Programmierung mit innovativen Weiterentwicklungen.

PLC-Programmspeicher

SINUMERIK 840D sl: Im PLC-Anwenderspeicher werden das PLC-Anwenderprogramm und die Anwenderdaten gemeinsam mit dem PLC-Grundprogramm abgelegt.

Polarkoordinaten

Koordinatensystem, das die Lage eines Punktes in einer Ebene durch seinen Abstand vom Nullpunkt und den Winkel festlegt, den der Radiusvektor mit einer festgelegten Achse bildet.

Polynom-Interpolation

Mit der Polynom-Interpolation können die unterschiedlichsten Kurvenverläufe erzeugt werden, wie **Gerade-, Parabel-, Potenzfunktionen** (SINUMERIK 840D sl).

Positionierachse

Achse, die eine Hilfsbewegung an einer Werkzeugmaschine ausführt. (z. B. Werkzeugmagazin, Palettentransport). Positionierachsen sind Achsen, die nicht mit den → Bahnachsen interpolieren.

Programmbaustein

Programmbausteine enthalten die Haupt- und Unterprogramme der → Teileprogramme.

Programmebene

Ein im Kanal gestartetes Teileprogramm läuft als → Hauptprogramm auf Programmebene 0 (Hauptprogramm-Ebene). Jedes im Hauptprogramm aufgerufene Teileprogramm läuft als → Unterprogramm auf einer eigenen Programmebene 1 ... n.

Programmierbare Arbeitsfeldbegrenzung

Begrenzung des Bewegungsraumes des Werkzeugs auf einen durch programmierte Begrenzungen definierten Raum.

Programmierbare Frames

Mit programmierbaren → Frames können dynamisch im Zuge der Teileprogramm-Abarbeitung neue Koordinatensystem-Ausgangspunkte definiert werden. Es wird unterschieden nach absoluter Festlegung anhand eines neuen Frames und additiver Festlegung unter Bezug auf einen bestehenden Ausgangspunkt.

Programmierschlüssel

Zeichen und Zeichenfolgen, die in der Programmiersprache für → Teileprogramme eine festgelegte Bedeutung haben.

Pufferbatterie

Die Pufferbatterie gewährleistet, dass das → Anwenderprogramm in der → CPU netzausfallsicher hinterlegt ist und festgelegte Datenbereiche und Merker, Zeiten und Zähler remanent gehalten werden.

Quadrantenfehlerkompensation

Konturfehler an Quadrantenübergängen, die durch wechselnde Reibverhältnisse an Führungsbahnen entstehen, sind mit der Quadrantenfehlerkompensation weitgehend eliminierbar. Die Parametrierung der Quadrantenfehlerkompensation erfolgt durch einen Kreisformtest.

Referenzpunkt

Punkt der Werkzeugmaschine, auf den sich das Messsystem der → Maschinenachsen bezieht.

Rohteil

Teil, mit dem die Bearbeitung eines Werkstücks begonnen wird.

Rotation

Komponente eines → Frames, die eine Drehung des Koordinatensystems um einen bestimmten Winkel definiert.

R-Parameter

Rechenparameter, kann vom Programmierer des → Teileprogramms für beliebige Zwecke im Programm gesetzt oder abgefragt werden.

Rundachse

Rundachsen bewirken eine Werkstück- oder Werkzeugdrehung in eine vorgegebene Winkellage.

Rundungsachse

Rundungsachsen bewirken eine Werkstück- oder Werkzeugdrehung in eine einem Teilungsraster entsprechende Winkellage. Beim Erreichen eines Rasters ist die Rundungsachse "in Position".

Satzsuchlauf

Zum Austesten von Teileprogrammen oder nach einem Abbruch der Bearbeitung kann über die Funktion "Satzsuchlauf" eine beliebige Stelle im Teileprogramm angewählt werden, an der die Bearbeitung gestartet oder fortgesetzt werden soll.

Schlüsselschalter

Der Schlüsselschalter auf der → Maschinensteuertafel besitzt 4 Stellungen, die vom Betriebssystem der Steuerung mit Funktionen belegt sind. Zum Schlüsselschalter gehören drei verschiedenfarbige Schlüssel, die in den angegebenen Stellungen abgezogen werden können.

Schlüsselwörter

Wörter mit festgelegter Schreibweise, die in der Programmiersprache für → Teileprogramme eine definierte Bedeutung haben.

Schneidenradiuskorrektur

Bei der Programmierung einer Kontur wird von einem spitzen Werkzeug ausgegangen. Da dies in der Praxis nicht realisierbar ist, wird der Krümmungsradius des eingesetzten Werkzeugs der Steuerung angegeben und von dieser berücksichtigt. Dabei wird der Krümmungsmittelpunkt um den Krümmungsradius verschoben äquidistant um die Kontur geführt.

Schnellabheben von der Kontur

Beim Eintreffen eines Interrupts kann über das CNC-Bearbeitungsprogramm eine Bewegung eingeleitet werden, die ein schnelles Abheben des Werkzeugs von der gerade bearbeiteten Werkstückkontur ermöglicht. Zusätzlich kann der Rückzugwinkel und der Betrag des Weges parametrisiert werden. Nach dem Schnellabheben kann zusätzlich eine Interruptroutine ausgeführt werden.

Schnelle digitale Ein-/Ausgänge

Über die digitalen Eingänge können z. B. schnelle CNC-Programmroutinen (Interruptroutinen) gestartet werden. Über die digitalen CNC-Ausgänge können schnelle, programmgesteuerte Schalfunktionen ausgelöst werden.

Schrägenbearbeitung

Bohr- und Fräsbearbeitungen an Werkstückflächen, die nicht in den Koordinatenebenen der Maschine liegen, können mit Unterstützung der Funktion "Schrägenbearbeitung" komfortabel ausgeführt werden.

Schraubenlinien-Interpolation

Die Schraubenlinien-Interpolation eignet sich besonders zum einfachen Herstellen von Innen- oder Außengewinden mit Formfräsern und zum Fräsen von Schmiernuten.

Dabei setzt sich die Schraubenlinie aus zwei Bewegungen zusammen:

- Kreisbewegung in einer Ebene
- Linearbewegung senkrecht zu dieser Ebene

Schrittmaß

Verfahrweglängenangabe über Inkrementanzahl (Schrittmaß). Inkrementanzahl kann als → Settingdatum hinterlegt sein bzw. durch entsprechend beschriftete Tasten 10, 100, 1000, 10000 gewählt werden.

Schutzraum

Dreidimensionaler Raum innerhalb des → Arbeitsraumes, in den die Werkzeugspitze nicht hineinreichen darf.

Settingdaten

Daten, die Eigenschaften der Werkzeugmaschine auf durch die Systemsoftware definierte Weise der NC-Steuerung mitteilen.

Sicherheitsfunktionen

Die Steuerung enthält ständig aktive Überwachungen, die Störungen in der → CNC, der Anpass-Steuerung (→ PLC) und der Maschine so frühzeitig erkennen, dass Schäden an Werkstück, Werkzeug oder Maschine weitgehend ausgeschlossen werden. Im Störfall wird der Bearbeitungsablauf unterbrochen und die Antriebe werden stillgesetzt, die Störungsursache gespeichert und als Alarm angezeigt. Gleichzeitig wird der PLC mitgeteilt, dass ein CNC-Alarm ansteht.

Skalierung

Komponente eines → Frames, die achsspezifische Maßstabsveränderungen bewirkt.

Softkey

Taste, deren Beschriftung durch ein Feld im Bildschirm repräsentiert wird, das sich dynamisch der aktuellen Bediensituation anpasst. Die frei belegbaren Funktionstasten (Softkeys) werden softwaremäßig definierten Funktionen zugeordnet.

Software-Endschalter

Software-Endschalter begrenzen den Verfahrbereich einer Achse und verhindern ein Auffahren des Schlittens auf die Hardware-Endschalter. Je Achse sind 2 Wertepaare vorgebar, die getrennt über die → PLC aktiviert werden können.

Speicherprogrammierbare Steuerung

Speicherprogrammierbare Steuerungen (SPS) sind elektronische Steuerungen, deren Funktion als Programm im Steuerungsgerät gespeichert ist. Aufbau und Verdrahtung des Gerätes hängen also nicht von der Funktion der Steuerung ab. Die speicherprogrammierbare Steuerung hat die Struktur eines Rechners; sie besteht aus CPU (Zentralbaugruppe) mit Speicher, Ein-/Ausgabebaugruppen und internem Bus-System. Die Peripherie und die Programmiersprache sind auf die Belange der Steuerungstechnik ausgerichtet.

Spiegelung

Bei Spiegelung werden die Vorzeichen der Koordinatenwerte einer Kontur bezüglich einer Achse vertauscht. Es kann bezüglich mehrerer Achsen zugleich gespiegelt werden.

Spindelsteigungsfehler-Kompensation

Ausgleich mechanischer Ungenauigkeiten einer am Vorschub beteiligten Kugelrollspindel durch die Steuerung anhand von hinterlegten Messwerten der Abweichungen.

Spline-Interpolation

Mit der Spline-Interpolation kann die Steuerung aus nur wenigen vorgegebenen Stützpunkten einer Sollkontur einen glatten Kurvenverlauf erzeugen.

Standardzyklen

Für häufig wiederkehrende Bearbeitungsaufgaben stehen Standardzyklen zur Verfügung:

- Für die Technologie Bohren/Fräsen
- Für die Technologie Drehen

Im Bedienbereich "Programm" werden unter dem Menü "Zyklusunterstützung" die zur Verfügung stehenden Zyklen aufgelistet. Nach Anwahl des gewünschten Bearbeitungszyklus werden die notwendigen Parameter für die Wertzuweisung im Klartext angezeigt.

Synchronachsen

Synchronachsen benötigen für ihren Weg die gleiche Zeit wie die Geometrieachsen für ihren Bahnweg.

Synchronaktionen

1. Hilfsfunktionsausgabe
Während der Werkstückbearbeitung können aus dem CNC-Programm heraus technologische Funktionen (→ Hilfsfunktionen) an die PLC ausgegeben werden. Über diese Hilfsfunktionen werden beispielsweise Zusatzeinrichtungen der Werkzeugmaschine gesteuert, wie Pinole, Greifer, Spannfutter etc.
2. Schnelle Hilfsfunktionsausgabe
Für zeitkritische Schaltfunktionen können die Quittierungszeiten für die → Hilfsfunktionen minimiert und unnötige Haltepunkte im Bearbeitungsprozess vermieden werden.

Synchronisation

Anweisungen in → Teileprogrammen zur Koordination der Abläufe in verschiedenen → Kanälen an bestimmten Bearbeitungsstellen.

Systemspeicher

Der Systemspeicher ist ein Speicher in der CPU, in der folgende Daten abgelegt werden:

- Daten, die das Betriebssystem benötigt
- Die Operanden Zeiten, Zähler, Merker

Systemvariable

Ohne Zutun des Programmierers eines → Teileprogramms existierende Variable. Sie ist definiert durch einen Datentyp und dem Variablennamen, der durch das Zeichen \$ eingeleitet wird. Siehe → Anwenderdefinierte Variable.

Teileprogramm

Folge von Anweisungen an die NC-Steuerung, die insgesamt die Erzeugung eines bestimmten → Werkstücks bewirken. Ebenso Vornahme einer bestimmten Bearbeitung an einem gegebenen → Rohteil.

Teileprogrammsatz

Teil eines → Teileprogramms, durch Line Feed abgegrenzt. Es werden → Hauptsätze und → Nebensätze unterschieden.

Teileprogrammverwaltung

Die Teileprogrammverwaltung kann nach → Werkstücken organisiert werden. Die Größe des Anwenderspeichers bestimmt die Anzahl der zu verwaltenden Programme und Daten. Jede Datei (Programme und Daten) kann mit einem Namen von maximal 24 alphanumerischen Zeichen versehen werden.

Text-Editor

Siehe → Editor

TOA-Bereich

Der TOA-Bereich umfasst alle Werkzeug- und Magazindaten. Standardmäßig fällt der Bereich bzgl. der Reichweite der Daten mit dem Bereich → Kanal zusammen. Über Maschinendaten kann jedoch festgelegt werden, dass sich mehrere Kanäle eine → TOA-Einheit teilen, so dass diesen Kanälen dann gemeinsame WZV-Daten zur Verfügung stehen.

TOA-Einheit

Jeder → TOA-Bereich kann mehrere TOA-Einheiten enthalten. Die Anzahl der möglichen TOA-Einheiten wird über die maximale Anzahl aktiver → Kanäle begrenzt. Eine TOA-Einheit umfasst genau einen WZ-Daten-Baustein und einen Magazindaten-Baustein. Zusätzlich kann noch ein WZ-Trägerdaten-Baustein enthalten sein (optional).

Transformation

Additive oder absolute Nullpunktverschiebung einer Achse.

Unterprogramm

Die Bezeichnung Unterprogramm stammt noch aus der Zeit, als Teileprogramm fest in → Haupt- und Unterprogramme unterteilt waren. Diese feste Einteilung besteht mit der heutigen SINUMERIK NC-Sprache nicht mehr. Prinzipiell kann jedes Teileprogramm oder jeder → Zyklus innerhalb eines anderen Teileprogramms als Unterprogramm aufgerufen werden. Es läuft dann in der nächsten → Programmebene (x+1) (Unterprogrammebene (x+1)) ab.

Urlöschen

Beim Urlöschen werden folgende Speicher der → CPU gelöscht:

- → Arbeitsspeicher
- Schreib-/Lesebereich des → Ladespeichers
- → Systemspeicher
- → Backup-Speicher

V.24

Serielle Schnittstelle für die Dateneingabe/-ausgabe. Über diese Schnittstelle können Bearbeitungsprogramme sowie Hersteller- und Anwenderdaten geladen und gesichert werden.

Variablendefinition

Eine Variablendefinition umfasst die Festlegung eines Datentyps und eines Variablennamens. Mit dem Variablennamen kann der Wert der Variablen angesprochen werden.

Verfahrbereich

Der maximal zulässige Verfahrbereich bei Linearachsen beträgt ± 9 Dekaden. Der absolute Wert ist abhängig von der gewählten Eingabe- und Lageregelfeinheit und dem Einheitensystem (inch oder metrisch).

Vorkoinzidenz

Satzwechsel bereits, wenn Bahnweg um ein vorgegebenes Delta der Endposition nahe gekommen ist.

Vorschub-Override

Der programmierten Geschwindigkeit wird die aktuelle Geschwindigkeitseinstellung über → Maschinensteuertafel oder von der → PLC überlagert (0-200%). Die Vorschubgeschwindigkeit kann zusätzlich im Bearbeitungsprogramm durch einen programmierbaren Prozentfaktor (1-200%) korrigiert werden.

Vorsteuerung, dynamisch

Ungenauigkeiten der → Kontur, bedingt durch Schleppfehler, lassen sich durch die dynamische, beschleunigungsabhängige Vorsteuerung nahezu eliminieren. Dadurch ergibt sich auch bei hohen → Bahngeschwindigkeiten eine hervorragende Bearbeitungsgenauigkeit. Die Vorsteuerung kann achsspezifisch über das → Teileprogramm an- und abgewählt werden.

Werkstück

Von der Werkzeugmaschine zu erstellendes/zu bearbeitendes Teil.

Werkstückkontur

Sollkontur des zu erstellenden/bearbeitenden → Werkstücks.

Werkstückkoordinatensystem

Das Werkstückkoordinatensystem hat seinen Ausgangspunkt im → Werkstücknullpunkt. Bei Programmierung im Werkstückkoordinatensystem beziehen sich Maße und Richtungen auf dieses System.

Werkstücknullpunkt

Der Werkstücknullpunkt bildet den Ausgangspunkt für das → Werkstückkoordinatensystem. Er ist durch Abstände zum → Maschinennullpunkt definiert.

Werkzeug

An der Werkzeugmaschine wirksames Teil, das die Bearbeitung bewirkt (z. B. Drehmeißel, Fräser, Bohrer, LASER-Strahl ...).

Werkzeugkorrektur

Berücksichtigung der Werkzeug-Abmessungen bei der Berechnung der Bahn.

Werkzeugradiuskorrektur

Um eine gewünschte → Werkstückkontur direkt programmieren zu können, muss die Steuerung unter Berücksichtigung des Radius des eingesetzten Werkzeugs eine äquidistante Bahn zur programmierten Kontur verfahren (G41/G42).

WinSCP

WinSCP ist ein frei verfügbares Open Source-Programm für Windows zum Transferieren von Dateien.

Zeitreziproker Vorschub

Anstelle der Vorschubgeschwindigkeit kann für die Achsbewegung auch die Zeit programmiert werden, die der Bahnweg eines Satzes benötigen soll (G93).

Zoll-Maßsystem

Maßsystem, das Entfernungen in "inch" und Bruchteilen davon definiert.

Zwischensätze

Verfahrbewegungen mit angewählter → Werkzeugkorrektur (G41/G42) dürfen durch eine begrenzte Anzahl Zwischensätze (Sätze ohne Achsbewegungen in der Korrektorebene) unterbrochen werden, wobei die Werkzeugkorrektur noch korrekt verrechnet werden kann. Die zulässige Anzahl Zwischensätze, die die Steuerung vorausliest, ist über Systemparameter einstellbar.

Zyklen

Geschützte Unterprogramme zur Ausführung von wiederholt auftretenden Bearbeitungsvorgängen am → Werkstück.

Index

\$

\$AA_BRAKE_CONDB, 546
\$AA_BRAKE_CONDM, 546
\$AA_BRAKE_STATE, 546
\$AA_COLLPOS, 302
\$AA_COUP_ACT, 416
\$AA_COUP_CORR, 554
\$AA_COUP_CORR_DIST, 556
\$AA_DTBREB, 303
\$AA_DTBREB_CMD, 303
\$AA_DTBREB_CORR, 303
\$AA_DTBREB_DEP, 303
\$AA_DTBREM, 304
\$AA_DTBREM_CMD, 304
\$AA_DTBREM_CORR, 304
\$AA_DTBREM_DEP, 304
\$AA_ESR_ENABLE, 599
\$AA_IN_SYNC, 451
\$AA_JERK_COUNT, 962
\$AA_JERK_TIME, 962
\$AA_JERK_TOT, 962
\$AA_LEAD_P, 451
\$AA_LEAD_P_TURN, 451
\$AA_LEAD_SP, 451
\$AA_LEAD_SV, 451
\$AA_LEAD_V, 451
\$AA_MASL_STAT, 768
\$AA_SYNC, 451, 520
\$AA_SYNCDIFF, 519
\$AA_TRAVEL_COUNT, 962
\$AA_TRAVEL_COUNT_HS, 962
\$AA_TRAVEL_DIST, 962
\$AA_TRAVEL_DIST_HS, 962
\$AA_TRAVEL_TIME, 962
\$AA_TRAVEL_TIME_HS, 962
\$AC_COLLPOS, 302
\$AC_TOOL_O_ACT, 112
\$AC_TOOL_O_DIFF, 112
\$AC_TOOL_O_END, 112
\$AC_TOOL_R_ACT, 113
\$AC_TOOL_R_DIFF, 113
\$AC_TOOL_R_END, 113
\$AC_TOOLO_ACT, 111
\$AC_TOOLO_DIFF, 111
\$AC_TOOLO_END, 111
\$AC_TOOLR_ACT, 111
\$AC_TOOLR_DIFF, 112
\$AC_TOOLR_END, 111
\$AC_TRAFO_CORR_ELEM_P, 393
\$AC_TRAFO_CORR_ELEM_T, 393
\$AC_TRAFO_ORIAX_LOC, 393
\$AN_ACTIVATE_COLL_CHECK, 302
\$AN_COLL_CHECK_OFF, 302
\$AN_COLL_IPO_ACTIVE, 302
\$AN_COLL_IPO_LIMIT, 302
\$AN_COLL_LOAD, 302
\$AN_COLL_MEM_AVAILABLE, 303
\$AN_COLL_MEM_USE_ACT, 303
\$AN_COLL_MEM_USE_MAX, 303
\$AN_COLL_MEM_USE_MIN, 303
\$AN_COLL_STATE, 302
\$AN_COLL_STATE_COND, 302
\$AN_ESR_TRIGGER, 599
\$NK_A_OFF, 214, 218, 220
\$NK_AXIS, 214, 217
\$NK_NAME, 209
\$NK_NEXT, 210
\$NK_OFF_DIR, 212, 216, 219, 221
\$NK_PARALLEL, 210
\$NK_SWITCH, 225
\$NK_SWITCH_INDEX, 222
\$NK_SWITCH_POS, 223
\$NK_TYPE, 212
\$NP_1ST_PROT, 249
\$NP_BIT_NO, 253
\$NP_CHAIN_ELEM, 247
\$NP_COLL_PAIR, 278, 299
\$NP_COLOR, 262
\$NP_D_LEVEL, 263
\$NP_DIR, 276
\$NP_FILENAME, 271
\$NP_INDEX, 256
\$NP_INIT_STAT, 253, 255
\$NP_NAME, 258
\$NP_NEXT, 259
\$NP_NEXTP, 261
\$NP_OFF, 275
\$NP_PARA, 274
\$NP_PROT_COLOR, 251
\$NP_PROT_D_LEVEL, 252
\$NP_PROT_NAME, 246
\$NP_PROT_TYPE, 248
\$NP_SAFETY_DIST, 300
\$NP_TYPE, 269
\$NP_USAGE, 264
\$NT_AUX_POS, 368

\$NT_BASE_ORIENT, 376
 \$NT_BASE_ORIENT_NORMAL, 377
 \$NT_BASE_TOOL_COMP, 374
 \$NT_CLOSE_CHAIN_P, 364
 \$NT_CLOSE_CHAIN_T, 365, 393
 \$NT_CNTRL, 372, 393
 \$NT_CORR_ELEM_P, 384, 392
 \$NT_CORR_ELEM_T, 383, 392
 \$NT_GEO_AX_NAME, 360
 \$NT_IDENT, 369
 \$NT_IGNORE_TOOL_ORIENT, 382
 \$NT_NAME, 351, 387
 \$NT_P_CHAIN_FIRST_ELEM, 356
 \$NT_P_CHAIN_LAST_ELEM, 358
 \$NT_POLE_LIMIT, 380
 \$NT_POLE_SIDE_FIX, 385
 \$NT_POLE_TOL, 381
 \$NT_ROT_AX_CNT, 373
 \$NT_ROT_AX_NAME, 362
 \$NT_ROT_AX_OFFSET, 363
 \$NT_ROT_AX_POS, 378
 \$NT_ROT_OFFSET_FROM_FRAME, 366
 \$NT_T_CHAIN_FIRST_ELEM, 355
 \$NT_T_CHAIN_LAST_ELEM, 357
 \$NT_T_REF_ELEM, 359
 \$NT_TRAFO_INCLUDES_TOOL, 367
 \$NT_TRAFO_INDEX, 352, 386
 \$NT_TRAFO_TYPE, 353
 \$P_PROG_EVENT, 768
 \$P_SEARCH_MASLC, 768
 \$P_SEARCH_MASLD, 768
 \$P_TOOL_O, 112
 \$P_TOOL_R, 112
 \$P_TOOLROT, 111
 \$VA_STOPSI, 606
 \$VA_SYNCDIFF, 519
 \$VC_TOOL_O, 112
 \$VC_TOOL_O_DIFF, 112
 \$VC_TOOL_R, 113
 \$VC_TOOL_R_DIFF, 113
 \$VC_TOOLO, 111
 \$VC_TOOLO_DIFF, 111
 \$VC_TOOLO_STAT, 111
 \$VC_TOOLR, 112
 \$VC_TOOLR_DIFF, 112
 \$VC_TOOLR_STAT, 112

2

2-Achsen-Schwenkkopf, 90

3

3- und 4-Achs-Transformationen, 58
 3-5-Achs-Transformation
 Aufruf und Anwendungen, 70
 3-Achs-Kinematiken, 88
 3-Achstransformationen, 73
 3- und 4-Achs-Transformation, 37

4

4-Achs-Kinematiken, 88
 4-Achs-Transformation, 37
 4-Achstransformationen, 73

5

5-Achs-Kinematik, 88
 5-Achs-Transformation
 Geometrie der Maschine, 47
 Singuläre Stellen, 56
 Werkzeugorientierung, 51

7

7-Achs-Transformation, 82
 Beispiel, 148
 Kinematik, 84

A

aaJerkCount, 962
 aaJerkTime, 962
 aaJerkTotal, 962
 aaTravelCount, 962
 aaTravelCountHS, 962
 aaTravelDist, 962
 aaTravelDistHS, 962
 aaTravelTime, 962
 aaTravelTimeHS, 962
 Abhängige Mitschleppachse, 411
 Abstandsregelung
 Inbetriebnahme, 715
 Kollisionsüberwachung, 714
 Kompensationsvektor, 710
 Programmierung, 721
 Randbedingungen, 738
 Regelkreisstruktur, 709

- Abstandsreglung
 Technologische Eigenschaften, 713
 Abstandsvektor
 maximale, 907
 ACCLIMA[FA], 562
 Achsbezeichner, 918
 Achscontainer, 903
 Achscontainer-Drehung, 903
 Achspaar-Kollisionsschutz
 Maximale Anzahl Achspaare, 893
 Achstausch, 541, 871
 Achszuordnung, 430
 Aktivierung, 71
 Aktivierung der Drehung, 128
 Aktivierung/Deaktivierung, 913
 Alarm 10653, 160
 Alarmtexte, 696
 Allgemeine Funktionalität, 911
 Allgemeines, 911
 Arbeitsebene, 851
 Archivierung, 454
 ASUP-Freigabe, 856
 Aufruf, 917
 Aufrufbedingung, 917
 Ausführliche Beschreibung, 911
 Ausführungen, 484
 Außen-/Innenecken, 926
 Auswirkungen auf die Orientierungen, 74
 AXCTSWED, 904
- B**
- Bahnrelative Orientierung, 101
 Beispiel
 Tangentialsteuerung, 770
 Beschleunigung
 funktionsspezifische, 900
 kanalspezifische, 900
 Beschleunigungsmodus, 563
 Betriebsarten
 JOG, 86
- C**
- CC_COPOFF, 833
 CC_COPON, 833
 CC_COPONM, 833
 CC_FASTOFF, 888
 CC_FASTON, 886
 CC_FASTON_CONT, 887
 CC_PREPRE, 845, 857
 CC_RESU.MPF, 859
 CC_RESU_ASUP.SPF, 863
 CC_RESU_BS_ASUP.SPF, 863
 CC_RESU_END.SPF, 862
 CC_RESU_INI.SPF, 860
 CLC_RESU_LENGTH_BS_BUFFER, 868
 COLLPAIR, 304
 Compile-Zyklen, 690
 Compile-Zyklus
 Interface-Version, 693
 Software-Version, 694
 SW-Version, 694
 CORRTRAFO, 387
 CP-BASIC, 484
 CPBC, 505
 CP-COMFORT, 484
 CPDEF, 492
 CPDEF+CPLA, 494
 CPDEL, 493
 CPDEL+CPLA, 496
 CP-EXPERT, 484
 CPFMOF, 511
 CPFMON, 510
 CPFMSON, 509
 CPFPOS, 506
 CPFRS, 503
 CPLDEF, 494
 CPLDEL, 495
 CPLDEN, 501
 CPLINSC, 517
 CPLINTR, 517
 CPLNUM, 500
 CPLOF, 499
 CPLON, 498
 CPLOUTSC, 518
 CPLOUTTR, 518
 CPLPOS, 507
 CPLSETVAL, 503
 CPMALARM, 532
 CPMBRAKE, 526
 CPMPT, 515
 CPMRESET, 513
 CPMSTART, 514
 CPMVDI, 528
 CPOF, 497
 CPOF+CPFPOS, 512
 CPON, 497
 CPRECOF, 197
 CPRECON, 197
 CPRES, 540
 CPSETTYPE, 536
 CP-STATIC, 484

- CPSYNCOPI, 521
- CPSYNCOPI2, 523
- CPSYNCOV, 521
- CPSYNFIP, 521
- CPSYNFIP2, 524
- CPSYNFIV, 522
- CTOL, 196
- CUT3DC, 940
- CUT3DCC, 951
- CUT3DCCD, 951
- CUT3DCD, 940
- CUT3DF, 945
- CUT3DFD, 945
- CUT3DFF, 945
- CUT3DFS, 945
- CYCLE750, 640
- CYCLE751
 - extern programmieren, 641
- CYCLE752
 - extern programmieren, 642
- CYCLE753
 - extern programmieren, 645
- CYCLE754
 - extern programmieren, 646
- CYCLE755
 - extern programmieren, 647
- CYCLE756
 - extern programmieren, 649
- CYCLE757
 - extern programmieren, 650
- CYCLE758
 - extern programmieren, 653
- CYCLE759
 - extern programmieren, 655

- D**
- DB10
 - DBB0, 740
 - DBB146, 740
 - DBX226.0 - DBX233.7, 253
 - DBX234.0 - DBX241.7, 253
- DB11, ...
 - DBX5.2, 166
- DB11, ...
 - DBX0.7, 856
- DB21, ...
 - DBX0.3, 415
 - DBX318.2, 135, 136
 - DBX318.3, 135
 - DBX7.3, 740
 - DBX7.4, 740
- DB21, ... DBB6.0, 580
- DB21, ...
 - DBX0.1, 847, 856, 859, 860, 863
 - DBX0.2, 848, 856, 864
 - DBX1.0, 170
 - DBX318.0, 866
 - DBX32.1, 864
 - DBX32.2, 856
 - DBX32.6, 848
 - DBX33.0, 166
 - DBX33.4, 866
 - DBX33.6, 70, 821
 - DBX7.7, 856
- DB31, ...
 - DBB101.4, 183, 184, 185
 - DBB101.6, 183, 184, 185
 - DBB101.7, 183, 184, 185
 - DBX 99.1, 463
 - DBX0.0, 415
 - DBX0.1, 415
 - DBX0.2, 415
 - DBX0.3, 415
 - DBX0.4, 415
 - DBX0.5, 415
 - DBX0.6, 415
 - DBX0.7, 415
 - DBX1.3, 415
 - DBX1.4, 169, 179, 416
 - DBX1.5, 169, 178, 416, 437, 454
 - DBX1.6, 169, 178, 416, 437, 454
 - DBX101.2, 159, 177, 178, 186
 - DBX101.3, 159, 177, 178, 186
 - DBX101.4, 166, 177, 178, 186
 - DBX101.5, 161, 166, 168, 177, 178, 180, 186
 - DBX101.6, 177, 178, 186
 - DBX101.7, 177, 178, 186
 - DBX103.0, 555
 - DBX103.4, 524
 - DBX103.5, 525
 - DBX2.1, 169, 178, 415, 416
 - DBX24.7, 772
 - DBX26.4, 463, 467
 - DBX29.4, 167, 177, 178, 183, 185
 - DBX29.5, 167, 169, 177, 178, 183
 - DBX31.6, 553
 - DBX31.7, 556
 - DBX4.0, 415
 - DBX4.1, 415
 - DBX4.2, 415
 - DBX4.3, 415, 460
 - DBX4.7/4.6, 163
 - DBX60.4, 179

DBX60.5, 179
 DBX61.5, 772
 DBX61.6, 772
 DBX61.7, 772
 DBX64.6, 179
 DBX64.7, 179
 DBX96.7, 772
 DBX97.1, 833
 DBX97.2, 833
 DBX98.0, 461, 462, 520, 551
 DBX98.1, 461, 462, 520, 551
 DBX98.4, 463
 DBX98.5, 463
 DBX98.6, 463
 DBX98.7, 593
 DBX99.0, 416
 DBX99.1, 416
 DBX99.2, 555
 DBX99.3, 463
 DBX99.4, 451, 462
 DB31, ... DBB4.3, 579
 DB31, ... DBX1.4, 416
 DB31, ... DBX96.5, 618
 DB31, ...
 DBX24.2, 833
 DBX24.3, 835
 DBX66.0, 836
 DBX97.0, 832
 DBX97.3, 834
 DB31, ... DBB20, 618
 DB31, ... DBB21, 618
 DB31, ... DBB92, 618
 DB31, ... DBB93, 618
 DB31, ... DBB94, 618
 DB31, ... DBB95, 618
 DB31, ... DBX2.1, 618
 DB31, ... DBX61.3, 618
 DB31, ... DBX96.5, 618
 DBX31, ...
 DBX1.5, 724
 DBX2.1, 724
 DBX21.7, 724
 Deaktivierung, 885
 Definition
 EG-Achsverband, 464
 Definition eines Achspaars, 896
 DELOBJ, 225
 Diagnose und Optimierung der
 Ressourcennutzung, 437
 Drehachse des Kegels, 129
 Drehrichtung, 131
 Drehung des Orientierungsvektors, 126

Drehungen des Orientierungsvektors, 125
 Drehwinkel, 131
 Dynamikgrenzen, 561

E

EG-Achsverband
 - ausschalten, 470
 - definieren, 464
 - einschalten, 465
 - löschen, 470
 EG-Achsverbände, 458
 Einrichten
 Kollisionsvermeidung, 283
 Einschränkungen, 429
 Einschränkungen für Kinematiken und
 Interpolationen, 88
 Eintauchtiefe, 925, 943
 Elektronisches Getriebe, 455, 456
 Systemvariablen, 472
 Element, 202
 Endorientierung, 129
 End-Programm, 862
 Erweiterte Interpolationen von Orientierungen, 129
 Erweiterte Überwachung
 Gantry-Achsen, 160
 ESR, 574
 Auslösequellen, 584, 585
 Verknüpfungslogik, 585
 Eulerwinkel, 51

F

Folgeachsüberlagerung, 488
 Fräserbearbeitungspunkt, 944
 Fräserhilfspunkt, 944
 Fräuserspitze, 944
 Freifahrrichtung, 896
 Freigabe, 895
 Funktionalität, 912

G

G0-Flankenwechsel, 878
 G450, 927
 G451, 927
 G91 Erweiterung
 Nullpunktverschiebung, 121
 Gantry-Achsen
 Unterschiede zum Mitschleppen, 181
 Generatorbetrieb, 574, 589

Generische 5-Achs-Transformation und Varianten, 71
Generische Orientierungstransformations-Varianten, 73
geschwenkte Linearachse
 Pol, 61
Geschwindigkeits-Schwellwert, 879
Getriebestufenwechsel bei eingeschalteter Master/Slave Kopplung, 766
Glättung
 des Orientierungsverlaufs, 99, 100
Grenzwinkel für die fünfte Achse, 57
Großkreis-Interpolation, 86
Grundorientierung, 91
GUD-Variablen, 868

H

Hardware-Voraussetzungen, 487
Hauptprogramm, 859

I

I/O-Peripherie, 877, 884
INI-Programm, 860
Interface-Versionen, 693
Interpolation, 454
Interpolation des Drehvektors, 127
Interpolation des Drehwinkels, 126
Interpolationsart und Auswahl, 122
ISD, 925
Istwertanzeige, 454

K

Kardanischer Fräskopf, 40, 66
 Anwendungen, 66
 JOG, 69
Kartesisches Handverfahren, 42
Kennzeichnung der Achsfolge, 46
Kinematik
 schwenkbare Linearachse, 39
Kinematische Ketten, 202
Kinematische Struktur, 201
Kinematische Transformation, 44
 Definition, 328
 Dynamische Orientierungstransformation, 330
 Kinematische Kette, 327
 TRACYL, 337
 TRANSMIT, 333
Kollisionspaare, 299

Kollisionstoleranz, 286
Kollisionsvermeidung
 Beispiel Grundlagen, 229, 309, 394
Kompilieren, 914
Kompressionsmodus, 96
Kompressor, 94
Konturgenauigkeit
 Programmierbare, 194
Konturtunnel
 -radius, 193
Koordinatenbezug, 503
Koordinatensystem, 411
Koppelart, 450
Koppelfaktor, 500
 Nenner, 501
 Zähler, 500
Koppelgesetz, 487, 500
Koppelmodul, 487
 Anlegen, 492
 Löschen, 493
Kopplung
 Ausschalten, 497
Kopplungsbezug, 502, 503
Kopplungstypen, 536
Kurvensegment, 420
Kurventabelle, 501
Kurventabellen, 419
 Aktivierung, 435
 Deaktivierung, 435
 Löschen, 423
 Mehrfachnutzung, 436
 Nahtstellensignale, 437
 Programmierung, 425
 Speichermangel, 422
 Speicheroptimierung, 422
 Speichertyp, 421
 Standard-Speichertyp, 424
 Transformationen, 441
 Überschreiben, 423
 Verhalten in den Betriebsarten, 437
 Werkzeugradiuskorrektur, 421

L

Laserschneiden, 704
Laufzeitoptimierung, 912
Leitachs/-spindel-Modul, 494
Leitachse
 Ausschalten, 499
 Einschalten, 498

Leitachsen	MD16505, 686
Definieren, 494	MD16506, 686
Löschen, 495	MD16507, 686
Leitwert	MD16508, 686
-kopplung, 443	MD16509, 686
-objekt, 444	MD16800, 207
Skalierung, 516	MD16900, 325
Verschiebung, 516	MD16901, 326
Leitwertkopplung	MD16902, 326
Ausschalten, 449	MD16903, 326
Einschalten, 447	MD18114, 81
Nahtstellensignale, 454	MD18170, 913
Verhalten in den Betriebsarten, 452	MD18180, 913
Linearachse, 902	MD18242, 915
	MD18351, 854, 914
	MD18400, 420, 423
	MD18402, 420, 423
	MD18403, 420, 423
	MD18404, 420, 423
	MD18406, 420, 423
	MD18408, 420, 423
	MD18409, 420, 423
	MD18410, 420, 423
	MD18450, 486
	MD18452, 487
	MD18860, 962
	MD18880, 207
	MD18882, 207, 347
	MD18890, 244, 297
	MD18892, 244, 297
	MD18893, 244
	MD18894, 245, 297
	MD18895, 244, 297
	MD18896, 297
	MD18897, 244
	MD18898, 297
	MD18899, 245
	MD19610, 895
	MD20050, 858, 885
	MD20060, 35, 918
	MD20080, 35, 918
	MD20105, 769
	MD20110, 70, 197, 413, 446, 449, 452, 453, 822
	MD20112, 197, 413, 446, 449, 452, 453
	MD20140, 822
	MD20142, 347
	MD20150, 54, 817
	MD20178, 99
	MD20210, 927
	MD20230, 927
	MD20470, 195
	MD20482, 96
M	
Maschinenkinematik, 72	
Maschinentypen, 46, 75	
5-Achs-Transformation, 72	
Maximale Achsbeschleunigung, 562	
Maximale Achsgeschwindigkeit, 562	
Maximale Schaltfrequenz, 881	
MD10000, 35, 903	
MD10002, 903	
MD10089, 605	
MD10300, 716	
MD10320, 720	
MD10350, 716	
MD10360, 885	
MD10362, 716, 719	
MD10366, 716, 719	
MD10619, 295	
MD10620, 51, 816	
MD10622, 296	
MD10640, 51, 816	
MD10674, 121	
MD10700, 911, 913, 915	
MD10712, 718	
MD11410, 412, 533	
MD11410 \$MN_SUPPRESS_ALARM_MASK	
Bit31, 461	
MD11450, 769	
MD11602, 769, 856	
MD11604, 769, 856	
MD11660 \$MN_NUM_EG, 458	
MD12701, 903	
MD12750, 903	
MD16501, 684	
MD16502, 684	
MD16503, 685	
MD16504, 686	

MD20610, 135
 MD20900, 424, 430
 MD20905, 425, 427, 429, 438
 MD21050, 193
 MD21060, 193
 MD21070, 194
 MD21080, 939
 MD21082, 940
 MD21084, 933, 940
 MD21094, 109
 MD21100, 52, 116, 126, 816
 MD21102, 116
 MD21104, 117, 817
 MD21106, 87
 MD21108, 58
 MD21120, 114
 MD21150, 116
 MD21155, 116
 MD21160, 116
 MD21165, 116
 MD21170, 116
 MD21180, 94, 149
 MD21186, 119
 MD21190, 134, 135
 MD21194, 135
 MD21198, 348
 MD21199, 348
 MD21202, 579
 MD21204, 579
 MD21380, 574, 609
 MD21380 \$MC_DELAY_TIME1, 577
 MD21380 \$MC_ESR_DELAY_TIME1, 577, 578, 579
 MD21381, 609
 MD21381 \$MC_ESR_DELAY_TIME2, 577, 579
 MD22410, 564
 MD24100, 46, 75, 76, 77, 78, 79, 780, 822, 824
 MD24110, 46, 76, 114, 780
 MD24120, 780, 825, 858
 MD24200, 75, 78
 MD24210, 76
 MD24410, 114
 MD24432, 115
 MD24460, 822
 MD24462, 115
 MD24500, 77, 78
 MD24510, 76
 MD24520, 50, 77
 MD24530, 56
 MD24540, 56, 89
 MD24550, 77, 78
 MD24558, 78
 MD24560, 77, 78
 MD24567, 82
 MD2457, 74
 MD24570, 72, 76
 MD24572, 73, 76
 MD24574, 74, 76
 MD24580, 114
 MD24582, 75
 MD24585, 114, 115
 MD24590, 119
 MD24620, 77
 MD24630, 56
 MD24640, 56, 89
 MD24670, 74
 MD24674, 76
 MD24680, 114
 MD24682, 75, 78
 MD24690, 119
 MD28010, 914
 MD28020, 914
 MD28040, 914
 MD28090, 715, 835, 852, 884
 MD28100, 715, 835, 852, 884
 MD28105, 853
 MD28291, 940
 MD28590, 99
 MD30110, 617
 MD30130, 444, 448, 450
 MD30132, 444, 741
 MD30230, 617
 MD30300, 902, 903
 MD30310, 902, 903
 MD30450, 180
 MD30455, 547, 553
 MD30552, 832
 MD32000, 561, 717
 MD32000 \$MA_MAX_AX_VELO, 463
 MD32100, 50, 172
 MD32200, 173, 717
 MD32230, 717
 MD32300, 561, 836
 \$MA_MAX_AX_ACCEL, 463
 MD32300 \$MA_MAX_AX_ACCEL, 463
 MD32400, 173
 MD32410, 173, 708
 MD32415, 196
 MD32420, 173
 MD32430, 173
 MD32434, 839
 MD32610, 173, 708
 MD32620, 173
 MD32650, 173
 MD32800, 173

MD32810, 173, 174
 MD32900, 174
 MD32910, 174
 MD33000, 173
 MD33060, 963
 MD33100, 95
 MD34040, 164
 MD34070, 164
 MD34080, 168, 175, 184
 MD34090, 168, 175, 184
 MD34100, 162, 164, 166, 168, 170, 185
 MD34110, 170
 MD34330, 164, 168, 171
 MD36000, 718
 MD36010, 718
 MD36012, 169
 MD36020, 718
 MD36030, 169
 MD36040, 719
 MD36060, 719
 MD36100, 740
 MD36110, 740
 MD36120, 740
 MD36130, 740
 MD36500, 170, 193
 MD36600, 550
 MD36954, 605
 MD36955, 605
 MD37014, 765
 MD37100, 174, 182, 741
 MD37110, 159, 164, 175, 182, 185, 186
 MD37120, 159, 172, 182, 185, 186
 MD37130, 159, 172, 182, 186
 MD37135, 160
 MD37140, 162, 167, 179, 180
 MD37150, 160, 168
 MD37160, 437, 447
 MD37200, 521
 MD37200
 \$MA_COUPLE_POS_TOL_COARSE, 461, 473
 MD37200
 \$MN_COUPLE_POS_TOL_COARSE, 461, 462
 MD37202, 523
 MD37210, 521
 MD37210 \$MA_COUPLE_POS_TOL_FINE, 462
 MD37210 \$MA_COUPLE_POS_TOL_FINE, 461, 473
 MD37212, 523
 MD37220, 521
 MD37230, 521
 MD37262, 771
 MD37268, 764
 MD37400, 624
 MD37500, 609
 MD37500 \$MA_ESR_REACTION, 578
 MD37550 \$MA_EG_VEL_WARNING, 463
 MD37560 \$MA_EG_ACC_TOL, 463
 MD38020, 195
 MD43108, 479
 MD51160, 298
 MD51161, 298
 MD51162, 298
 MD60900+i, 851
 MD60940, 715
 MD60948, 884, 886
 MD60972, 895, 904
 MD61516, 896, 903
 MD61517, 896, 903
 MD61518, 897
 MD61519, 897
 MD61532, 898
 MD61535, 895
 MD62500, 718
 MD62502, 718
 MD62504, 714, 718, 723
 MD62505, 720, 728, 729
 MD62506, 720, 728, 729
 MD62508, 720, 740
 MD62510, 727, 734
 MD62511, 720, 727, 734
 MD62512, 727, 734
 MD62513, 720, 727, 734
 MD62522, 720, 733
 MD62523, 731
 MD62524, 722
 MD62528, 717
 MD62560, 885
 MD62571, 854
 MD62572, 854
 MD62574, 855
 MD62575, 867
 MD62580, 851
 MD62600, 776, 794, 801, 807, 811, 825
 MD62601, 791, 794, 801, 808
 MD62602, 811, 812, 813, 815, 825
 MD62603, 782, 794, 801, 807, 825
 MD62604, 784, 794, 801, 808, 825
 MD62605, 788, 794, 801, 807, 825
 MD62606, 784, 801, 808, 825
 MD62607, 783, 794, 801, 808, 825
 MD62608, 784, 802, 808
 MD62609, 784, 802, 808
 MD62610, 784, 795, 807, 808
 MD62611, 784, 795, 801, 807, 808

MD62612, 782, 795, 802, 808
MD62613, 782, 795, 802, 808
MD62614, 808, 825
MD62615, 825
MD62616, 785, 808, 825
MD62617, 791, 794, 801, 808, 825
MD62618, 790, 794, 801, 808, 825
MD62620, 789, 794, 801, 807, 825
MD62629, 792
MD62630, 793
MD62631, 793
MD62632, 793
MD62636, 820
MD62637, 820
MD63514, 900
MD63540, 832
MD63541, 834
MD63542, 835
MD63543, 835
MD63544, 835
MD63545, 835
MD65520, 717
MD65530, 717
MELDW
 Bit9, 600
Minimaler Schaltpositionsabstand, 881
Mitschleppachse als Leitachse, 411
Mitschleppachsen, 410
Mitschleppen, 409
 Dynamikbegrenzung, 417
 Ein/Ausschalten, 412
 Nahtstellensignale, 415
 Programmierung, 414
 Regeldynamik, 417
 Restweg, 413
Mitschleppverband, 410
 Ausschalten, 414
 Definition und Einschalten, 414
Modulo Rundachsen, 903
Modulo-Anzeige, 120
Modulo-Leitachse, 436

N

NAMETOINT, 228
NC-geführtes Erweitertes Stillsetzen, 577
NC-Satz-Kompressor, 94

O

Öffnungswinkel, 131

Öffnungswinkel des Kegels, 130
Offset, 84
Onboard-Ausgänge, 877
Online-Werkzeuiglängenkorrektur, 133
Option, 895
ORIC, 927, 936
ORICONCCW, 130, 132
ORICONCW, 130, 132
ORICONIO, 131, 132
ORICONTO, 131
ORICURVE, 131
ORID, 927, 936
Orientierung, 86, 91
Orientierung der Maschinenachsen, 898
Orientierung des Werkzeugs, 41
Orientierung im WKS und MKS, 52
Orientierung mit Achsinterpolation, 86
Orientierungsachsen, 41, 113
 Modulo-Anzeige, 120
Orientierungsänderung, 131
Orientierungsbewegungen
 mit Achsbegrenzungen, 93
Orientierungsprogrammierung, 51
Orientierungsrichtung, 125
Orientierungsrichtung und Drehung, 126
Orientierungstransformation, 44
 Programmierung, 116
Orientierungstransformation und orientierbare
Werkzeugträger, 119
Orientierungsvektor, 122
ORIMKS, 52
ORIPANE, 130, 132
ORISOF, 100
ORISON, 99, 100
ORIWKS, 52, 83

P

p0888, 594, 596, 597
p0890, 607
p0891, 595, 596
p0892, 595, 596
p0893, 596
p1240, 597
p1248, 598
p1414, 708
p2161, 598
p9554, 606, 607
p9555, 606
p9580, 606, 607
p9697, 606, 607
Parallele Teilketten, 203

- PI-Regler, 563
 Pol, 56
 Polynom 5. Grades, 127
 Polynome für 2 Winkel, 122
 Polynominterpolation, 123
 Orientierungsvektor, 121
 POLYPATH, 122
 Programmhandling, 913
 Programmierung
 kartesische Position, 85
 Orientierung, 85
 Polynome, 121
 PROTA, 305
 PROTD, 307
 PROTS, 306
- R**
- r9721, 607
 r9723, 607
 Randbedingungen, 918
 Raumfeste Werkzeugspitze, 52
 Reaktion auf Stopp, 446
 Regeldynamik, 454
 RESET-Verhalten, 858
 RESU-Arbeitsebene, 851
 RESU-ASUP, 863
 RESU-fähiger Konturbereich, 846
 RESU-Hauptprogramm, 859
 Richtungsvektors, 130
 RPY-Winkel, 51
 Rückziehen, 573, 574
 Rundachse, 902
 Rundachsposition
 berechnen, 93
- S**
- Satzsuchlauf
 Master-Slave, Drehzahlkopplung, 768
 Satzwechselkriterium, 504
 Schaltkriterien, 878
 Schaltposition, 878
 Schaltpositionsverschiebung, 882, 883
 Schaltzeitpunkte, 881
 Schlüsselwörter, 489
 Schnellstopp, 550
 Schutzfenster, 897
 Schwenkbare Linearachse, 39
 SD41610, 348, 393
 SD41611, 348, 393
 SD42450, 196
 SD42460, 196
 SD42475, 95
 SD42476, 95
 SD42477, 95
 SD42660, 88
 SD42670, 106
 SD42672, 107
 SD42678, 100
 SD42680, 100
 SD42970, 135
 SD42974, 76
 SD43100, 450
 SD43102, 445
 SD43104, 445
 SD43106, 445
 SD43108, 445
 Serieninbetriebnahme, 855
 Sicherheitsabstand, 286
 Siemens-Compile-Zyklen, 690
 Simulierter Leitwert, 444
 Singuläre Stellen, 56
 Singularitäten, 89
 Skalierung
 eines Leitwerts, 516
 Slot, 903
 Sonderfunktionen, 690
 Speicherbedarf, 914
 Speicherkonfiguration
 Generische Kopplung, 486
 Kurventabelle, 420
 Spiegelung
 Rückzugsrichtung (Schnellabheben), 579
 Spindel
 in Leitwertkopplung, 447
 Sprachumfang, 918
 Startorientierung, 129
 Startwert, 430
 Status der Kopplung, 416, 454
 Stillsetzen, 573
 Erweitertes, 574
 Stirnfräsen, 930
 Stirnseitentransformation, 333
 SW-Version, 694
 Synchronisation, 563
 Folgeachse, 456
 Synchronisationsmodus, 508
 Synchronlauf
 -abweichung, 551
 -differenz, 519
 -korrektur, 555
 -überwachung, 519

Synchronlaufdifferenz, 461
- abfragen, 463
Synchronlaufüberwachung
 Stufe 2, 523
Synchronposition
 Folgeachse, 506
 Leitachse, 507
Syntax-Check, 918
Systemvariable
 für Orientierung, 36
Systemvariablen, 491

T

TANG, 625
TANGDEL, 630
Tangentialsteuerung, 623
 Beispiele, 770
TANGOF, 630
TANGON, 629
Technologie-Funktion
 Aktivieren, 695
Technologie-Funktionen, 689
Temporäre Kurventabelle, 422
TLIFT, 627
TRAANG, 441
TRACYL, 337
TRAFOON, 386
TRAILOF, 414
TRAILON, 414
Transformation, 872
Transformation aktiv, 70
Transformation inaktiv, 70
TRANSMIT, 333

U

Überwachung
 Synchronlauf-, 519
Unabhängige Mitschleppachse, 411
Unterbrechungspunkt, 846

V

V2 Vorverarbeitung
 Kurzbeschreibung, 911
VELOLIMA[FA], 562
Verhalten am Pol, 56
Verhalten an Innenecken, 937
Verschiebung
 eines Leitwerts, 516

Verschiebungsvektor, 896
Virtuelle Leitachse, 444

W

WAITC, 505
Weltkoordinatensystem, 204
Werkstück-Kette, 391
Werkzeug-Kette, 391
Werkzeugkorrekturen, 873
Werkzeugorientierung mittels
 Orientierungsvektoren, 55, 89
Werkzeugträger, orientierbare
 Programmierung, 89
Wiederaufsetz-ASUP, 863
Wiederaufsetzpunkt, 846

Z

Zugriffsrecht, 914
Zustände
 Schutzbereiche, 284
Zwischenkreis
 Energiebilanz, 588
Zwischenorientierung, 131
Zylindermanteltransformation, 337