

SIEMENS

SINUMERIK

SINUMERIK ONE Tools

Function Manual

Introduction

1

Fundamental safety
instructions

2

Tool offset

3

3D tool radius
compensation

4

Grinding-specific tool
offset and tool monitoring

5

Appendix

A

Valid for

controller
SINUMERIK ONE

CNC software Version 6.23

01/2024

A5E48053685B AH

Legal information

Warning notice system

This manual contains notices you have to observe in order to ensure your personal safety, as well as to prevent damage to property. The notices referring to your personal safety are highlighted in the manual by a safety alert symbol, notices referring only to property damage have no safety alert symbol. These notices shown below are graded according to the degree of danger.

DANGER

indicates that death or severe personal injury will result if proper precautions are not taken.
--

WARNING
--

indicates that death or severe personal injury may result if proper precautions are not taken.

CAUTION
--

indicates that minor personal injury can result if proper precautions are not taken.
--

NOTICE

indicates that property damage can result if proper precautions are not taken.
--

If more than one degree of danger is present, the warning notice representing the highest degree of danger will be used. A notice warning of injury to persons with a safety alert symbol may also include a warning relating to property damage.

Qualified Personnel

The product/system described in this documentation may be operated only by **personnel qualified** for the specific task in accordance with the relevant documentation, in particular its warning notices and safety instructions. Qualified personnel are those who, based on their training and experience, are capable of identifying risks and avoiding potential hazards when working with these products/systems.

Proper use of Siemens products

Note the following:

WARNING
--

Siemens products may only be used for the applications described in the catalog and in the relevant technical documentation. If products and components from other manufacturers are used, these must be recommended or approved by Siemens. Proper transport, storage, installation, assembly, commissioning, operation and maintenance are required to ensure that the products operate safely and without any problems. The permissible ambient conditions must be complied with. The information in the relevant documentation must be observed.
--

Trademarks

All names identified by ® are registered trademarks of Siemens Aktiengesellschaft. The remaining trademarks in this publication may be trademarks whose use by third parties for their own purposes could violate the rights of the owner.

Disclaimer of Liability

We have reviewed the contents of this publication to ensure consistency with the hardware and software described. Since variance cannot be precluded entirely, we cannot guarantee full consistency. However, the information in this publication is reviewed regularly and any necessary corrections are included in subsequent editions.

Table of contents

1	Introduction	9
1.1	About SINUMERIK	9
1.2	About this documentation	9
1.3	Documentation on the internet	12
1.3.1	Documentation overview SINUMERIK ONE	12
1.3.2	Documentation overview SINUMERIK operator components	12
1.4	Feedback on the technical documentation	13
1.5	mySupport documentation	13
1.6	Service and Support.....	14
1.7	Using OpenSSL	15
1.8	Compliance with the General Data Protection Regulation	15
2	Fundamental safety instructions.....	17
2.1	General safety instructions.....	17
2.2	Warranty and liability for application examples	17
2.3	Cybersecurity information	17
3	Tool offset	19
3.1	Introduction.....	19
3.1.1	Programmed contour and tool path.....	19
3.1.2	Tool length compensation.....	19
3.1.3	Tool radius compensation	20
3.2	Tool change and activation of tool offset	21
3.2.1	Selection of tool and cutting edge and tool change	21
3.2.2	Activation of the tool offset	24
3.3	Tool offset memory	24
3.3.1	Compensation memory structure	24
3.3.2	Calculating the tool compensation	26
3.3.3	Address extension for NC addresses T and M.....	26
3.3.4	Free assignment of D numbers	28
3.3.5	Compensation block in case of error during tool change	33
3.3.6	Definition of the effect of the tool parameters.....	36
3.4	Tool parameters	37
3.4.1	Overview	37
3.4.2	Tool parameter 1: Tool type.....	40
3.4.3	Tool parameter 2: Cutting edge position.....	44
3.4.4	Tool parameters 3 ... 5: Geometry - tool lengths	45
3.4.5	Tool parameters 6 ... 11: Geometry - tool shape.....	46
3.4.6	Tool parameters 12 ... 14: Wear - tool lengths.....	49
3.4.7	Tool parameters 15 ... 20: Wear - tool shape	50

3.4.8	Tool parameters 21 ... 23: Base dimension / adapter dimension.....	50
3.4.9	Tool parameter 24: Clearance angle	53
3.5	2D tool radius compensation.....	54
3.5.1	Function	54
3.5.1.1	Calculating the equidistant tool path	54
3.5.1.2	Response when approaching and/or retracting	58
3.5.1.3	Behavior when traveling around outside corners.	62
3.5.1.4	Behavior at inside corners	66
3.5.1.5	Blocks with variable compensation value	68
3.5.1.6	Alarm behavior	70
3.5.2	Parameterization.....	70
3.5.3	Programming.....	74
3.5.3.1	Activating/deactivating tool radius compensation (G40, G41, G42, OFFN):	74
3.5.3.2	Adapting the approach/retract response (NORM, KONT, KONTC, KONTT)	81
3.5.3.3	Defining the response when traveling around outside corners (G450, G451, DISC).....	85
3.6	Collision monitoring ("bottleneck detection").....	88
3.6.1	Function	88
3.6.2	Parameterization.....	89
3.6.3	Programming.....	89
3.6.4	Supplementary conditions	90
3.6.5	Example	91
3.7	Keep tool radius compensation constant	91
3.8	Smooth approach and retraction	94
3.8.1	Function	94
3.8.2	Parameters	95
3.8.3	Velocities	102
3.8.4	System variables	103
3.8.5	Supplementary conditions	104
3.8.6	Examples	105
3.9	Extension of the approach and retract strategy	108
3.10	2 1/2 D tool radius compensation	111
3.11	Slot shape recognition (option)	116
3.12	Tool carriers with orientation capability	119
3.12.1	General information.....	119
3.12.2	Create new tool carrier.....	123
3.12.3	Configuring tool carrier kinematics via machine data	124
3.12.3.1	Kinematic interaction and machine design	124
3.12.3.2	Parameter sets for tool carriers	131
3.12.3.3	Tool carrier data set.....	132
3.12.4	Configuring tool carrier kinematics with kinematic chain	135
3.12.4.1	Tool carrier with kinematic chains	135
3.12.5	Programming.....	143
3.12.5.1	Tool length compensation for orientable toolholders (TCARR, TCOABS, TCOFR, TCOFRX, TCOFRY, TCOFRZ)	143
3.12.5.2	Modifying the orientable tool carrier according to the machine measurement (CORRTC) ...	147
3.12.6	Examples	151
3.12.6.1	Example: Toolholder with orientation capability	151
3.12.6.2	Example of toolholder with orientation capability with rotary table	152

3.12.6.3	Calculation of compensation values on a location-specific and workpiece-specific basis	154
3.12.6.4	Example: Tool carrier with orientation capability via kinematic chain.....	157
3.12.7	Inclined surface machining with 3 + 2 axes	159
3.12.8	Machine with rotary work table.....	160
3.12.9	Procedure when using toolholders with orientation capability	163
3.12.10	Supplementary conditions and control system response for orientation	167
3.13	Online tool length compensation	170
3.13.1	Function	170
3.13.2	Parameterization.....	172
3.13.3	Programming.....	175
3.13.3.1	Online tool length compensation (TOFFON, TOFFOF)	175
3.13.4	Constraints	181
3.14	Modification of the offset data for rotatable tools.....	182
3.14.1	Introduction.....	182
3.14.2	Rotating turning tools	183
3.14.2.1	Cutting edge position, cut direction, and angle for turning tools	183
3.14.2.2	Modifications during the rotation of turning tools	186
3.14.3	Rotation of milling and drilling tools	194
3.14.3.1	Cutting edge position for milling and tapping tools.....	194
3.14.3.2	Modifications during rotation of milling and tapping tools	195
3.14.4	Commissioning	195
3.14.4.1	Parameter assignment	195
3.14.5	Programming.....	198
3.14.5.1	Calculating orientations (ORISOLH)	198
3.14.5.2	Activating the modification of the offset data for rotatable tools (CUTMOD, CUTMODK).....	206
3.14.6	Example	211
3.15	Incrementally programmed compensation values	213
3.15.1	G91 extension	213
3.15.2	Traversing in the direction of tool orientation (MOVT)	214
3.16	Assignment of tool length components to geometry axes	216
3.16.1	Assignment according to tool type and working plane.	216
3.16.2	Assignment when changing plane.....	216
3.16.3	Assignment independent of tool type.....	217
3.17	Paraxial tool orientation	218
3.17.1	Basic tool orientation	218
3.17.2	Tool orientation for plane change.....	218
3.18	Parameterizable basic tool orientation	220
3.18.1	Function	220
3.18.2	Commissioning	220
3.18.2.1	Activation	220
3.18.2.2	Parameterization.....	221
3.18.3	Programming.....	222
3.18.4	Examples	225
3.19	Special handling of tool offsets.....	225
3.19.1	Relevant setting data	225
3.19.2	Mirroring tool lengths	226
3.19.3	Mirroring wear lengths.....	227
3.19.4	Tool lengths in the WCS, allowing for the orientation.....	229
3.19.5	Tool length offsets in tool direction.....	229

3.19.6	Special characteristics of orientable tool carriers	234
3.20	Sum and setup offsets	234
3.20.1	General information.....	234
3.20.2	Description of function	235
3.20.3	Activation	238
3.20.4	Examples	244
3.20.5	Extensions for tool length determination	245
3.20.5.1	Calculation of compensation values on a location-specific and workpiece-specific basis	245
3.20.5.2	Functionality of the individual wear values	249
3.21	Working with tool environments	252
3.21.1	Save tool environment (TOOLENV)	253
3.21.2	Delete tool environment (DELTOOLENV)	255
3.21.3	Read T, D and DL number (GETTENV)	256
3.21.4	Read information about the saved tool environments (\$P_TOOLENVN, (\$P_TOOLENV)	257
3.21.5	Read tool lengths and/or tool length components (GETTCOR)	257
3.21.6	Change tool components (SETTCOR)	263
3.22	Read the assignment of the tool lengths L1, L2, L3 to the coordinate axes (LENTOAX)	275
3.23	Boundary conditions	278
3.23.1	Tools with a relevant tool point direction	278
3.23.2	SD42935 expansions	279
3.23.3	Scratching	279
4	3D tool radius compensation	281
4.1	Function	281
4.1.1	Introduction.....	281
4.1.2	Circumferential milling.....	282
4.1.2.1	Corners in circumferential milling.....	284
4.1.2.2	Behavior at outer corners	285
4.1.2.3	Behavior at inside corners	286
4.1.2.4	Monitoring of path curvature	288
4.1.3	Face milling	288
4.1.3.1	Tool shapes and tool data for face milling	289
4.1.3.2	Face milling with a specified surface normal vector.....	290
4.1.3.3	Compensation on path.....	292
4.1.3.4	Corners in face milling	293
4.1.3.5	Behavior at outer corners	294
4.1.3.6	Behavior at inside corners	294
4.1.3.7	Monitoring of path curvature	296
4.2	Commissioning	296
4.2.1	Parameter assignment	296
4.3	Programming.....	297
4.3.1	Selecting 3D tool radius compensation for 3D circumferential milling (CUT3DC, CUT3DCD, ISD).....	297
4.3.2	Selecting 3D tool radius compensation for the 3D face milling (CUT3DF, CUT3DFS, CUT3DFF, CUT3DFD)	302
4.3.3	3D circumferential milling taking into account a limitation surface (CUT3DCC, CUT3DCCD)	308
4.4	Boundary conditions	312

4.5	Examples	313
4.5.1	Example 1: Orientation change at outside corner during 3D circumferential milling	313
4.5.2	Example 2: Orientation change at inside corner during 3D circumferential milling	315
5	Grinding-specific tool offset and tool monitoring	317
5.1	Grinding-specific tool data	317
5.1.1	Structure of tool data	317
5.1.2	Cutting edge-specific parameters	318
5.1.2.1	List of cutting edge-specific parameters	318
5.1.2.2	\$TC_DP1	320
5.1.2.3	Definition of additional parameters \$TC_DPC1 ... 10	320
5.1.3	Tool-specific parameters	321
5.1.3.1	List of tool-specific parameters	321
5.1.3.2	\$TC_TPG1	322
5.1.3.3	\$TC_TPG2	322
5.1.3.4	\$TC_TPG3, \$TC_TPG4	323
5.1.3.5	\$TC_TPG5	323
5.1.3.6	\$TC_TPG6 and \$TC_TPG7	324
5.1.3.7	\$TC_TPG8	324
5.1.3.8	\$TC_TPG9	325
5.1.3.9	\$TC_TPG_DRSPATH and \$TC_TPG_DRSPROG	325
5.1.3.10	Definition of additional parameters \$TC_TPC1 ... 10	326
5.1.3.11	Access to tool-specific parameters	326
5.1.4	Planes and axis assignments	326
5.1.5	Examples	328
5.2	Online tool offset	330
5.2.1	Function	330
5.2.2	Commissioning	332
5.2.3	Programming	333
5.2.3.1	Defining a polynomial function (FCTDEF)	333
5.2.3.2	Write online tool offset continuously (PUTFTOCF)	334
5.2.3.3	Write online tool offset, discrete (PUTFTOC)	335
5.2.3.4	Activate/deactivate online tool offset (FTOCON/FTOCOF)	336
5.2.4	Supplementary conditions	336
5.2.5	Examples	338
5.2.5.1	Example: Write online tool offset continuously	338
5.3	Online tool radius compensation	340
5.4	Grinding-specific tool monitoring	341
5.4.1	Function	341
5.4.2	Commissioning	343
5.4.3	Programming	343
5.4.3.1	Activate/deactivate grinding-specific tool monitoring (TMON, TMOF)	343
5.5	Constant grinding wheel peripheral speed (GWPS)	344
5.5.1	Function	344
5.5.2	Commissioning	345
5.5.3	Programming	346
5.5.3.1	Switching constant grinding wheel peripheral speed (GWPSON, GWPSOF) on/off:	346
5.5.4	Example	347

A **Appendix..... 349**

 A.1 List of abbreviations..... 349

Index 355

Introduction

1.1 About SINUMERIK

From simple, standardized CNC machines to premium modular machine designs – the SINUMERIK CNCs offer the right solution for all machine concepts. Whether for individual parts or mass production, simple or complex workpieces – SINUMERIK is the highly dynamic automation solution, integrated for all areas of production. From prototype construction and tool design to mold making, all the way to large-scale series production.

Visit our website for more information SINUMERIK (<https://www.siemens.com/sinumerik>).

1.2 About this documentation

This documentation is part of the documentation category of SINUMERIK Function Manuals.

SINUMERIK Function Manuals

The SINUMERIK Function Manuals describe the NC functions of a SINUMERIK control.

The target group are configuring engineers, technologists, commissioning engineers and programmers.

Each Function Manual covers a specific topic area and contains all the function descriptions for this topic area.

The following table shows which Function Manuals are available for your SINUMERIK control and which topic areas a Function Manual covers:

Function Manual	Topic area
Basic functions	Basic functions of a CNC control
Axes and spindles	Axis and spindle functions, axis couplings
Transformations	Transformation functions
Monitoring and compensating	Axis monitoring and compensation functions, collision avoidance
Tools	Functions for selection, correction and monitoring of tools
Synchronized actions	Functionality of synchronized actions
PLC	Structure and functions of the PLC
Tool management	Function, commissioning and programming of tool management
Technologies	Extended technological functionalities

Overview of contents

You will find an overview of the main chapters of the descriptions of functions that are contained in a Function Manual on the title page.

Validity

The title page also contains all data about the validity of a document, i.e. for which SINUMERIK control and for which software version this edition of the Function Manual is valid.

System data

In the descriptions of functions, the system data relevant for a function (machine data, setting data, system variables, interface signals and alarms) are only described with the level of detail that is strictly necessary to understand the function. You will find detailed information on this data in the relevant List Manuals and for alarms in the Diagnostics Manual.

Basic Program Plus and Basic Program

In the TIA Portal, two different types of basic program are available for programming the PLC functionality:

- Basic Program Plus works only with symbolic addressing. The functionality is implemented via data types, PLC blocks, functions and instructions. Basic Program Plus is available only for SINUMERIK ONE.
- Basic Program works with both symbolic and absolute addressing. The programs are thus largely compatible with already existing programs from the SIMATIC Manager. Basic Program is available for ONE and MC controllers.

Basic Program Plus and Basic Program cannot be used together.

PLC blocks and data types

The descriptions of functions often refer to the PLC signals and functions. In the text, they are primarily the blocks and data types of Basic Program Plus, which is only available for SINUMERIK ONE.

You can find detailed information in the PLC Function Manual for the relevant controller.

Interface signals

The symbolic addressing of Basic Program Plus is the first choice for addressing interface signals. To derive signals for Basic Program, there are tables at the end of the relevant section with the PLC signals used for the different signal directions, as shown in the following example:

NC → PLC

Basic Program Plus	Basic Program	
<Axis>.basic.in.enc1Synchronized	LBP_Axis*.E_RefSyn1	DB31, ...DBX60.4
...		

PLC → NC

Basic Program Plus	Basic Program	
<Axis>.basic.out.refPointApproachCam	LBP_Axis*.A_DelayRef	DB31, ...DBX12.7
...		

Note**Quantity structure**

In Basic Program Plus, axes, channels etc. are represented via instances of the corresponding objects.

The absolute addressing in Basic Program is based on the absolute maximum number of the following components: Mode groups (DB11), channels (DB21, ...), axes/spindles (DB31, ...)

Standard scope

This documentation only describes the functionality of the standard version. This may differ from the scope of the functionality of the system that is actually supplied. Please refer to the ordering documentation only for the functionality of the supplied drive system.

It may be possible to execute other functions in the system which are not described in this documentation. This does not, however, represent an obligation to supply such functions with a new control or when servicing.

For reasons of clarity, this documentation cannot include all of the detailed information on all product types. Further, this documentation cannot take into consideration every conceivable type of installation, operation and service/maintenance.

The machine manufacturer must document any additions or modifications they make to the product themselves.

Websites of third-party companies

This document may contain hyperlinks to third-party websites. Siemens is not responsible for and shall not be liable for these websites and their content. Siemens has no control over the information which appears on these websites and is not responsible for the content and information provided there. The user bears the risk for their use.

1.3 Documentation on the internet

1.3.1 Documentation overview SINUMERIK ONE

Comprehensive documentation about the functions provided in SINUMERIK ONE Version 6.13 and higher is provided in the Documentation overview SINUMERIK ONE (<https://support.industry.siemens.com/cs/ww/en/view/109768483>).



You can display documents or download them in PDF and HTML5 format.

The documentation is divided into the following categories:

- User: Operating
- User: Programming
- Manufacturer/Service: Functions
- Manufacturer/Service: Hardware
- Manufacturer/Service: Configuration/Setup
- Manufacturer/Service: Safety Integrated
- Information and training
- Manufacturer/Service: SINAMICS

1.3.2 Documentation overview SINUMERIK operator components

Comprehensive documentation about the SINUMERIK operator components is provided in the Documentation overview SINUMERIK operator components (<https://support.industry.siemens.com/cs/document/109783841/technische-dokumentation-zu-sinumerik-bedienkomponenten?dti=0&lc=en-WW>).

You can display documents or download them in PDF and HTML5 format.

The documentation is divided into the following categories:

- Operator Panels
- Machine control panels
- Machine Pushbutton Panel
- Handheld Unit/Mini handheld devices
- Further operator components

An overview of the most important documents, entries and links to SINUMERIK is provided at SINUMERIK Overview - Topic

Page (<https://support.industry.siemens.com/cs/document/109766201/sinumerik-an-overview-of-the-most-important-documents-and-links?dti=0&lc=en-WW>).

1.4 Feedback on the technical documentation

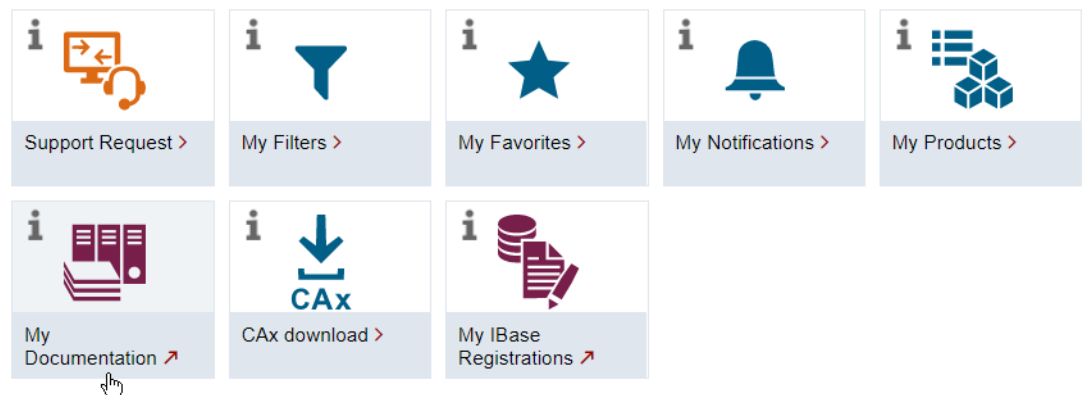
If you have any questions, suggestions or corrections regarding the technical documentation which is published in the Siemens Industry Online Support, use the link "Send feedback" link which appears at the end of the entry.

1.5 mySupport documentation

With the "mySupport documentation" web-based system you can compile your own individual documentation based on Siemens content, and adapt it for your own machine documentation.

To start the application, click on the "My Documentation" tile on the "mySupport links and tools" (<https://support.industry.siemens.com/cs/ww/en/my>) portal page:

mySupport Links and Tools



The configured manual can be exported in RTF, PDF or XML format.

Note

Siemens content that supports the mySupport documentation application can be identified by the presence of the "Configure" link.

1.6 Service and Support

Product support

You can find more information about products on the internet:

Product support (<https://support.industry.siemens.com/cs/ww/en/>)

The following is provided at this address:

- Up-to-date product information (product announcements)
- FAQs (frequently asked questions)
- Manuals
- Downloads
- Newsletters with the latest information about your products
- Global forum for information and best practice sharing between users and specialists
- Local contact persons via our Contacts at Siemens database (→ "Contact")
- Information about field services, repairs, spare parts, and much more (→ "Field Service")

Technical support

Country-specific telephone numbers for technical support are provided on the internet at address (<https://support.industry.siemens.com/cs/ww/en/sc/4868>) in the "Contact" area.

If you have any technical questions, please use the online form in the "Support Request" area.

Training

You can find information on SITRAIN at the following address (<https://www.siemens.com/sitrain>).

SITRAIN offers training courses for automation and drives products, systems and solutions from Siemens.

Siemens support on the go



With the award-winning "Industry Online Support" app, you can access more than 300,000 documents for Siemens Industry products – any time and from anywhere. The app can support you in areas including:

- Resolving problems when implementing a project
- Troubleshooting when faults develop
- Expanding a system or planning a new system

Furthermore, you have access to the Technical Forum and other articles from our experts:

- FAQs
- Application examples
- Manuals
- Certificates
- Product announcements and much more

The "Industry Online Support" app is available for Apple iOS and Android.

1.7 Using OpenSSL

This product can contain the following software:

- Software developed by the OpenSSL project for use in the OpenSSL toolkit
- Cryptographic software created by Eric Young.
- Software developed by Eric Young

You can find more information on the internet:

- OpenSSL (<https://www.openssl.org>)
- Cryptsoft (<https://www.cryptsoft.com>)

1.8 Compliance with the General Data Protection Regulation

Siemens observes standard data protection principles, in particular the data minimization rules (privacy by design).

1.8 Compliance with the General Data Protection Regulation

For this product, this means:

The product does not process or store any personal data, only technical function data (e.g. time stamps). If the user links this data with other data (e.g. shift plans) or if he/she stores person-related data on the same data medium (e.g. hard disk), thus personalizing this data, he/she must ensure compliance with the applicable data protection stipulations.

Fundamental safety instructions

2.1 General safety instructions

 WARNING
Danger to life if the safety instructions and residual risks are not observed If the safety instructions and residual risks in the associated hardware documentation are not observed, accidents involving severe injuries or death can occur. • Observe the safety instructions given in the hardware documentation. • Consider the residual risks for the risk evaluation.

 WARNING
Malfunctions of the machine as a result of incorrect or changed parameter settings As a result of incorrect or changed parameterization, machines can malfunction, which in turn can lead to injuries or death. • Protect the parameterization against unauthorized access. • Handle possible malfunctions by taking suitable measures, e.g. emergency stop or emergency off.

2.2 Warranty and liability for application examples

Application examples are not binding and do not claim to be complete regarding configuration, equipment or any eventuality which may arise. Application examples do not represent specific customer solutions, but are only intended to provide support for typical tasks.

As the user you yourself are responsible for ensuring that the products described are operated correctly. Application examples do not relieve you of your responsibility for safe handling when using, installing, operating and maintaining the equipment.

2.3 Cybersecurity information

Siemens provides products and solutions with industrial cybersecurity functions that support the secure operation of plants, systems, machines and networks.

In order to protect plants, systems, machines and networks against cyber threats, it is necessary to implement – and continuously maintain – a holistic, state-of-the-art industrial cybersecurity concept. Siemens' products and solutions constitute one element of such a concept.

2.3 Cybersecurity information

Customers are responsible for preventing unauthorized access to their plants, systems, machines and networks. Such systems, machines and components should only be connected to an enterprise network or the internet if and to the extent such a connection is necessary and only when appropriate security measures (e.g. firewalls and/or network segmentation) are in place.

For additional information on industrial cybersecurity measures that may be implemented, please visit
<https://www.siemens.com/cybersecurity-industry>.

Siemens' products and solutions undergo continuous development to make them more secure. Siemens strongly recommends that product updates are applied as soon as they are available and that the latest product versions are used. Use of product versions that are no longer supported, and failure to apply the latest updates may increase customer's exposure to cyber threats.

To stay informed about product updates, subscribe to the Siemens Industrial Cybersecurity RSS Feed under
<https://new.siemens.com/cert>.

Further information is provided on the Internet:

Industrial Security Configuration Manual (<https://support.industry.siemens.com/cs/ww/en/view/108862708>)



WARNING

Unsafe operating states resulting from software manipulation

Software manipulations, e.g. viruses, Trojans, or worms, can cause unsafe operating states in your system that may lead to death, serious injury, and property damage.

- Keep the software up to date.
- Incorporate the automation and drive components into a state-of-the-art, integrated industrial cybersecurity concept for the installation or machine.
- Make sure that you include all installed products in the integrated industrial cybersecurity concept.
- Protect files stored on exchangeable storage media from malicious software by with suitable protection measures, e.g. virus scanners.
- Carefully check all cybersecurity-related settings once commissioning has been completed.

Tool offset

3.1 Introduction

3.1.1 Programmed contour and tool path

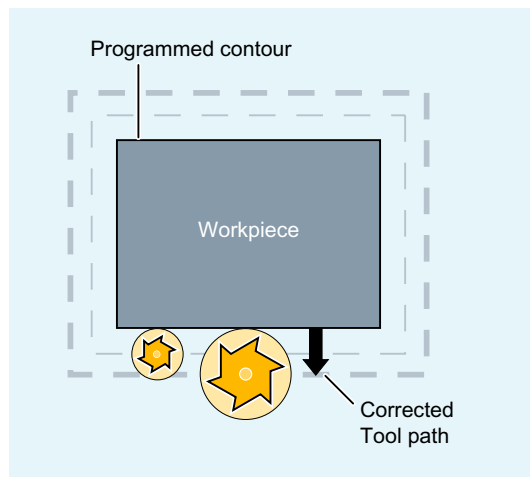
Workpiece dimensions are programmed directly (e.g. according to the production drawing). Therefore, tool data, such as milling tool diameter, cutting edge position of the turning tool (counterclockwise/clockwise turning tool) and tool length, does not have to be taken into consideration when creating the program.

The control corrects the travel path

When machining a workpiece, the tool paths are controlled according to the tool geometry so that the programmed contour can be created with any tool.

So that the control can calculate the tool paths, the tool data must be entered in the tool compensation memory of the control. Only the required tool (T...) and the required offset data record (D...) are called via the NC program.

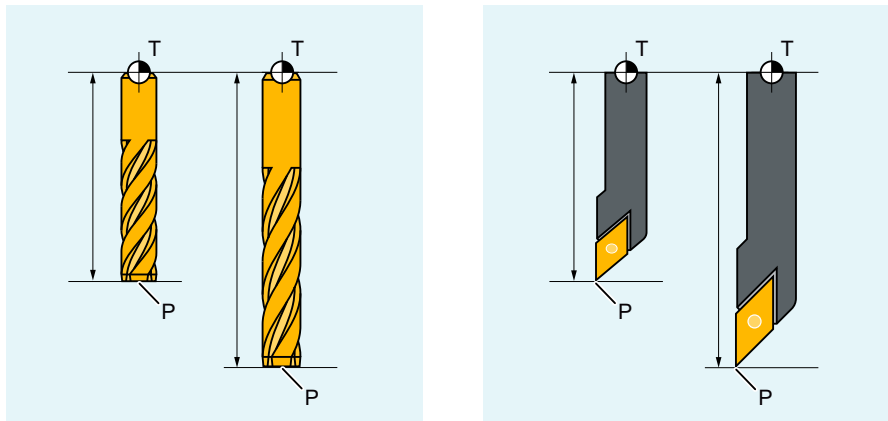
While the program is being processed, the control fetches the offset data it requires from the tool offset memory, and corrects the tool path individually for different tools:



3.1.2 Tool length compensation

The tool length compensation compensates for the differences in length between the tools used.

The tool length is the distance between the tool carrier reference point and the tool tip:



T Tool carrier reference point

P Tool tip

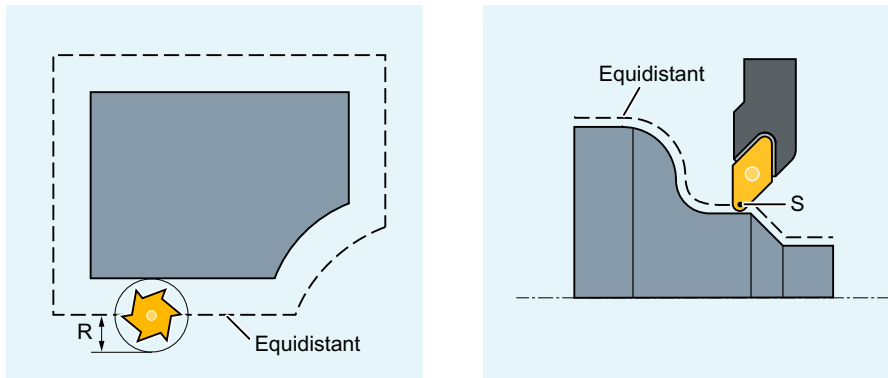
This length is measured and entered in the tool compensation memory of the control together with definable wear values. From this data, the control calculates the traversing movements in the infeed direction.

Note

The correction value of the tool length depends on the spatial orientation of the tool.

3.1.3 Tool radius compensation

The contour and tool path are not identical. The milling tool or cutting edge center point must travel along a path corresponding to the tool radius that is equidistant from the contour (tool center point path). To do this, while executing the program, the control shifts the programmed tool center point path – based on the radius of the active tool (tool offset memory) – so that the tool cutting edge traverses precisely along the programmed contour.



R Tool radius

S Cutting edge center point

The tool radius compensation is described in detail in the "2D tool radius compensation (Page 54)" Chapter.

3.2 Tool change and activation of tool offset

3.2.1 Selection of tool and cutting edge and tool change

Tool selection

A tool is selected in the NC program with the T function.

Syntax

T=<No> / T<n>=<No>

T=<Name> / T<n>=<Name>

Meaning

T	Address for the tool selection			
<No>	With active tool management:	Number of the magazine location		
	Without tool management:	Number of the tool		
		Value range:	0 ... 32000	
			0	No tool (≠ Tool deselection)
			32000	Tool with number 32000
<Name>	Tool name (only possible when tool management is active!) On a machine tool with active tool management, the tools must be assigned a name and number for clear identification (e.g. "Drill", "3"). As a consequence, tool selection is only possible based on the tool name, e.g.: T=" Drill" Note: The correct notation (uppercase/lowercase) must be used when programming a tool name.			
<n>	Spindle number as address extension			

Tool change

Whether the new tool will be loaded immediately by means of the T function depends on the setting in the machine data:

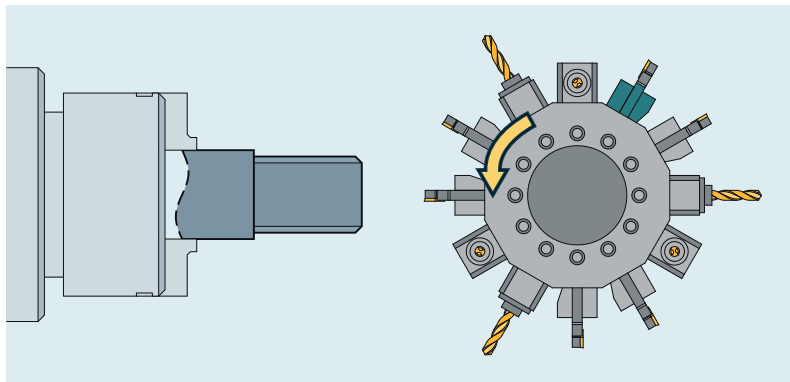
MD22550 \$MC_TOOL_CHANGE_MODE (new tool offset for M function)

Tool change immediate

MD22550 \$MC_TOOL_CHANGE_MODE = 0

The new tool is loaded immediately with the T function.

This setting is primarily used for turning machines with turret magazines.



Tool change with M function

MD22550 \$MC_TOOL_CHANGE_MODE = 1

The new tool is prepared for changing with the T function.

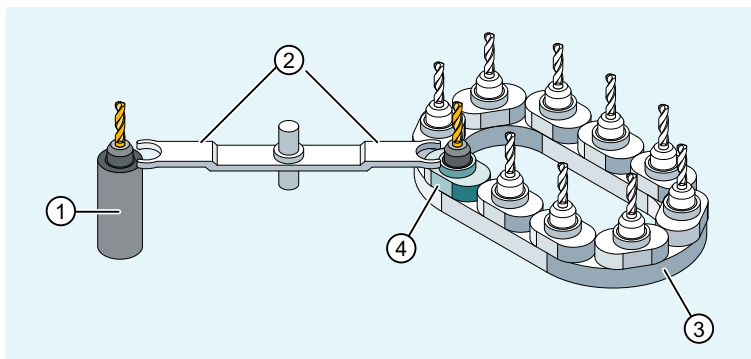
This setting is primarily used for milling machines with tool magazine in order to position the new tool into the tool change position without interrupting the machining process.

The M function for tool change allows the old tool to be removed from the spindle and the new tool to be loaded into the spindle.

The M function for tool change is defined in the machine data:

MD22560 \$MC_TOOL_CHANGE_M_CODE (M function for tool change)

Default setting: M06 (acc. to DIN 66025)



- ① Spindle
- ② Gripper
- ③ Magazine (here: chain magazine)
- ④ Tool change position

Preselected tool

The tool for which the tool length offset values are to be taken into consideration during power up of the controller and for reset or part program end depending on MD20110 \$MC_RESET_MODE_MASK and for part program start depending on MD20112 \$MC_START_MODE_MASK is defined with the machine data:

MD20121 \$MC_TOOL_PRESEL_RESET_VALUE

Tool cutting edge D

Several cutting edges D1 ... D<n> can be assigned to one tool T<x>.

Maximum number of tool cutting edges per tool

The maximum number of tool cutting edges (D numbers) which can be defined for each tool is specified by the following machine data:

MD18106 \$MN_MM_MAX_CUTTING_EDGE_PERTOOL

D function

Selection of the desired tool cutting edge 1 ... <n> for processing is performed in the NC program by programming the corresponding D function D1 ... D<n>.

The selected tool cutting edge always refers to the active tool. A tool cutting edge without active tool (actual T number is T0 ⇒ no tool selected), is not effective.

All tool compensations for the active tool T<x> are deselected with D0.

Selection of cutting edge for tool change

Following selection and loading of a new tool, the following options are provided for selection of the cutting edge:

- Programming of the cutting edge number D<x>.
- The cutting edge number is defined by the machine data:
MD20270 \$MC_CUTTING_EDGE_DEFAULT (basic position of the tool cutting edge without programming)

Value	Meaning
0	No automatic cutting edge selection after M06.
≠ 0	Number of the cutting edge, which is selected in accordance with M06.
-1	The cutting edge number of the old tool is retained and is also selected for the new tool after M06.
-2	The tool nose radius compensation of the old tool remains active until a new cutting edge is programmed with D.

3.2.2 Activation of the tool offset

With D1 to D<n>, the tool offset for a cutting edge is activated for the active tool. Tool length compensation and tool radius compensation become effective at different times, however:

- The **tool length compensation** (TLC) is performed on the first traversing motion of the axis on which the TLC is to act.
This traversing motion must be a linear interpolation (G0, G1, POS, POSA) or polynomial interpolation (POLY). If the POS/POSA axis is one of the active geometry axes, tool length compensation is applied with the first axis motion, in which it is operative.
- **Tool radius compensation** (TRC) becomes active when G41/G42 is programmed in the active plane (G17, G18 or G19).
In the NC block in which the tool radius compensation is activated with G41/G42, G0 (rapid traverse) or G1 (linear interpolation) must be active and at least one geometry axis of the active working plane must be programmed.
If only one geometry axis is specified when switching on, the last position of the second geometry axis is automatically supplemented and then traversed in both axes.
If the axes are not geometry axes, then they must be transformed into geometry axes using GEOAX programming.
The selection of the TRC with G41/G42 is only permitted in a program block with G0 (rapid traverse) or G1 (linear interpolation).

3.3 Tool offset memory

3.3.1 Compensation memory structure

Tool offset memory size

Each channel can have a dedicated tool offset memory (TO unit).

Which tool offset memory exists for the relevant channel is set with the machine data:

MD28085 \$MC_MM_LINK_TOA_UNIT (assignment of TO unit to a channel)

The maximum number of tool cutting edges for all tools managed by the NC is set with the machine data:

MD18100 \$MN_MM_NUM_CUTTING_EDGES_IN_TOA (number of tool cutting edges in NC)

Tools

The tool offset memory consists of tools numbered T1 to T32000.

Each tool can be set up via TOA files or individually, using the "New tool" softkey. Offset values not required must be assigned the value zero. This is the default setting when the

offset memory is created. The individual values in the offset memory (tool parameters) can be read and written from the program using system variables.

Note

The tools (T1 to T32000) do not have to be stored in ascending order or contiguously in the tool offset memory, and the first tool does not have to be assigned number T1.

Tool cutting edges

Each tool can have up to 12 cutting edges (D1 to D12). The first cutting edge (D1) is set up automatically when a new tool is loaded to the tool offset memory. Other cutting edges (max. 11) are set up consecutively and contiguously using the "New cutting edge" softkey. A different number of tool cutting edges can be assigned to each tool in this way.

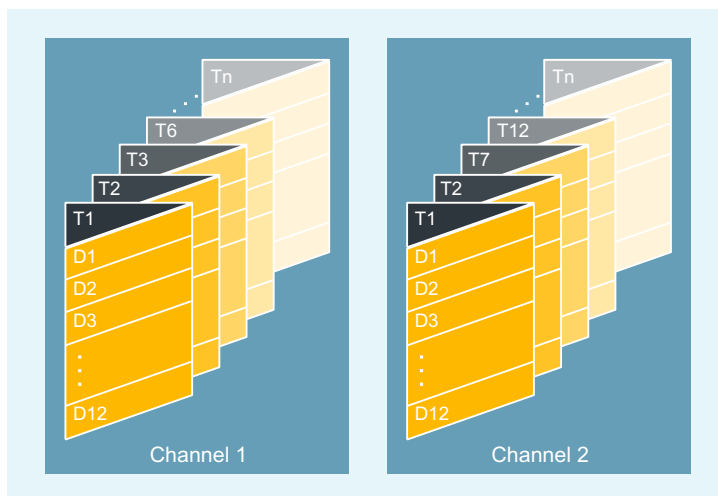


Figure 3-1 Example of a tool offset memory structure for two channels

3.3.2 Calculating the tool compensation

D No.

The **D No.** is sufficient for determining tool compensation values (can be set via MD).

D-Nr.	DP1	DP2	DP3	...	...	DP25
D1						
D2						
⋮						
Dn		CV	CV			CV

CV Total compensation value

Programming

The above compensation block is to be taken into account in the NC.

Part program call:

```

...
Dn

```

3.3.3 Address extension for NC addresses T and M

MD20096

Whether the address extension of T and M is also to be interpreted as spindle number when the tool management is **not** activated, can be set via the machine data:

MD20096 \$MC_T_M_ADDRESS_EXT_IS_SPINO (spindle number as address extension)

The same rules then apply to the relationship between the D number and T number as when the "Tool management" function is active.

Effect on the D number

An offset data set is determined by the D number.

The D address cannot be programmed with an address extension.

The evaluation of the D address always refers to the currently active tool.

The programmed D address refers to the active tool in relation to the master spindle (same as for tool management function), when machine data is set:

MD20096 \$MC_T_M_ADDRESS_EXT_IS_SPINO = TRUE (spindle number as address extension).

Effect on the T number

If the "Tool management" function is active, the values programmed with reference to the master spindle (or master toolholder) are displayed as programmed/active T numbers.

If tool management is not active, **all** programmed T values are displayed as programmed/active, regardless of the programmed address extension.

Only the T value programmed in relation to the master spindle is shown as programmed/active, when:

MD20096 \$MC_T_M_ADDRESS_EXT_IS_SPINO = TRUE (spindle number as address extension).

Example

The following example below shows the effect of MD20096.

Two spindles are considered. Spindle 1 is the master spindle. M6 was defined as the tool change signal.

T1 = 5	
M1 = 6	
T2 = 50	
M2 = 6	
D4	

- If tool management is active, D4 refers to tool "5".
T2=50 defines the tool for the secondary spindle, whose tool does not influence the path compensation. The path is determined exclusively by the tool programmed for the master spindle.
- D4 refers to tool "50" without active tool management and with the machine data:
MD20096 \$MC_T_M_ADDRESS_EXT_IS_SPINO = **FALSE** (significance of address extension for T, M tool change).
The address extensions of neither T nor M are evaluated in the NC.
Each tool change command defines a new path compensation.
- D4 refers to tool "5" (as when tool management is active) without active tool management and with the machine data:
MD20096 \$MC_T_M_ADDRESS_EXT_IS_SPINO = **TRUE**.
Address extension 1 (T1= ..., M1= ...) addresses the master spindle.

Note

Previously, when tool management was not activated, each tool change command (programmed with T or M) caused the tool offset to be recalculated in the path. The address extension is not defined further by this operation. The significance of the extension is defined by the user (in the PLC user program).

3.3.4 Free assignment of D numbers

"Relative" D numbers

In the NC it is possible to manage the D numbers as "relative" D numbers for the tool offset data sets. The corresponding D numbers are assigned to each T number.

Functions

Expansions to functions when assigning D numbers:

- The maximum permitted D numbers are defined via the machine data:
MD18105 \$MN_MM_MAX_CUTTING_EDGE_NO (max. value of the D numbers (DRAM))
The default value is 9, in order to maintain compatibility with existing applications.
- The number of cuts (or the offset data sets) **for each tool** can be defined via the machine data:
MD18106 \$MN_MM_MAX_CUTTING_EDGE_PERTOOL (max. number of the D numbers per tool (DRAM))
This allows you to customize the number of cutting edges to be configured for each tool to the actual number of real cutting edges for monitoring purposes.
- It is also possible to rename D numbers in the NC and thus to allocate arbitrary D numbers to the cutting edges.

Cutting edge number CE

When you rename D numbers, the information in the tool Catalog detailing the numbers defined for these cutting edges is lost. It is, therefore, impossible to determine, following renaming, which cutting edge of the Catalog is being referenced.

Since this information is required for retooling procedures, a **cutting edge number CE** has been introduced for each cutting edge. This number remains stored when the D number is renamed.

The D number identifies the cutting edge compensation in the part program. This **compensation number D** is administered separately from the **cutting edge number CE** (the number in the tool Catalog). Any number can be used. The number is used to identify a compensation in the part program and on the display.

The CE number identifies the actual physical cutting edge during retooling. The cutting edge number CE is not evaluated by the NC on compensation selection during a tool change (only available via the OPI).

The cutting edge number CE is defined with system variable **\$TC_DPCE[t,d]**:

- **t** stands for the internal T number
- **d** stands for the D number.

Write accesses are monitored for collisions, i.e. all cutting edge numbers of a tool must be different. The variable \$TC_DPCE is a component of the cutting edge parameter data set \$TC_DP1 to \$TC_DP25.

It is only practical to parameterize \$TC_DPCE if the maximum cutting edge number (MD18105) is greater than the maximum number of cutting edges per tool (MD18106).

In this case, the default cutting edge number is the same as the classification number of the cutting edge. Compensations of a tool are created starting at number 1 and are incremented up to the maximum number of cutting edges per tool (MD18106).

The cutting edge number CE is the same as the D number (in compatibility with the behavior till now) if:

$MD18105 \leq MD18106$.

A read operation returns $CE=D$. A write operation is ignored without an alarm message.

Note

The offset values \$TC_DP1 to \$TC_DP25 of the active tool offset can be read with system variable \$P_AD[n], with n=1 to 25. The CE cutting edge number of the active compensation is returned with n=26.

Commands

If the maximum cutting edge number is **higher** than the maximum number of cutting edges for each tool, then the commands listed in the following table, are available

- $(MD18105 \$MN_MM_MAX_CUTTING_EDGE_NO) > (MD18106 \$MN_MM_MAX_CUTTING_EDGE_PERTOOL)$

Command	Meaning
CHKDNO	Checks the uniqueness of the available D numbers; The D numbers of all tools defined within a TO unit may not occur more than once. No allowance is made for replacement tools.
GETDNO	Determines the D number for the cutting edge of a tool. If no D number matching the input parameters exists, d=0. If the D number is invalid, a value greater than 32000 is returned.
SETDNO	Sets or changes the D number of the CE cutting edge of tool T. If there is no data set for the specified parameter, the value FALSE is returned. Syntax errors generate an alarm. The D number cannot be set explicitly to 0.
GETACTTD	Determines the associated T number for an absolute D number. There is not check for uniqueness. If several D numbers within a TO unit are the same, the T number of the first tool found in the search is returned.
DZERO	Marks all D numbers of the TO unit as invalid; This command is used for support during retooling. Compensation data sets tagged with this command are no longer verified by the CHKDNO language command. These data sets can be accessed again by setting the D number once more with SETDNO.

Note

If the maximum cutting edge number is **less** than the maximum number of cutting edges for each tool, then the commands listed in the table are ineffective. This default setting is in the system for reasons of compatibility.

The individual commands are described in detail in:

Further information

Programming Manual Fundamentals

Activation

In order to work with unique D numbers and, therefore, with the defined language commands, it must be possible to name D numbers freely for the tools.

To do this, the following precondition must be satisfied:

- MD18105 > MD18106

Examples

MD18105 \$MN_MM_MAX_CUTTING_EDGE_NO = 1 (max. value of the D numbers)

A maximum of one compensation can be defined per tool (with D number = 1).

MD18105 \$MN_MM_MAX_CUTTING_EDGE_NO = 9999

Tools can be assigned unique D numbers.

Example: Assigning D (cutting edges) to T numbers (tools) and listing options for checking uniqueness:

- Assignment:
 - T No. 1 $\Rightarrow$ assignment of D numbers 1, 2, 3
 - T No. 2 $\Rightarrow$ assignment of D numbers 10, 20, 30, 40, 50
 - T No. 3 $\Rightarrow$ assignment of D numbers 100, 200, 30 (**error**: 30 was entered instead of 300)
- Check for uniqueness:
 The D numbers of all tools defined within a TO unit must be unique.
 The check is made using the `CHKDNO` command. When no parameters are specified, all D numbers of all tools are checked with respect to one another. In this particular case, the return value == `FALSE`, as D30 is available in tool T2 and T3.

Command	Test	Test result
<code>CHKDNO</code>	All D numbers of all tools are checked against all other tools	Return value == <code>FALSE</code> , as D=30 is duplicated.
<code>CHKDNO (2, 3, 30)</code>	D number 30 in tools 2 and 3	Return value == <code>FALSE</code> , as D=30 is duplicated.
<code>CHKDNO (1, 3)</code>	All D numbers of tools 1 and 3	Return value == <code>TRUE</code> although there is a collision between the D=30 of the third tool and D=30 of the second tool.

Note

Cutting edges per tool

If tools with **n** cutting edges are used, then the value of the machine data should be set to **n**. This prevents inadmissible cutting edges to be defined for a tool.

MD18106 \$MN_MM_MAX_CUTTING_EDGE_PERTOOL = **n**

Programming examples

Renaming a D number

The D number of cutting edge CE = 3 is to be renamed from 2 to 17. The following specifications apply:

- `$TC_DPx[ <tool Tn>, <cutting edge Dm> ]`
- Internal T number $T_n = 1$
- D number $D_m = 2$
- Tool with one cutting edge with:

Program code	Comment
<code>\$TC_DP3[1, 2] = 120</code>	Tool length = 120
<code>\$TC_DP6[1, 2] = 5.5</code>	Tool radius = 5.5
<code>\$TC_DPCE[1, 2] = 3</code>	Cutting edge number CE = 3

- MD18105 \$MN_MM_MAX_CUTTING_EDGE_NO = 20 (max. value of the D numbers)

Within the part program, this compensation is programmed as standard with T1, ... D2.

3.3 Tool offset memory

You assign the current D number of cutting edge 3 to a variable (DNoOld) and define the variable DNoNew for the new D number:

Program code

```
def int DNoOld, DNoNew = 17
DNoOld = GETDNO( 1, 3 )
SETDNO( 1, 3, DNoNew )
```

The new D value 17 is then assigned to cutting edge CE=3.

Now the data for the cutting edge are addressed via D number 17, both via the system variable and in programming with the NC address D.

This compensation is now programmed in the part program with T1, ...D17 and the data is addressed as follows:

Program code
Comment

```
$TC_DP3[ 1, 17 ] = 120
$TC_DP6[ 1, 17 ] = 5.5
$TC_DPCE[ 1, 17 ] = 3 ; Cutting edge number CE
```

Note

If the tool has defined a further cutting edge, e.g. \$TC_DPCE[1, 2] = 1 ; = CE, D number 2 of cutting edge 1 cannot have the same name as the D number of the cutting edge 3, i.e.: SETDNO(1, 1, 17) returns the status = FALSE as return value.

DZERO - Invalidate D numbers

The activation of this command invalidates all D numbers of the tools in the TO unit. It is no longer possible to activate a compensation until valid D numbers are again available in the NC. The D numbers must be reassigned using the SETDNO command.

The following tools must be defined (all with cutting edge number 1):

T1, D1	D no. of cutting edge CE=1
T2, D10	D no. of cutting edge CE=1
T3, D100	D no. of cutting edge CE=1

The following command is then programmed:

Program code

```
DZERO
```

If one of the compensations is now activated (e.g. with T3 D100), an alarm is generated, because D100 is not currently defined.

The D numbers are redefined with:

Program code
Comment

```
SETDNO( 1, 1, 100 ) ; T=1, cutting edge 1 is assigned the (new) D number 100
SETDNO( 2, 1, 10 ) ; T=2, cutting edge 1 is assigned the (old) D number 10
SETDNO( 3, 1, 1 ) ; T=3, cutting edge 1 is assigned the (new) D number 1
```


Note

In the event of a power failure, the DZERO command can leave the NC in an undefined state with reference to the D numbers. If this happens, repeat the DZERO command when the power is restored.

Operating principle of a retooling program

Let us assume you want to ensure that the required tools and cutting edges are available. The status of the tool-holding magazine of the NC is arbitrary. The D numbers in the part programs for the new machining operation generally do not match the D numbers of the actual cutting edges. The retooling program can have the following appearance:

Program code	Comment
DZERO	; All D numbers of the TO unit are tagged as invalid
....	; One or more loops over the locations of the magazine to check the tools and their cutting edge numbers.
	; If a tool is found, which is still enabled (\$TC_TP8) and has the required cutting edge number CE (GETDNO), the new D number is allocated to the cutting edge (SETDNO).
....	; Loading and unloading operations are performed.
	; It is possible to work with the tool status 'to be unloaded' and 'to be loaded'.
CHKDNO	; Loading/unloading and the operation for renaming D numbers are complete.
	; Individual tools and/or D numbers can be checked, and collisions can be handled automatically according to the return value.

3.3.5 Compensation block in case of error during tool change

MD22550

If a tool preparation has been programmed in the part program and the NC detects an error (e.g. the data set for the programmed T number does not exist in the NC), the user can assess the error situation and perform appropriate tasks, in order to subsequently resume machining.

The tool change may be programmed independently, depending on the machine data:

MD22550 \$MC_TOOL_CHANGE_MODE (new tool offset for M function)

MD22550 \$MC_TOOL_CHANGE_MODE = 0

T= 'T no.'	; Tool preparation + tool change in one NC block,
	; i.e. when T is programmed a new D compensation becomes
	; active in the NC. See machine data:
	; MD20270 \$MC_CUTTING_EDGE_DEFAULT (basic position of the tool cutting edge without programming)

MD22550 \$MC_TOOL_CHANGE_MODE = 1

```

T= 'T no.'      ; Tool preparation
M06             ; Tool change
                ; (the number of the tool-change M code can be set),
                ; i.e. when M06 is programmed, a new D compensation becomes
                ; active in the NC (see
                ; machine data MD20270 $MC_CUTTING_EDGE_DEFAULT)

```

If the tool management is not active, the following problems can be detected:

- D compensation data set missing
- Error in the part program

Note

The "tool is not in the magazine" problem cannot be detected since the NC did not have access to any magazine information for the tool offset.

D compensation data set missing

Program execution is interrupted at the block containing the invalid D value (regardless of the value of machine data MD22550). The operator must either correct the program or reload the missing data set. To do this, he requires the D and T number. These parameters are transferred when the alarm is triggered.

Error in the part program

The options for intervention in the event of an error depend on how the tool change was programmed, defined via the machine data:

MD22550 \$MC_TOOL_CHANGE_MODE (new tool offset for M function)

Tool change with T programming (MD22550 = 0)

In this case, the "Compensation block" function available in the NC is used. The NC program stops at the NC block in which an error was detected for the programmed T value. The "Compensation block" is executed again when the program is resumed.

The operator can perform the following:

- Correct the part program.
- Reload the missing cutting edge compensation data from the HMI.
- Include the missing cutting edge compensation data in the NC using "Overstore".

Following operator intervention, the *START* key is pressed and the block, which caused the error, is executed again. If the error was corrected, the program is resumed. Otherwise, an alarm is output again.

Tool change with T and M06 programming (MD22550 = 1)

In this case, an error is detected in the NC block of the tool preparation (programming of T), which however is to be ignored at first. Execution is continued until the tool change request (usually M06) is executed in the NC program. The program is to stop at this point.

The programmed T address can be any number of program lines before the M06 command, or the two instructions can be in different (sub)programs. For this reason, it is not usually possible to modify a block which has already been executed via the compensation block.

The operator has the same options for intervention as with = 0.

Reloading of missing data is possible. In this case, however, T must be programmed with "Overstore".

If a program error has occurred, the line with the error cannot be corrected (Txx); only the line at which the program stopped and which generated the alarm can be edited. Only when machine data:

MD22562 \$MC_TOOL_CHANGE_ERROR_MODE bit 0 = 1 (response to errors during tool change)

This results in the following sequence:

```
Txx          ; Error! Data set with xx does not exist
              ; Note state; note xx;
              ; continue in program
....
M06          ; Detect bit memory "xx missing" → output alarm,
              ; stop program
              ; Correct block with, e.g. Tyy M06, start,
              ; block Tyy M06 is interpreted and is OK.
              ; Execution continues.
```

Renewed execution of the program points results in the following:

```
Txx          ; Error! Data set with xx does not exist,
              ; Note state; note xx;
              ; continue in program
....
Tyy M06      ; Detect bit memory "xx missing" → reject without further response,
              ; as Tyy M06 is correct → program does not stop (correct).
```

If required, the original point of the T call can be corrected at the end of the program. If the tool change logic on the machine cannot process this, the program must be aborted and the point of the error corrected.

If only one data set is missing, it is transferred to the NC, Txx is programmed in "Overstore" and the program is subsequently resumed.

As in the case of "missing D number", the required parameter (T number) can be accessed by the user for "missing T number" via the appropriate alarm (17191).

Note

In order to enable program correction, it stops immediately at the faulty Txx block.

The program test mode is also stopped when machine data:

MD22562 \$MC_TOOL_CHANGE_ERROR_MODE bit 0=1 (response to errors during tool change).

3.3.6 Definition of the effect of the tool parameters

MD20360

The effect of the tool parameters on the transverse axis in connection with diameter programming can be controlled selectively with the machine data:

MD20360 \$MC_TOOL_PARAMETER_DEF_MASK (definition of tool parameters)

Details are described with the mentioned MD.

DRF manual handwheel traversing with half distance

During DRF handwheel traversal, it is possible to move a transverse axis through only half the distance of the specified increment as follows:

Specify the distance with handwheel via the machine data:

MD11346 \$MN_HANDWHEEL_TRUE_DISTANCE = 1 (handwheel path or speed specification)

Define the DRF offset in the transverse axis as a diameter offset with the machine data:

MD20360 \$MC_TOOL_PARAMETER_DEF_MASK bit 9 = 1 (definition of tool parameters)

Deselecting an axial DRF offset (DRFOF) also deletes an existing tool offset (handwheel override in tool direction).

Note

For further information about superimposed movements with the handwheel, please refer to:

Further information

Function Manual Axes and Spindles; Manual Traversing

Programming Manual Fundamentals

(The Programming Manual describes the complete technical program options in order to deselect the DRF offset axis-specifically.)

3.4 Tool parameters

3.4.1 Overview

A tool cutting edge is clearly defined with the following details:

- Tool type (end mill, drill, etc.)
- Geometrical description
- Technological description

This data is mapped to the tool parameters. Which individual tool parameters are required depends on the particular tool type. Depending on the specific tool type, several tool parameters can have different meanings.

The following table provides an overview:

Tool parameter	Meaning	
\$TC_DP1	Tool type	
\$TC_DP2	Cutting edge position	For turning tools or for milling/grinding tools with 2D TRC with contour tools.
Geometry - tool lengths		
\$TC_DP3	Geometry length 1	
\$TC_DP4	Geometry length 2	
\$TC_DP5	Geometry length 3	
Geometry - tool shape		
\$TC_DP6 ¹⁾	Geometry radius 1 (tool radius 1)	2D TRC
	Cutting edge radius	Turning tools
	Tool radius	3D face milling
	Diameter	Grooving saw
\$TC_DP7 ¹⁾	Geometry radius 2	Tools with corner rounding
	Outside radius 1	Facing tool
	Corner radius	Milling tools with corner rounding
	Groove width	Grooving saw
\$TC_DP8 ¹⁾	Geometry length 4	Milling/grinding tools
	Cutting tip length	Turning tools
	Internal circuit - Prime	Prime tools
	Projection	Grooving saw

3.4 Tool parameters

Tool parameter	Meaning	
\$TC_DP9 ¹⁾	Geometry length 5	Milling/grinding tools
	Width	Side mill
		Saw
	Cutting tip width	Turning tools
	Pitch	Tap
	Tool radius rotary drill (drilled hole radius)	Rotary drill
	Cutting speed (ManualTurn)	Milling tools
	Upper ball diameter	Bevel cutter, tapered die milling tool → 3D face milling
	Reference length outside radius	Barrel milling cutter
\$TC_DP10 ¹⁾	Geometry angle 1	Milling/grinding tools
	Minimum limit angle	2D TRC with contour tools
	Holder angle	Turning tools
	Tip angle	Drill and thread milling cutter
	Correction angle	Probe
\$TC_DP11 ¹⁾	Geometry angle 2	2D TRC with contour tools: Maximum limit angle 3D face milling: Angle between envelope line and tool longitudinal axis (of conical tools)
	Cutting direction	Turning tools
	Angle	Tapered milling cutters
Wear - tool lengths		
\$TC_DP12	Wear length 1	
\$TC_DP13	Wear length 2	
\$TC_DP14	Wear length 3	
Wear - tool shape		
\$TC_DP15	Wear radius 1	
\$TC_DP16	Wear radius 2	
\$TC_DP17	Wear length 4	
\$TC_DP18	Wear length 5	
\$TC_DP19 ¹⁾	Wear angle 1	
	Inclined position	Oblique turning tools
\$TC_DP20	Wear angle 2	
Tool base dimension / adapter dimension		
\$TC_DP21	Base length 1	
\$TC_DP22	Base length 2	
\$TC_DP23	Base length 3	

Tool parameter	Meaning	
Technology		
\$TC_DP24 ¹⁾	Clearance angle	Turning tools
	Number of teeth	Milling tools
	Tip angle	Drilling tools
Basic tool orientation		
\$TC_DPV	Predefined orientation vector	
\$TC_DPV3	Basic tool orientation: L1 component of the orientation vector	
\$TC_DPV4	Basic tool orientation: L2 component of the orientation vector	
\$TC_DPV5	Basic tool orientation: L3 component of the orientation vector	
\$TC_DPVN3	Basic tool orientation: L1 component of the normal vector	Only for non-rotationally symmetrical tools.
\$TC_DPVN4	Basic tool orientation: L2 component of the normal vector	Only for non-rotationally symmetrical tools.
\$TC_DPVN5	Basic tool orientation: L3 component of the normal vector	Only for non-rotationally symmetrical tools.
\$TC_DPNT	Number of teeth, cutting edge parameter 34	
Technology		
\$TC_DPNT	Number of teeth	Milling tools
\$TC_DPROT	Clamping angle	Only for turning tools.
\$TC_DP36 ¹⁾	Outside radius 2	Barrel milling cutter, tapered milling cutter
\$TC_DP37 ¹⁾	Wear outside radius 2	Barrel milling cutter, tapered milling cutter
Other		
\$TC_DP_RADIUS3 ¹⁾	Geometry radius 3	
\$TC_DP_WEAR_RADIUS3 ¹⁾	Wear radius 3	
\$TC_DP_LEN6 ¹⁾	Geometry length 6	
\$TC_DP_WEAR_LEN6 ¹⁾	Wear length 6	
\$TC_DPCE [<t>,<d>]	System variable of an offset data record with T=<t> and D=<d> containing cutting edge number CE. (Unique D no. or user assignment of D no. for cutting edge numbers)	Value range for permissible cutting edge numbers: 1, 2, 3 ... MD18106
\$TC_DPH [<t>,<d>]	H parameter (Y / extraCuttEdgeParams), Bit 0 = 1	

¹⁾ These tool parameters have different meanings depending on the tool type.

3.4.2 Tool parameter 1: Tool type

This tool type defines the respective tool. The selection of this tool type determines further components such as geometry, wear and tool base dimensions in advance.

Each tool type is assigned a unique 3-digit number. The assignment of the tool to one of the following technologies or tool groups is realized using the first digit (the hundreds position):

Tool type	Tool group
1xy	Milling cutters
2xy	Drilling tools
3xy	Reserved
4xy	Grinding tools
5xy	Turning tools
6xy	Reserved
7xy	Special tools

The following is applicable to the "Tool type" parameter:

- The tool type must be specified for each tool cutting edge.
- Only the values specified can be used for the tool type.
- Tool type "0" means that no valid tool has been defined.

Relevant tool parameters

The parameter values to be entered in accordance with the tool type (tool offset data; TOA data) are stored in system variables (\$TC_DPx and \$TC_TPGx).

The following tables show which parameter values are required for which tool type:

1) CLDATA = "cutter location data" (tool position data according to DIN66215)

3.4 Tool parameters

Cutting edge parameter \$TC_DPx																								Tool parameter \$TC_TPGx						
x =	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	3	4	5	6	7	8
Tool type		Cutting edge position		Geometry - tool length	Radius 1	Radius 2	Length 4	Length 5	Angle 1	Angle 2	Length 1	Length 2	Length 3	Radius 1	Radius 2	Length 4	Length 5	Angle 1	Angle 2	Length 1	Length 2	Length 3	Clearance angle	Min. wheel radius	Min. wheel width	Act. wheel width	Max. speed	Max. peripheral speed	Angle of inclined wheel	
Grinding tools	400 Surface grinding wheel	✓	✓	✓	✓	✓					✓	✓	✓	✓							✓	✓	✓		✓	✓				
	401 Surface grinding wheel with monitoring	✓	✓	✓	✓	✓					✓	✓	✓	✓							✓	✓	✓		✓	✓				
	403 As with 401, but without base dimension for GWPS ¹⁾	✓	✓	✓	✓	✓					✓	✓	✓	✓							✓	✓	✓		✓	✓				
	410 Faceplate	✓	✓	✓	✓	✓					✓	✓	✓	✓							✓	✓	✓					✓		
	411 Faceplate with monitoring	✓	✓	✓	✓	✓					✓	✓	✓	✓							✓	✓	✓		✓	✓				
	412 Faceplate without monitoring	✓	✓	✓	✓	✓					✓	✓	✓	✓							✓	✓	✓		✓	✓		✓		
	413 As with 411, but without base dimension for GWPS ¹⁾	✓	✓	✓	✓	✓					✓	✓	✓	✓							✓	✓	✓		✓	✓				
	490 Dresser	✓	✓	✓	✓	✓					✓	✓	✓	✓							✓	✓	✓							
SW ²⁾	700 Grooving saw		✓	✓	✓	✓	✓	✓			✓	✓	✓	✓	✓	✓				✓	✓	✓								

¹⁾ GWPS = Grinding wheel peripheral speed

²⁾ ST = Special tools

Figure 3-3 Tool parameters for grinding and special tools

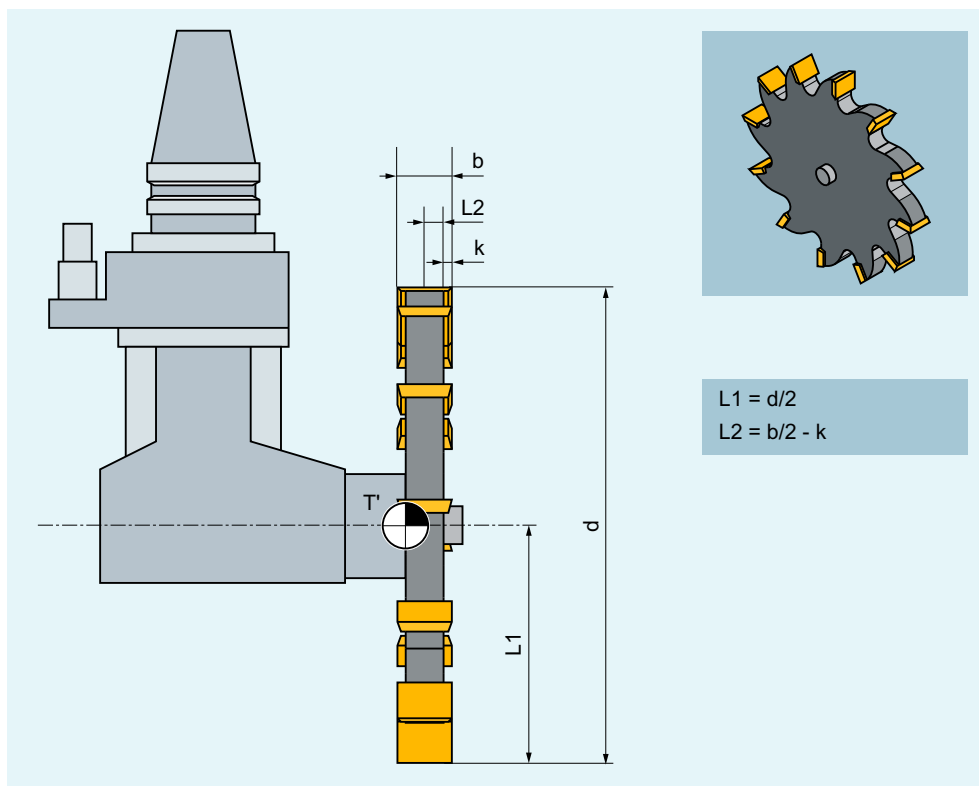
Note

Unlisted numbers are also permitted, in particular with grinding tools (400-499).

Calculating tool offset data

The CNC combines the individual components in the tool offset data to produce a value (e.g. total length, total radius). The relevant overall dimension becomes effective when the offset memory is activated. The use of these values in the axes is determined by the tool type and current machining plane G17, G18 or G19.

Example: Grooving saw tool type (Type 700)



- T' Tool carrier reference point
 L1 Geometry - length 1
 L2 Geometry - length 2
 d Diameter
 b Groove width
 k Projection

	Geometry	Wear	Tool base dimension	Unit
Tool length compensation				
Length 1	\$TC_DP3	\$TC_DP12	\$TC_DP21	mm
Length 2	\$TC_DP4	\$TC_DP13	\$TC_DP22	mm
Length 3	\$TC_DP5	\$TC_DP14	\$TC_DP23	mm
Saw blade compensation				
Diameter d	\$TC_DP6	\$TC_DP15		mm
Groove width b	\$TC_DP7	\$TC_DP16		mm
Projection k	\$TC_DP8	\$TC_DP17		mm

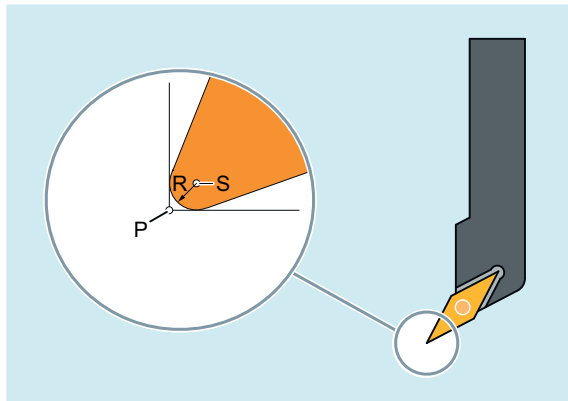
The width of the saw blade is taken into account for the tool radius compensation (G40/G41/G42) as follows:

Function	Meaning
G40	No saw blade compensation
G41	Saw blade compensation left
G42	Saw blade compensation right

3.4.3 Tool parameter 2: Cutting edge position

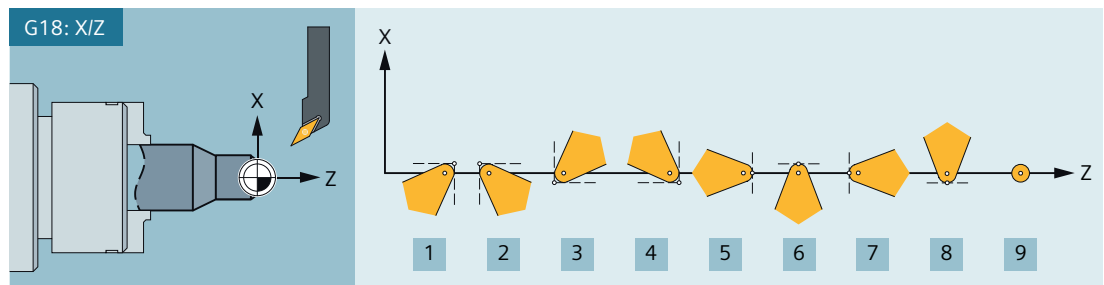
The cutting edge position describes the position of the tool tip P in relation to the cutting edge center point S. It is entered in tool parameter 2.

The cutting edge position is required together with the cutting edge radius (tool parameter 8) for the calculation of the tool radius compensation for turning tools (tool type 5xx).

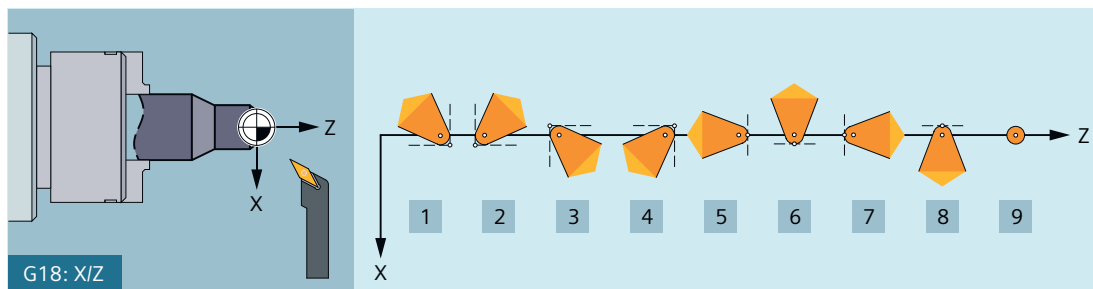


- P Tool tip
- S Cutting edge center point
- R Cutting edge radius

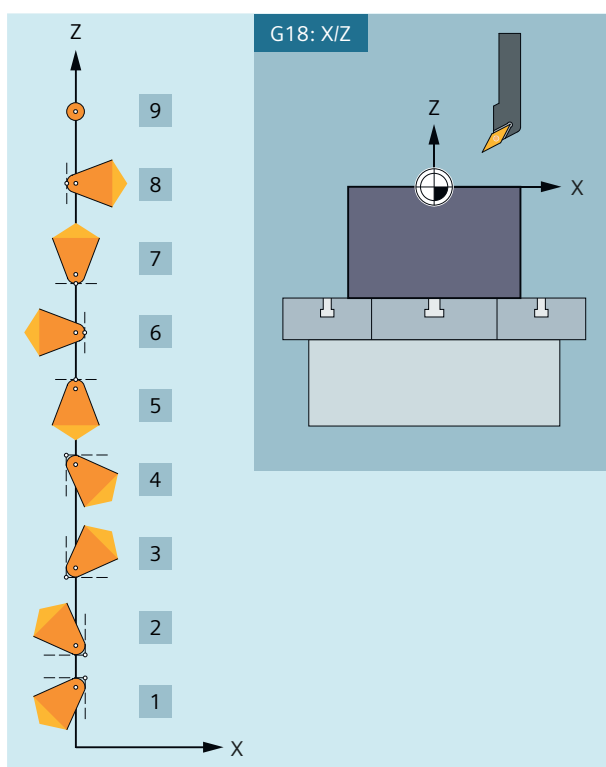
Cutting edge position (1 ... 9) for machining behind the turning center



Cutting edge position (1 ... 9) for machining in front of the turning center



Cutting edge position (1 ... 9) for vertical boring and turning mills



Special aspects

- If the cutting edge center point S is used instead of point P as a reference point to calculate the tool length compensation, the identifier 9 must be entered for the cutting edge position.
- The identifier 0 (zero) is not permitted as a cutting edge position.

3.4.4 Tool parameters 3 ... 5: Geometry - tool lengths

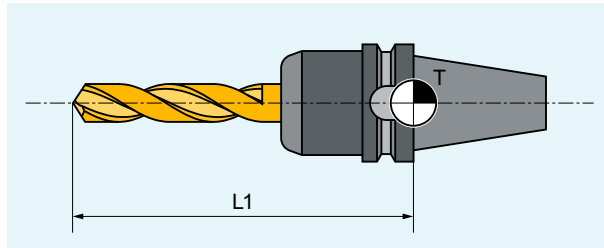
The lengths of the tools are required for the geometry tool length compensation. They are input as tool lengths 1 to 3 in the tool parameters 3 to 5.

3.4 Tool parameters

The minimum required tool length depends on the respective tool type:

Tool type	Required tool lengths
Tool type 12x, 140, 145, 150	Tool length 1
Tool type 13x	Tool length 1 ... 3 (depending on plane G17/G18/G19)
Tool type 2xx	Tool length 1
Tool type 5xx	Tool plane 1 ... 3

Example: Twist drill (tool type 200)



- T Tool carrier reference point
- L1 Length 1 (tool parameter 3)

Note

All three tool parameters 3 to 5 (tool length 1 to 3) are always calculated in the three geometry axes, irrespective of the tool type. If more tool lengths are input in the tool parameters 3 to 5 for a tool type than is required as the minimum, then these extra tool lengths are settled in the geometry axes without any alarm.

The active size of the tool is only defined when the geometry tool length compensation (tool parameters 3 to 5) and the wear tool length compensation (tool parameters 12 to 14) are added together. The base-dimension/adaptor-dimension tool length compensation is also added in order to calculate the total tool length compensation in the geometry axes.

3.4.5 Tool parameters 6 ... 11: Geometry - tool shape

The shape of the tool is defined using the tool parameters 6 to 11. The data is required for the geometry tool radius compensation.

In most cases, only tool parameter 6 (tool radius 1) is used:

Tool parameter	Meaning	Use
6	Tool radius 1	The tool radius must be specified for the following tool types:
		1xx Milling tools
		5xx Turning tools Tool parameter 2: Cutting edge position (Page 44) must also be specified for turning tools.
		Note Tool type 2xx: A tool radius does not have to be specified for drilling tools.
7	Tool radius 2	Not used
8	Tool length 4	Not used
9	Tool length 5	Not used
10	Tool angle 1	Not used
11	Tool angle 2	Not used

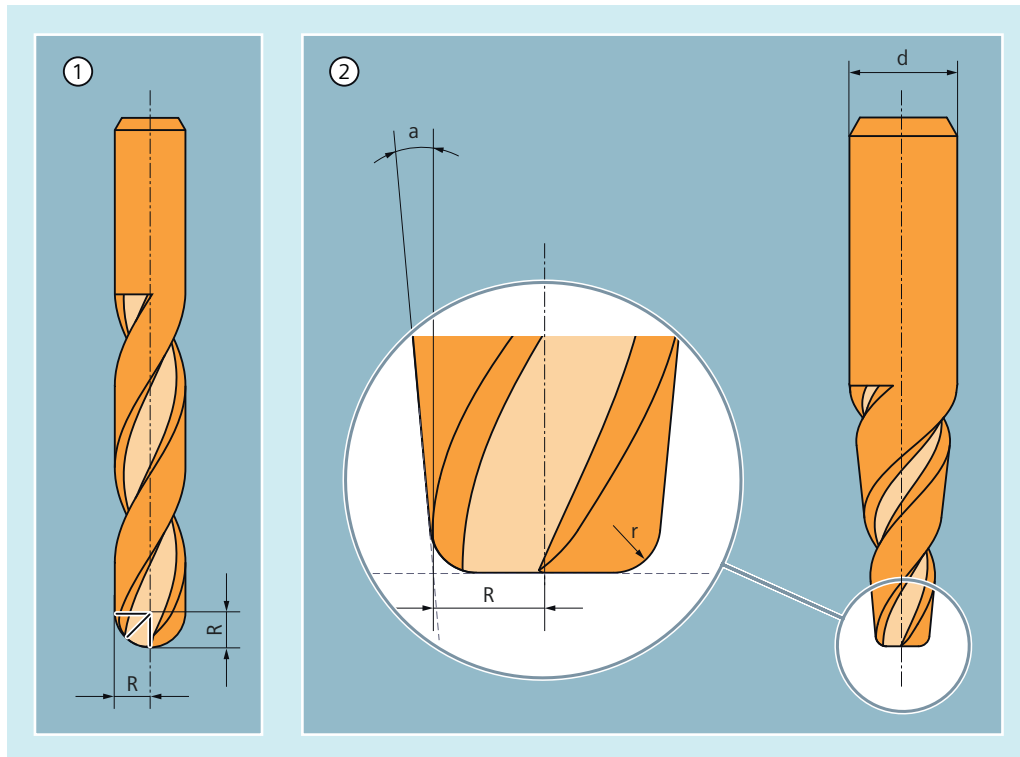
2D TRC with contour tools

For the definition of contour tools with multiple tool cutting edges, the minimum and maximum limit angle can be entered. Both limit angles each relate to the vector of the cutting edge center point to the cutting edge reference point and are counted clockwise.

Tool angle 1	Minimum limit angle per tool cutting edge
Tool angle 2	Maximum limit angle per tool cutting edge

3D face milling

The tool parameters relevant to the tool description in 3D face milling are dependent on the tool type used. Thus, for example, for a ball end mill, only tool parameter 6, for a bevel cutter with corner rounding additionally tool parameters 7, 9 and 11, are relevant.



① Ball end mill

② Bevel cutter with corner rounding

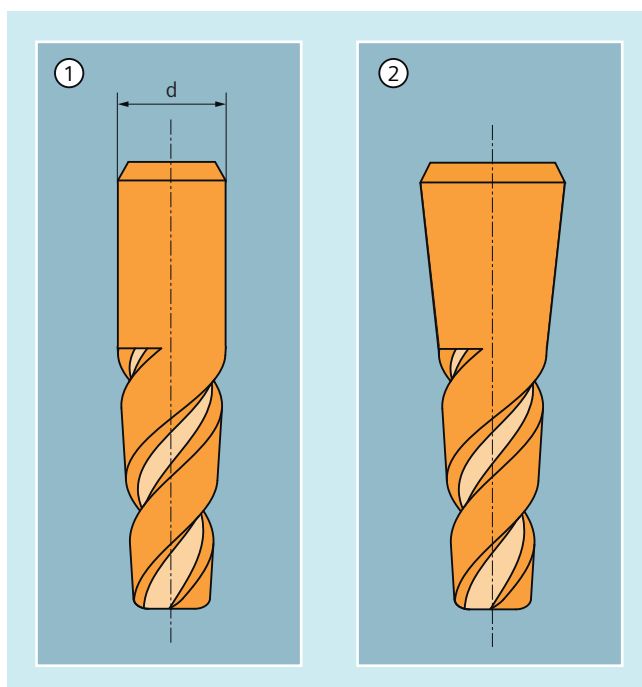
R Tool parameter 6: Tool radius

r Tool parameter 7: Corner radius

d Tool parameter 9: Upper bevel cutter diameter

a Tool parameter 11: Angle between envelope line and tool longitudinal axis

Specification of tool parameter 9 (upper bevel diameter) is optional. If this tool parameter is not specified, a conical shape stretching along the entire tool length is assumed:



- ① With specification of tool parameter 9
- ② Without specification of tool parameter 9

Further information

Please refer to the following documentation for information about entering tool shapes in tool parameters 6 to 11 and how these are calculated by geometry tool radius compensation in the three geometry axes:

- Programming Manual NC programming; Section "Fundamentals" > "Tool offsets" > "2½ D tool offset"
- 3D tool radius compensation (Page 281)

For 3D face milling, refer to:

- 3D tool radius compensation (Page 281)

3.4.6 Tool parameters 12 ... 14: Wear - tool lengths

While geometry tool length compensation (tool parameters 3 to 5) is used to define the size of the tool, wear tool length compensation can be used to correct the change in the active tool size.

The active tool variable can change as a result of the following influences:

- Differences between the tool fixture on the tool measurement machine and the tool fixture on the machine tool
- Tool wear caused during service life by machining
- Definition of a finishing allowance

The geometry tool compensation (tool parameters 3 to 5) and the wear tool length compensation (tool parameters 12 to 14) are added together (geometry tool length 1 is added to wear tool length 1, ...) to arrive at the **size of the active tool**.

3.4.7 Tool parameters 15 ... 20: Wear - tool shape

While geometry tool radius compensation (tool parameters 6 to 11) is used to define the shape of the tool, wear tool radius compensation can be used to correct the change in the active tool shape.

The active tool variable can change as a result of the following influences:

- Tool wear caused during service life by machining
- Definition of a finishing allowance

The geometry tool radius compensation (tool parameters 6 to 11) and the wear tool radius compensation (tool parameters 15 to 20) are added together (geometry tool radius 1 is added to wear tool radius 1) to arrive at the **shape of the active tool**.

Special feature for turning tools

For turning tools, the tool parameter \$TC_DP19 is not used as a wear parameter, but to specify the inclined position for inclined turning tools.

More information: → Chapter "Modification of the offset data for rotatable tools (Page 182)".

3.4.8 Tool parameters 21 ... 23: Base dimension / adapter dimension

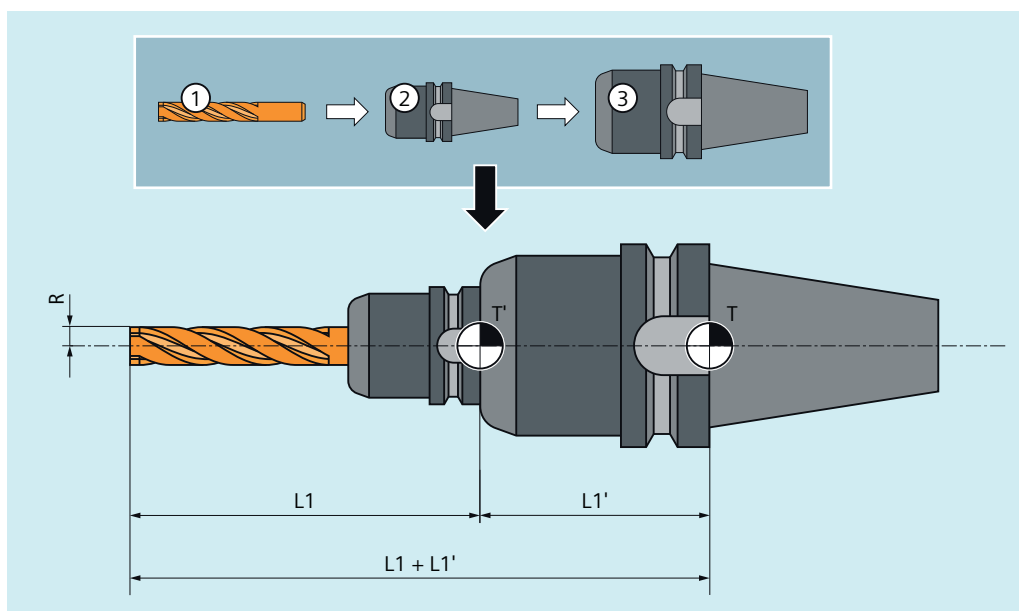
Tool base dimension/adapter dimension can be used when the reference point of the toolholder (tool size) differs from the reference point of the toolholder.

This will occur in the following situations:

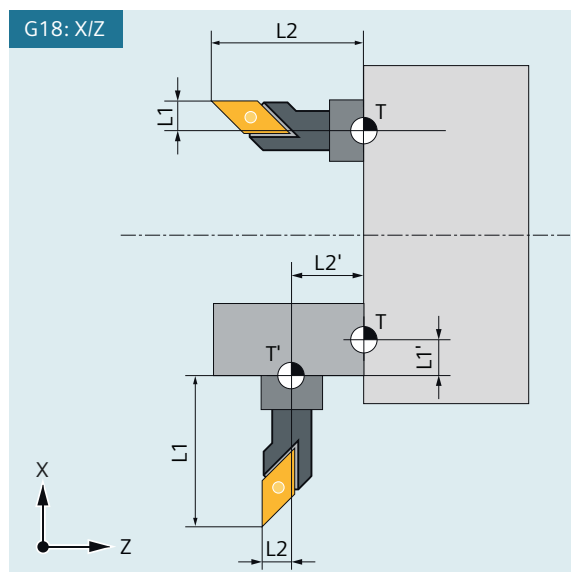
- Tool and tool adapter are measured separately, however, are installed in the machine as one unit (the tool size and adapter size are entered separately in a tool cutting edge).
→ See example 1.
- The tool is inserted in a second tool adapter which is located in a different position (e.g. vertical and horizontal spindle).
- The tool adapters of a tool turret are in different positions.
→ See example 2.

In order that the discrepancy between the toolholder reference point F and the tool carrier reference point T' can be corrected on the three geometry axes (3-dimensional), all 3 basic lengths are always active irrespective of the tool type. In other words, a twist drill (tool type 200) with a tool length compensation (length 1) can also have a base dimension/adapter dimension in 3 axes.

Example 1: Tool and tool adapter are measured separately, however, are installed in the machine as one unit



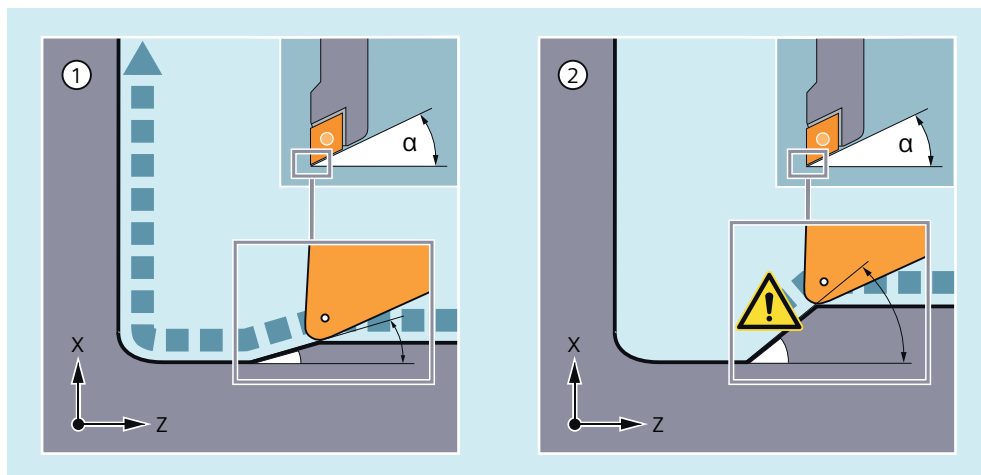
- ① Tool
- ② Tool holder
- ③ Tool adapter
- T Adapter reference point (for inserted tool = tool carrier reference point)
- T' Tool carrier reference point
- L1 Geometry - length 1
- L1' Adapter dimension - length 1
- L1 + L1' Total length L1
- R Radius

Example 2: The tool adapters of a tool turret are in different positions

- T Tool carrier reference point
- T' Tool carrier reference point
- L1 Geometry - length 1
- L2 Geometry - length 2
- L1' Base dimension - length 1
- L2' Base dimension - length 2

3.4.9 Tool parameter 24: Clearance angle

Certain turning cycles, in which traversing motions with relief cuts are generated, monitor the clearance angle of the active tool for possible contour violation.



α Clearance angle

- ① The contour to be machined is not violated.
- ② The contour to be machined would be violated.

The clearance angle is entered in tool parameter 24.

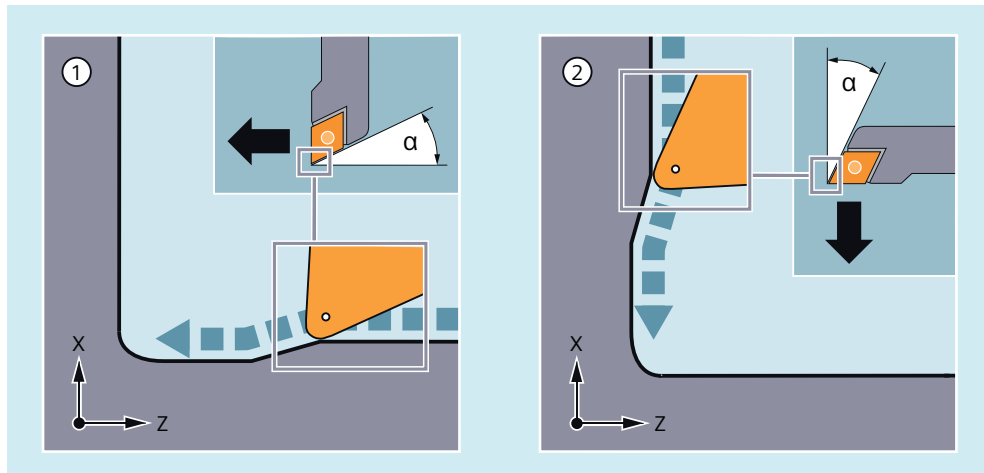
Value range: $0^\circ \dots 90^\circ$ (without sign)

Note

If a clearance angle of zero is entered in tool parameter 24, then relief cuts are not monitored in the turning cycles.

Longitudinal or face machining

The clearance angle is entered differently according to the type of processing. If a tool is to be used for both longitudinal **and** face machining, two tool cutting edges must be entered for different clearance angles.



- α Clearance angle
- ① Longitudinal machining
- ② Face machining

Further information

See the Programming Manual Cycles for detailed information about the clearance angle.

3.5 2D tool radius compensation

3.5.1 Function

3.5.1.1 Calculating the equidistant tool path

Commands of G Group 7 are used to activate and deactivate tool radius compensation in the NC program:

Command	Meaning
G40	No tool radius compensation
G41	Tool radius compensation left of the contour
G42	Tool radius compensation right of the contour

For detailed information on programming G40/G41/G42, see Chapter "Activating/deactivating tool radius compensation (G40, G41, G42, OFFN): (Page 74)".

When tool radius compensation is active, the control automatically computes the equidistant tool paths for the programmed workpiece contour for different tools.

The control requires the following information to do this:

- Tool number (T...)
- Tool offset number D (with cutting edge number)
- Machining direction (G41/G42) (direction of motion to the left or right of the programmed contour)
- Machining plane (G17/G18/G19)

Based on this information, the tool path is compensated in the programmed plane.

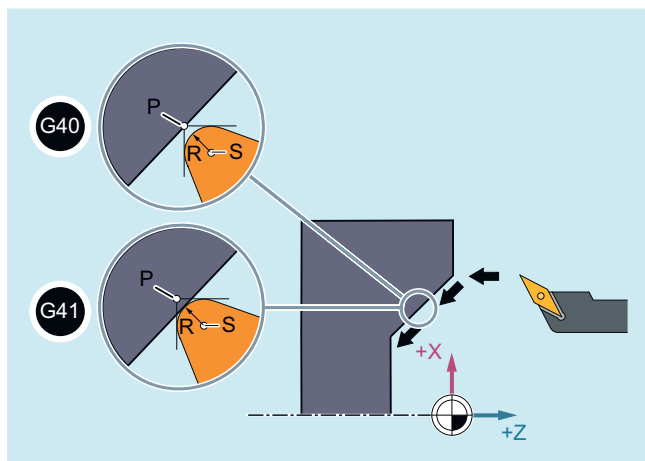
Tool no. (T...), cutting edge no. (D...)

The distance between tool path and workpiece contour is calculated from the milling cutter radii or cutting edge radii and specifications relating to the cutting edge position.

Machining direction (G41/G42)

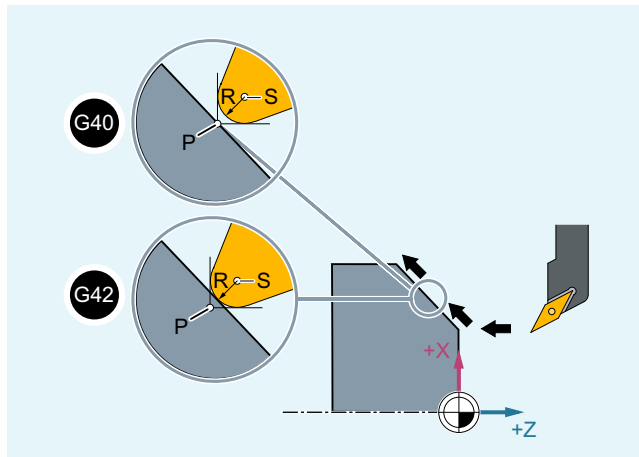
From the machining direction data (G41/G42) the control identifies the direction in which the tool path is to be shifted.

The compensation is possible in the direction of motion to the left or right of the programmed contour.



- P Tool tip
 S Cutting edge center point
 R Cutting edge radius

Figure 3-4 Tool radius compensation when turning with machining direction to the left of the contour (G41)



- P Tool tip
- S Cutting edge center point
- R Cutting edge radius

Figure 3-5 Tool radius compensation when turning with machining direction to the right of the contour (G42)

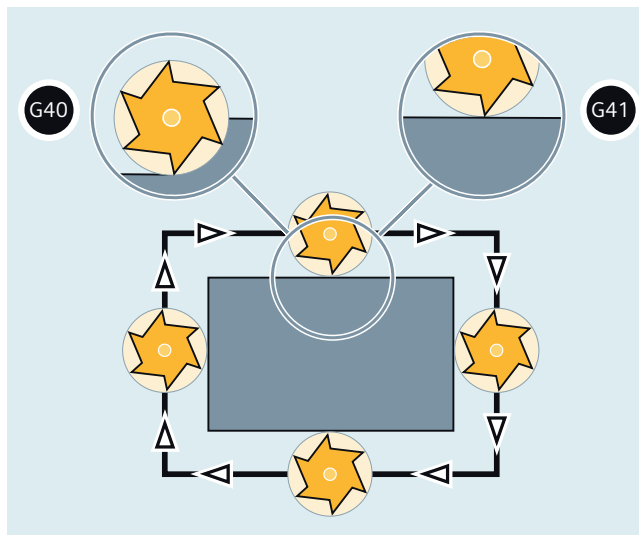


Figure 3-6 Tool radius compensation when milling with machining direction to the left of the contour (G41)

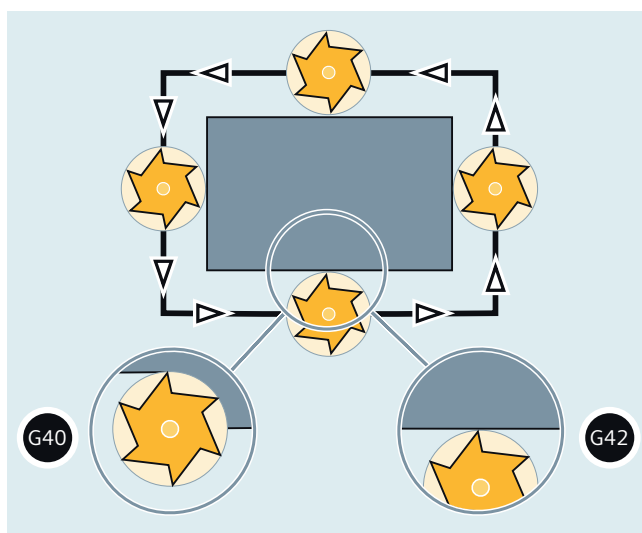
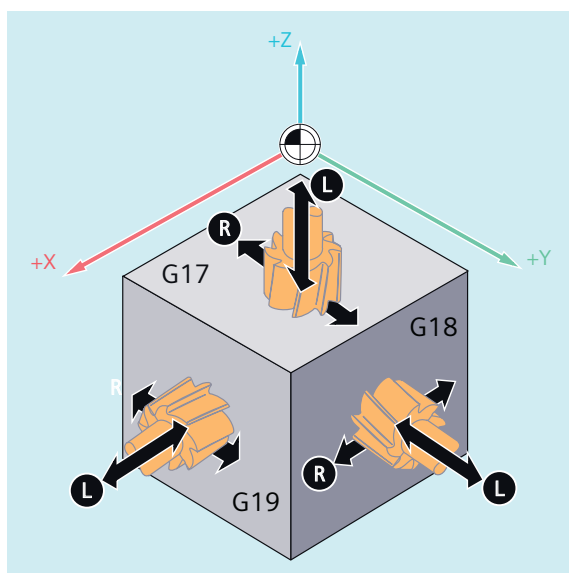


Figure 3-7 Tool radius compensation when milling with machining direction to the right of the contour (G42)

Machining plane

Based on the machining plane information, the control detects the plane and therefore the axis directions in which a compensation is applied.



L Length

R Radius

Example:

Program code	Comment
...	

3.5 2D tool radius compensation

Program code	Comment
N10 G17 G41 ...	; The tool radius compensation is performed in the X/Y plane, the tool length compensation is performed in the Z direction.
...	

Note

On 2-axis machines, tool radius compensation is only possible in "real" planes, usually for G18.

3.5.1.2 Response when approaching and/or retracting

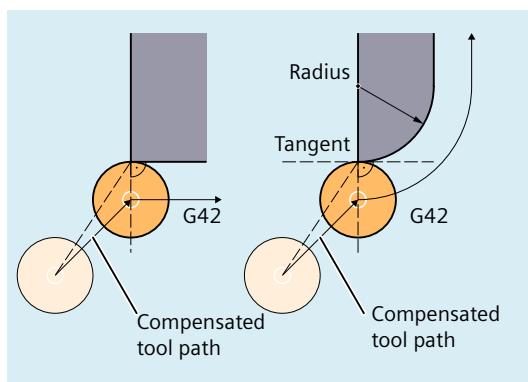
In the NC program, the approach and retract paths when activating and deactivating the tool radius compensation (G40/G41/G42) can be adapted to the required contour or to shapes of blanks (unmachined parts) using the commands of G Group 17.

G command	Meaning
NORM	Normal setting at start point / end point (basic setting)
KONT	Travel around contour at start point / end point
KONTC	Approach/retraction with constant curvature
KONTT	Approach/retraction with constant tangent

For detailed information on programming NORM/KONT/KONTC/KONTT, see Chapter "Adapting the approach/retract response (NORM, KONT, KONTC, KONTT) (Page 81)".

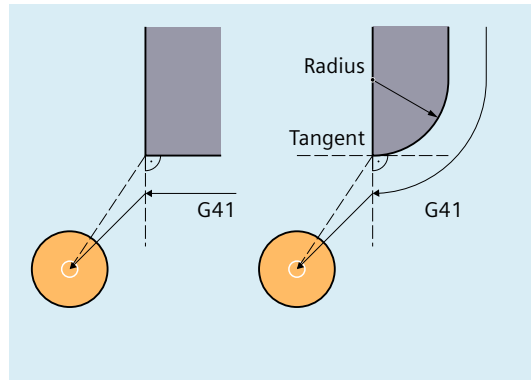
NORM**Approaching with NORM**

With NORM, the tool moves directly to the compensated start position along a straight line (irrespective of the preset approach angle programmed for traversing motion) and is positioned perpendicular to the path tangent at the starting point:

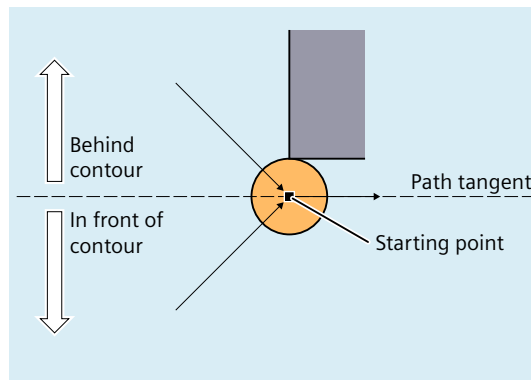


Retracting with NORM

The tool is perpendicular to the last compensated path end point and then moves (irrespective of the preset approach angle programmed for the traversing motion) directly in a straight line to the next non-compensated position, e.g. to the tool change point.

**KONT**

Prior to the approach, the tool can be located **in front of** or **behind** the contour. The path tangent at the starting point serves as a separation line:



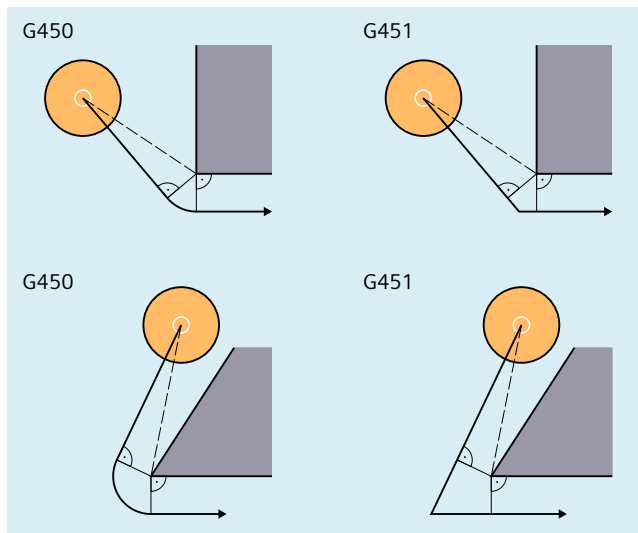
Accordingly, a distinction has to be made between two scenarios where approach/retract with KONT is involved:

1. The tool is located in front of the contour.
→ The approach/retract strategy is the same as for NORM.
2. The tool is located behind the contour.

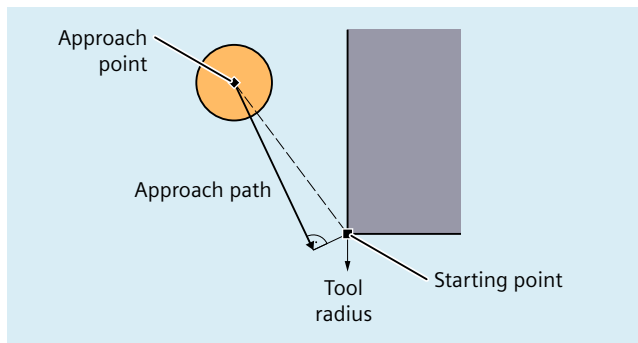
Approach with KONT (the tool is located behind the contour).

The tool travels around the starting point either along a circular path or over the intersection of the equidistants depending on the programmed corner behavior (G450/G451).

The commands G450/G451 apply to the transition from the current block to the next block.



In both cases (G450/G451), the following approach path is generated:



A straight line is drawn from the uncompensated approach point. This line is a tangent to a circle with circle radius = tool radius. The center point of the circle is on the starting point.

Retracting with KONT

The same applies to retract as to approach, but in the inverse sequence.

KONTC / KONTT

Note

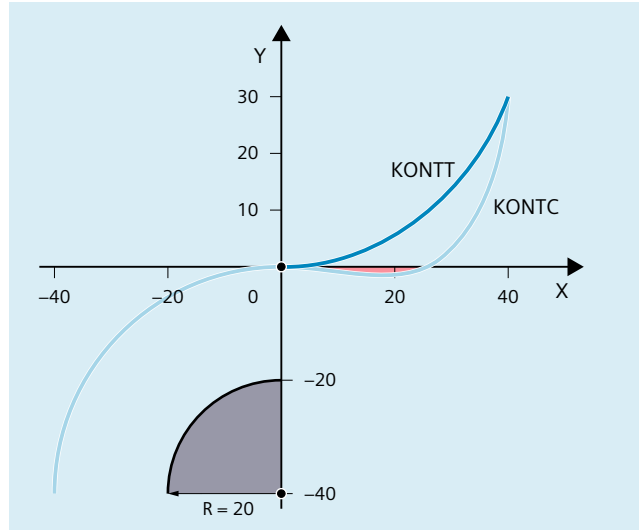
A license is required for option "Polynomial interpolation" in order to use KONTC and KONTT.

KONTC	The contour point is approached/exited with constant curvature. There is no jump in the acceleration at the contour point.
KONTT	The contour point is approached/exited with constant tangent. A jump in the acceleration can occur at the contour point.

Only **linear blocks** are permitted for the original approach and retract blocks with KONTT/ KONTC. These programmed linear blocks are replaced in the control by the corresponding polynomial curves.

The continuity condition is observed in all **three** axes. It is thus possible to program a simultaneous path component perpendicular to the compensation plane for approach/ retraction.

KONTT and KONTC compared



The diagram below shows the differences in approach/retract behavior between KONTT and KONTC. A circle with a radius of 20 mm about the center point at X0 Y40 is compensated with a tool with an external radius of 20 mm. The tool center point therefore moves along a circular path with radius 40 mm. The end point of the approach block is at X40 Y30. The transition between the circular block and the retraction block is at the zero point. Due to the extended continuity of curvature associated with KONTC, the retraction block first executes a movement with a negative Y component. This will often be undesired. This response does not occur with the KONTT retraction block. However, with this block, an acceleration step change occurs at the block transition.

If the KONTT or KONTC block is the approach block rather than the retract block, the contour is exactly the same, but is only machined in the opposite direction.

Note

The diagram shows that a straight line bordering on the contour quadrant, e.g. to X20 Y20, would result in a contour violation for retract/approach to/from X0, Y0 with KONTC.

3.5.1.3 Behavior when traveling around outside corners.

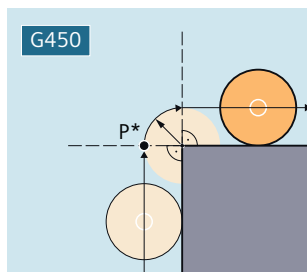
The strategy when traveling around outside workpiece corners when tool radius compensation is activated (G41/G42) can be set in the NC program with commands of G Group 18.

Command	Meaning
G450	The tool travels around outside corners along an arc with the tool radius.
G451	The tool approaches the point of intersection of the two equidistants, which are located at a distance equivalent to the tool radius from the programmed contour.

For detailed information on programming G450/G451 see Chapter "Defining the response when traveling around outside corners (G450, G451, DISC) (Page 85)".

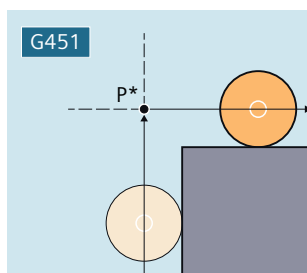
G450

When G450 is active, at outside corners, the center point of the tool traverses a circular path with the tool radius. The circular path starts with the normal position (perpendicular to the path tangent) at the end point of the previous path section (program block) and ends in normal position at the start point of the new path section (program block).



G451

When G451 is active, the position (intersection) resulting from the path lines (straight line, circle or helix) located at a distance of the tool radius to the programmed contour (center-point path of the tool) is approached.



Intersection procedure for polynomials

Function "Intersection procedure for polynomials" is active as default, which means that G451 can also be used at outside corners where polynomials (splines) are involved. This means that in the continuous path mode (G64), velocity dips can be avoided that inevitably occur as a result of G450.

If the function must be deactivated using MD20256 \$MC_CUTCOM_INTERS_POLY_ENABLE for compatibility reasons, at outside corners, where polynomials are involved, G450 is always active independent of what has been programmed.

Overshoot of the G450 circular path (DISC)

No sharp outside contour corners can occur with G450 as the path of the tool center point through the transition circle is controlled so that the cutting edge stops at the outside corner (programmed position). If sharp outside corners are still to be machined with G450, the DISC address in the NC program can be used to program an overshoot. Thus, the transition circle becomes a conic section and the tool cutting edge retracts from the outside corner.

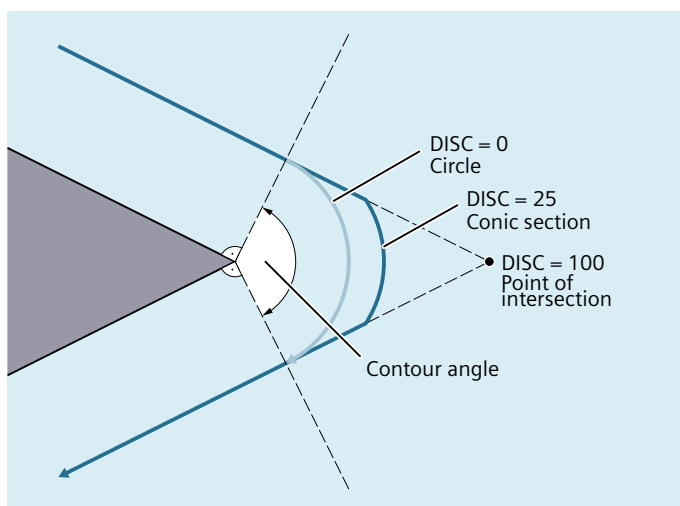


Figure 3-8 Example: Overshoot with DISC=25

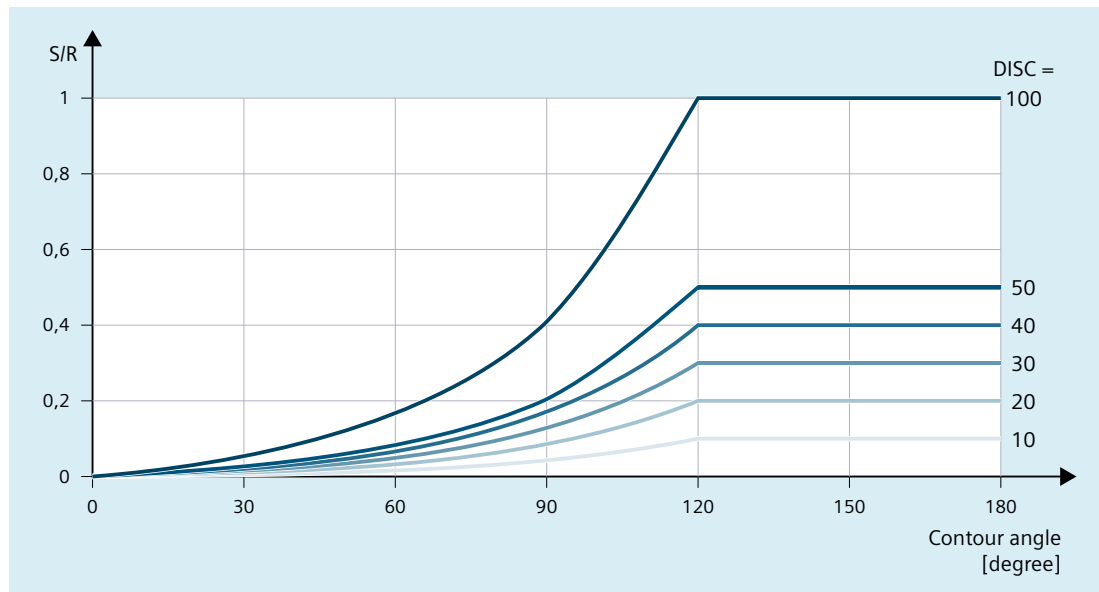
The range of values of the DISC instruction is 0 to 100, in increments of 1.

Value	Meaning
DISC = 0	Overshoot disabled, transition circle active
DISC = 100	Overshoot large enough to theoretically produce a response similar to intersection (G451).

The maximum value for DISC can be set using machine data MD20220 \$MC_CUTCOM_MAX_DISC.

Values greater than 50 are generally not practical with DISC.

3.5 2D tool radius compensation



R Tool radius

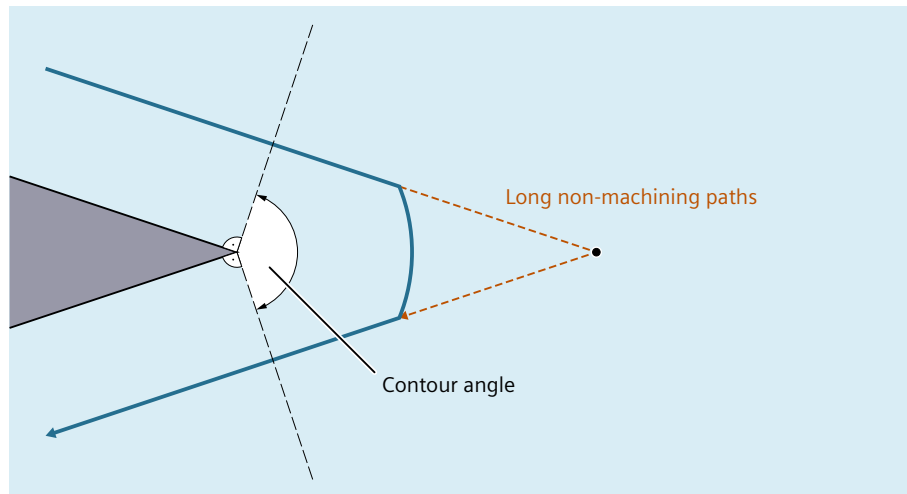
S Traversed superelevation

S/R Normalized superelevation (in relation to tool radius)

Figure 3-9 Overshoot resulting from DISC depending on the contour angle

Automatic switchover G451 → G450 for very sharp outside corners

G451 can result in long idle paths when outside corners are very sharp:



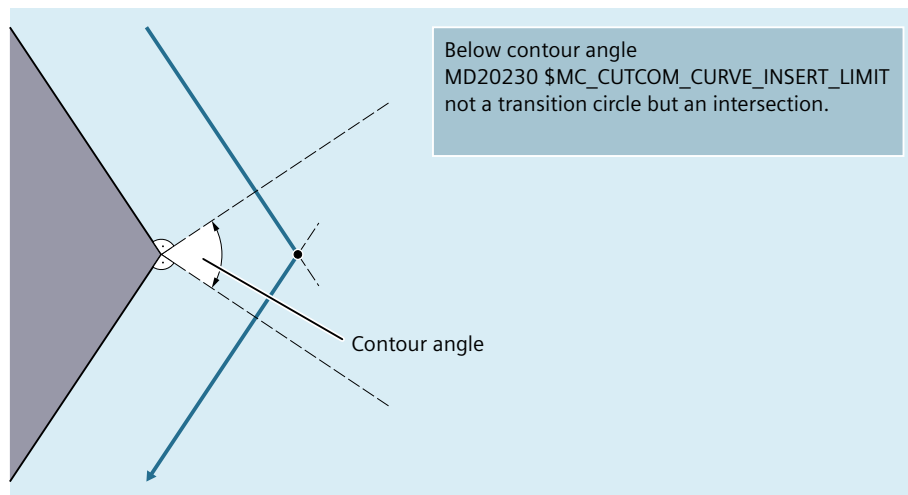
Therefore, the system switches automatically from G451 (intersection) to G450 (transition circle, with DISC where appropriate) when outside corners are very sharp.

The limit (threshold) angle (contour angle) for this automatic switchover (intersection point → transition circle) can be specified in machine data MD20210 $\$MC_CUTCOM_CORNER_LIMIT$.

Automatic switchover G450 → G451 for very flat outside corners

For very flat outside corners, the procedures with G450 (transition circle) and G451 (intersection) become increasingly similar. In this case, it no longer makes sense to insert a transition circle. Especially for 5-axis machining, it is not permissible to insert a transition curve at these outside corners as this would have a negative impact on the speed in the continuous-path mode (G64). Therefore, the system switches automatically from G450 (transition circle, with DISC where appropriate) to G451 (intersection) when outside corners are very flat.

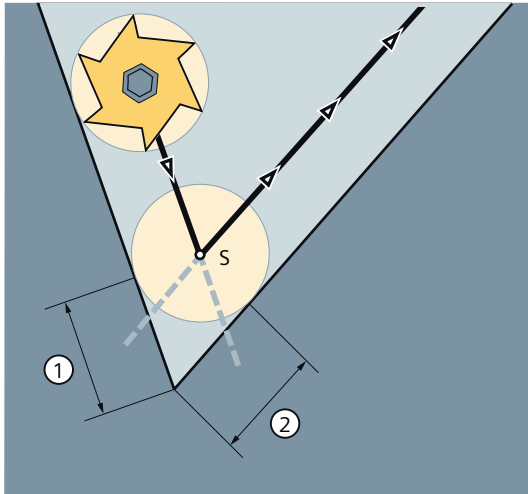
The limit (threshold) angle (contour angle) for this automatic switchover (transition circle → intersection point) can be specified in machine data MD20230 `$MC_CUTCOM_CURVE_INSERT_LIMIT`.



3.5.1.4 Behavior at inside corners

Point of intersection

If two consecutive blocks form an inside corner, an attempt is made to find a point at which the two equidistants intersect. If an intersection is found, the programmed contour is shortened to the intersection:



S: Point of intersection

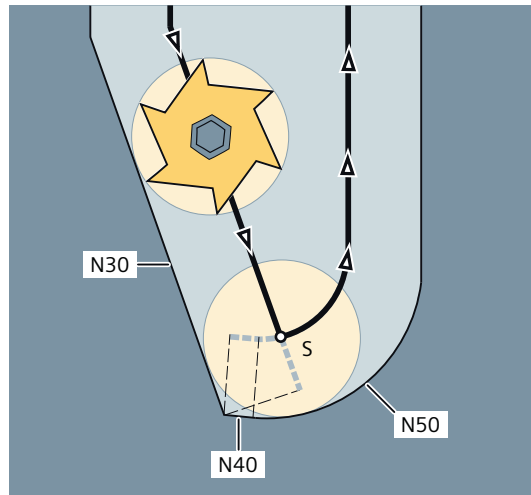
① First block shortened at end.

② Second block shortened at beginning.

Figure 3-10 Programmed contour is shortened

No intersection

For inside corners, it is possible that no intersection is found between two consecutive blocks. In such cases, the control automatically checks the next block and attempts to find an intersection with the equidistants of this block:



S: Point of intersection

Figure 3-11 Predictive contour calculation

This automatic evaluation of the next blocks, the predictive contour calculation, is always performed until the number of blocks defined using machine data MD20240 \$MC_CUTCOM_MAXNUM_CHECK_BLO has been reached.

If no intersection is found within the number of blocks defined for predictive evaluation, program execution is interrupted and an alarm is output.

Multiple intersections

With inside corners it is also possible that predictive contour calculation finds multiple intersections of the equidistants in several consecutive blocks. In this case, the last intersection is always used as the valid intersection. The previous intersection points are not approached:

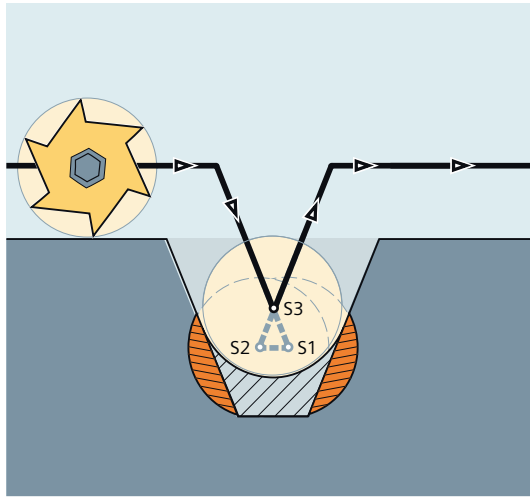


Figure 3-12 In this example, the pocket is machined only as much as is possible without causing a contour violation.

For more information, see also Chapter "Collision monitoring ("bottleneck detection") (Page 88)".

Special features

Where multiple intersections with the next block are found, the intersection nearest the start of the next block applies.

3.5.1.5 Blocks with variable compensation value

A variable compensation value is permissible for all types of interpolation (including circular and spine interpolation).

It is also permitted to change the sign (and, therefore, the compensation side).

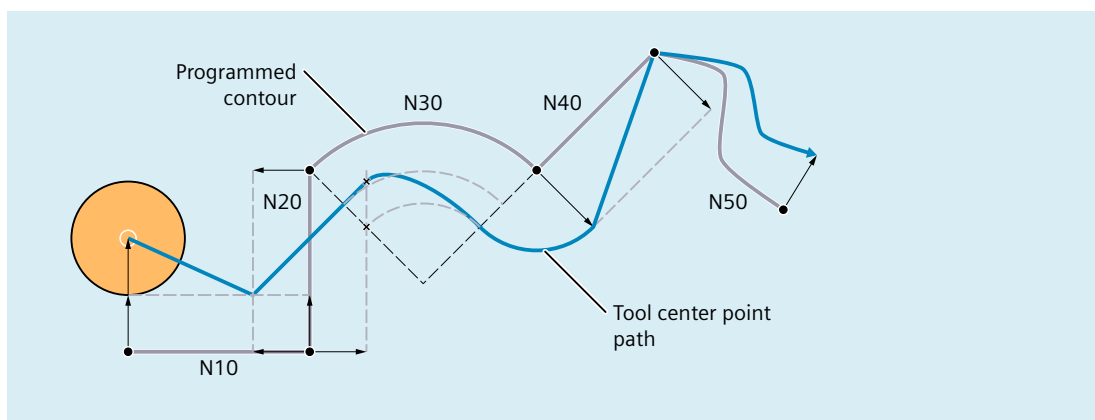


Figure 3-13 Tool radius compensation with variable compensation value

Intersection calculation

When the intersections in blocks with variable compensation value are calculated, the intersection of the offset curves (tool paths) is always calculated based on the assumption that the compensation value is constant.

If the block with the variable compensation value is the first of the two blocks to be assessed in the traversing direction, then the compensation value at the block end is used for the calculation; otherwise, the compensation value at the block start is used.

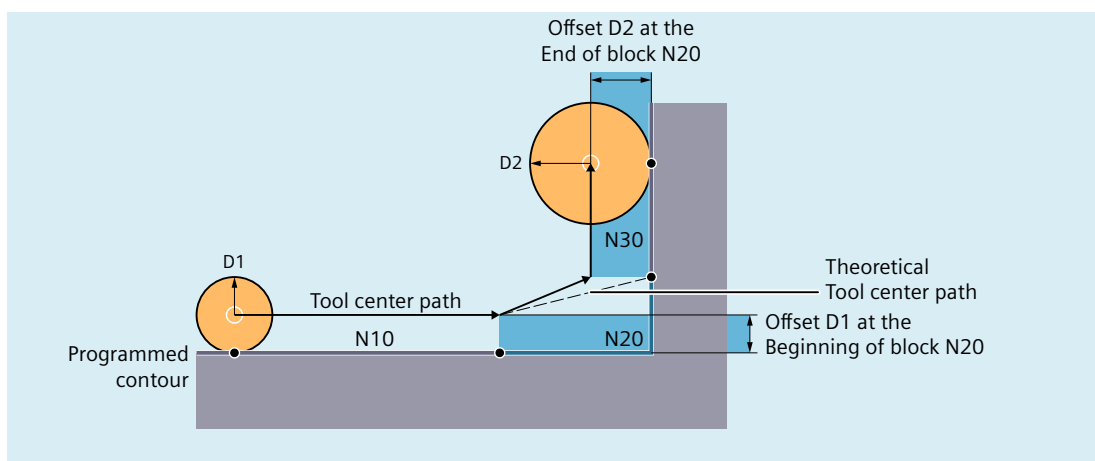


Figure 3-14 Intersection calculation with variable compensation value

Restriction

When machining the inside of the circle, if the compensation radius becomes higher than the programmed circle radius, then machining is rejected with the following alarm:

Alarm 10758 "Curvature radius with variable compensation value too small"

Maintaining the stability of closed contour

If a radius of two circles is increased slightly, a third block may be necessary in order to maintain the stability of the closed contour. This is the case if two adjacent blocks, which represent two possible intersection points for a closed contour, are skipped due to the compensation.

A stable, closed contour can be achieved by selecting the first intersection point instead of the second:

SD42496 \$SC_CUTCOM_CLSD_CONT $\neq$ 0 (response of TRC for a closed contour).

In that case, the second intersection point is always reached, even if the block is skipped. A third block is then not required.

3.5.1.6 Alarm behavior

If a tool radius compensation alarm is output during preprocessing, main-run machining stops at the next block end reached. This usually means at the end of the block currently being interpolated (if Look Ahead is active, once the axes have come to a stop).

Tool radius compensation generally requires at least one of the following traversing blocks (even more for bottlenecks) to determine the end point of a block. Since the preprocessing stop of such a block is not available, traversing continues to the offset point in the last block. Correspondingly, the offset point in the start point is approached in the first block after a preprocessing stop.

The contour obtained may deviate considerably from the one that would result without preprocessing stop. This means that contour violations are possible.

This means that with the default setting (SD42480 \$MC_STOP_CUTCOM_STOPRE = TRUE), for a preprocessing stop while tool radius compensation is active, an alarm is output and the program is no longer processed. The operator can acknowledge this alarm and continue the NC program with NC start or cancel it with reset.

3.5.2 Parameterization

Machine data

Maximum value for DISC

The G450 transition circle does not produce sharp outside contour corners as the path of the tool center point through the transition circle is controlled so that the cutting edge stops at the outside corner (programmed position). If sharp outside corners are still to be machined with G450, the DISC instruction in the NC program can be used to program an overshoot. Thus, the transition circle becomes a conic section and the tool cutting edge retracts from the outside corner.

The programmable maximum value for DISC can be set using machine data:

MD20220 \$MC_CUTCOM_MAX_DISC (max. value for DISC).

Values greater than 50 are generally not practical with DISC.

Limit (threshold) angle for automatic switchover for very sharp outside corners

G451 can result in long idle paths when outside corners are very sharp. Therefore, the system switches automatically from G451 (intersection) to G450 (transition circle, with DISC where appropriate) when outside corners are very sharp.

The limit (threshold) angle (contour angle) for this automatic switchover (intersection point → transition circle) can be specified in machine data:

MD20210 \$MC_CUTCOM_CORNER_LIMIT (Max. angle for compensation blocks with tool radius compensation).

Limit (threshold) angle for automatic switchover for very flat outside corners

For very flat outside corners, the procedures with G450 (transition circle) and G451 (intersection) become increasingly similar. In this case, it no longer makes sense to insert a transition circle. Especially for 5-axis machining, it is not permissible to insert a transition curve at these outside corners as this would have a negative impact on the speed in the continuous-path mode (G64). Therefore, the system switches automatically from G450 (transition circle, with DISC where appropriate) to G451 (intersection) when outside corners are very flat.

The limit (threshold) angle (contour angle) for this automatic switchover (transition circle → intersection point) can be specified in machine data:

MD20230 \$MC_CUTCOM_CURVE_INSERT_LIMIT (Max. angle for intersection calculation with tool radius compensation).

Maximum number of blocks for predictive contour calculation

For inside corners, it is possible that no intersection is found between two consecutive blocks. In such cases, the control automatically checks the next block and attempts to find an intersection with the equidistants of this block:

This automatic evaluation of the next blocks, the predictive contour calculation, is always performed until the number of blocks defined using machine data has been reached:

MD20240 \$MC_CUTCOM_MAXNUM_CHECK_BLOCKS (blocks for predictive contour calculation for TRC)

If no intersection is found within the number of blocks defined for predictive evaluation, program execution is interrupted and an alarm is output.

Maximum number of dummy blocks

Generally, only program blocks with positions on geometry axes in the current plane are programmed when tool radius compensation (TRC) is active. However, dummy blocks can still also be programmed when TRC is active. Dummy blocks are program blocks, which do not contain any positions on a geometry axis in the current plane:

- Positions on the infeed axis
- Auxiliary functions
- etc.

The maximum number of dummy blocks can be defined in machine data:

MD20250 \$MC_CUTCOM_MAXNUM_DUMMY_BLOCKS (number of blocks with no traversing movements for TRC).

Intersection procedure for polynomials

The response of tool radius compensation at outside corners, where polynomials (splines) are involved, is defined using machine data:

MD20256 \$MC_CUTCOM_INTERS_POLY_ENABLE (intersection procedure for polynomials is possible)

Value	Meaning
FALSE	At outside corners where polynomials are involved, G450 is always active and a block is inserted (can be very short) even if transitions are almost tangential. These short blocks always result in unwanted speed dips in the continuous path mode (G64).
TRUE (default)	The intersection procedure (G450) can also be used at outside corners where polynomials are involved.

Setting data**Alarm behavior with preprocessing stop**

For a preprocessing stop while tool radius compensation is active, the alarm response is defined with the setting data:

SD42480 \$MC_STOP_CUTCOM_STOPRE

Value	Meaning
FALSE	No alarm is output. Program is not stopped.
TRUE (default)	Alarm output Program is stopped.

Retraction behavior with preprocessing stop

The retraction response for a preprocessing stop before deselecting tool radius compensation is set with the setting data:

SD42490 \$SC_CUTCOM_G40_STOPRE

Value	Meaning
FALSE (default)	If a preprocessing stop (programmed or generated internally by the control) is located before the deselection block (G40) when tool radius compensation is active, the starting point of the deselection block is first approached starting from the last end point before the preprocessing stop. Subsequently, the deselection block itself is processed, i.e. two traversing blocks are usually created from the deselection block. In these blocks, tool radius compensation is no longer active.
TRUE	In the same situation as described above, the intersection positioned on the first partial contour closer to the block start, is selected.

Behavior with closed contour

The tool radius compensation behavior with closed contour can be set via the setting date:

SD42496 \$SC_CUTCOM_CLSD_CONT

Value	Meaning	
FALSE	Case A:	If two intersections appear on the inside when offsetting an (almost) closed contour, which consists of two successive circle blocks or one circle block and one linear block, the intersection positioned closer to the start of the block on the first partial contour is selected in accordance with the standard procedure. This method can be used to omit contours. A contour is considered to be (almost) closed if the distance between the starting point of the first block and the end point of the second block is less than 10% of the effective compensation radius, but not more than 1000 path increments (corresponds to 1 mm for 3 decimal places).
	Case B:	If, during correction on the inside and a block transition between a straight line and a circle with an opening angle of more than about 315 degrees, the offset curve of the straight line intersects the offset curve of the circle at two points, the intersection point that is closer to the start of the block on the first subcontour is selected in accordance with the standard procedure. In this situation, the two curves involved are almost completely omitted.
TRUE (default)	In the same situation as described above, the intersection positioned on the first partial contour closer to the block end, is selected. This preserves more of the two subcontours. This is the better method to achieve the programmed original contour.	
	Case A:	For two consecutive curves (as described above) whose offset curves intersect twice, the intersection point closer to the end of the first curve is selected. A slight contour violation is accepted at the beginning of the first curve and at the end of the second curve in order to be able to process both curves completely. However, this is only the case if the contour is (almost) closed.
	Case B:	In the same situation as described above, the intersection point closer to the end of the first curve is selected. However, this is only the case if the amount of the resulting contour violation is less than the fixed tolerance of 0.002 mm.

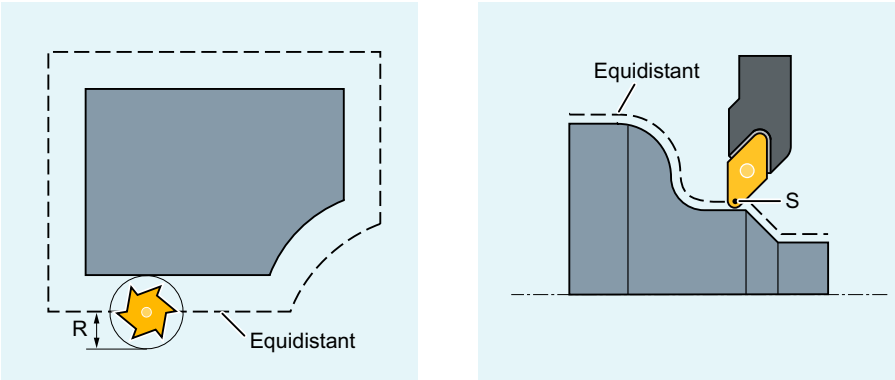
Note

For (almost) closed contours, it is generally always better to divide contours into subcontours (e.g. program two semicircles instead of one full circle) so that the tool radius compensation can find the unique intersection point.

3.5.3 Programming

3.5.3.1 Activating/deactivating tool radius compensation (G40, G41, G42, OFFN):

When tool radius compensation (TRC) is active, the control automatically calculates the equidistant tool paths to the programmed workpiece contour for various tools.



- R Tool radius
- S Cutting edge center point

Commands of G Group 7 are used to activate and deactivate tool radius compensation.

Syntax

G0/G1 X... Y... Z... G41/G42 [OFFN=<value>]	
...	
G40 X... Y... Z...	

Meaning

G41	Activate TRC with machining direction left of the contour.
G42	Activate TRC with machining direction right of the contour.
OFFN=<value>:	Allowance on the programmed contour (normal contour offset) (optional), e.g. to generate equidistant paths for rough finishing.
G40	Deactivate TRC.

Note

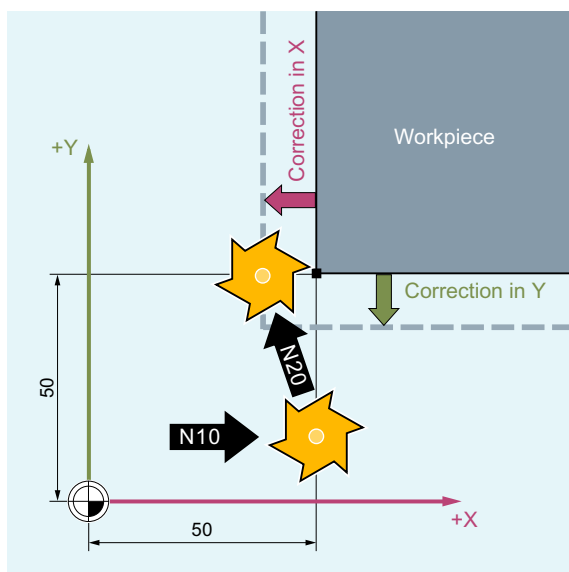
In the NC block with G40/G41/G42, G0 or G1 must be active and at least one axis on the selected machining plane has to be specified.

If only one axis is specified on activation, the last position on the second axis is added automatically and traversed with **both** axes.

The two axes must be active as geometry axes in the channel. This can be ensured by programming GEOAX.

If no geometry axis is programmed for the current plane in the block with the tool radius compensation selection, then no selection is made.

If a geometry axis is programmed in the block with the tool radius compensation deselection, then compensation is deselected even if it is not on the current plane.

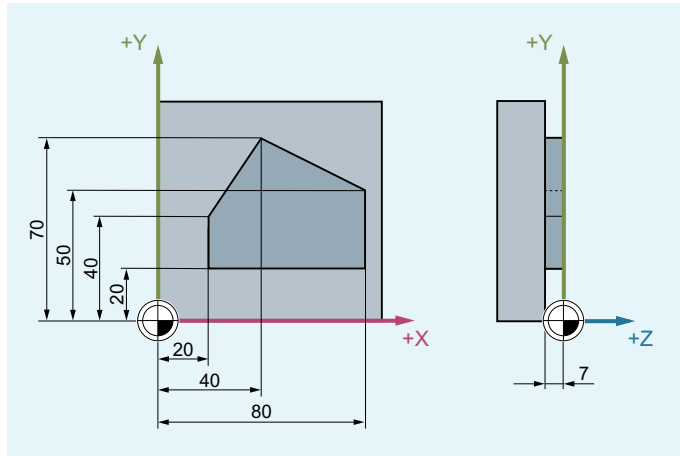
Examples**Example 1: Milling**

Program code	Comment
N10 G0 X50 T1 D1	; Only tool length compensation is activated. X50 is approached without compensation.
N20 G1 G41 Y50 F200	; Radius compensation is activated, point X50/Y50 is approached with compensation.
N30 Y100	
...	

Example 2: "Conventional" procedure using milling as an example

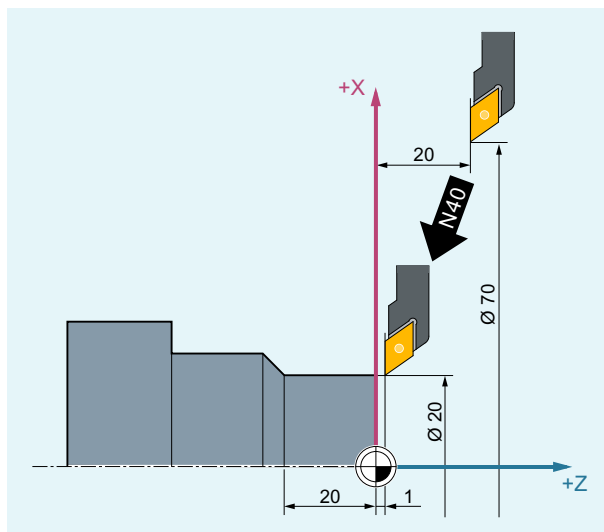
"Conventional" procedure:

1. Tool call.
2. Change tool.
3. Activate machining plane and tool radius compensation.



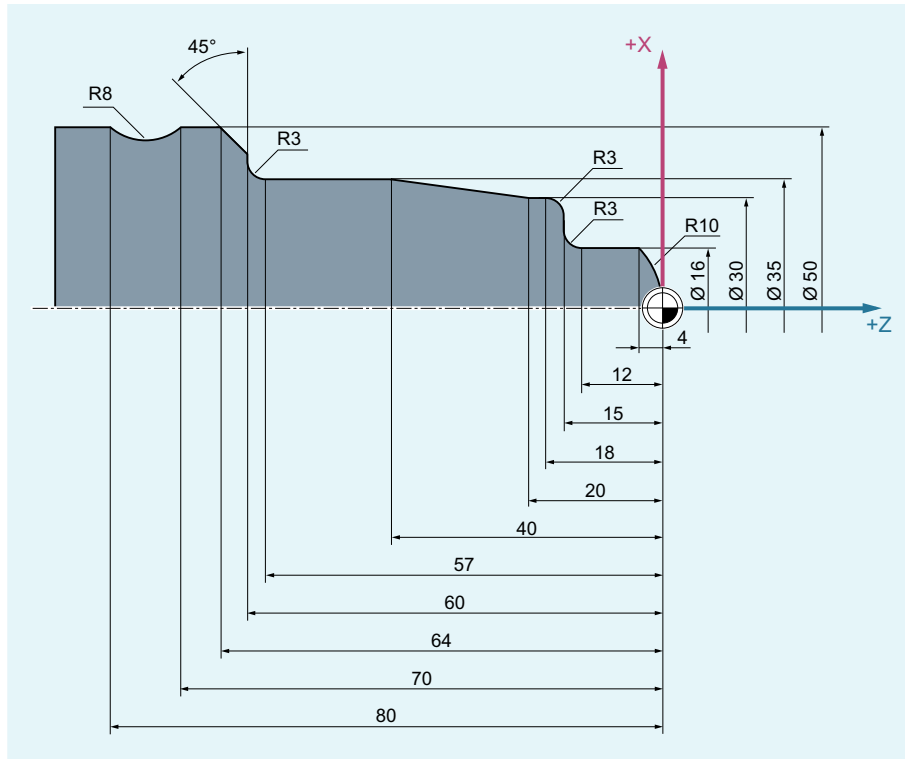
Program code	Comment
N10 G0 Z100	; Retraction for tool change.
N20 G17 T1 M6	; Tool change
N30 G0 X0 Y0 Z1 M3 S300 D1	; Call tool offset values, select length compensation.
N40 Z-7 F500	; Feed in tool.
N50 G41 X20 Y20	; Activate tool radius compensation, tool machines to the left of the contour.
N60 Y40	; Mill contour.
N70 X40 Y70	
N80 X80 Y50	
N90 Y20	
N100 X20	
N110 G40 G0 Z100 M30	; Retract tool, end of program.

Example 3: Turning



Program code	Comment
...	
N20 T1 D1	; Only tool length compensation is activated.
N30 G0 X70 Z20	; X70 Z20 is approached without compensation.
N40 G42 X20 Z1	; Radius compensation is activated, point X20/Z1 is approached with compensation.
N50 G1 Z-20 F0.2	
...	

Example 4: Turning



Program code	Comment
N5 G0 G53 X280 Z380 D0	; Starting point.
N10 TRANS X0 Z250	; Work offset.
N15 LIMS=4000	; Speed limitation (G96).
N20 G96 S250 M3	; Select constant feedrate
N25 G90 T1 D1 M8	; Select tool selection and offset.
N30 G0 G42 X-1.5 Z1	; Set tool with tool radius compensation.
N35 G1 X0 Z0 F0.25	
N40 G3 X16 Z-4 I0 K-10	; Turn radius 10.
N45 G1 Z-12	
N50 G2 X22 Z-15 CR=3	; Turn radius 3.
N55 G1 X24	
N60 G3 X30 Z-18 I0 K-3	; Turn radius 3.
N65 G1 Z-20	
N70 X35 Z-40	
N75 Z-57	
N80 G2 X41 Z-60 CR=3	; Turn radius 3.
N85 G1 X46	
N90 X52 Z-63	
N95 G0 G40 G97 X100 Z50 M9	; Deselect tool radius compensation and approach tool change location.
N100 T2 D2	; Call tool and select offset.

Program code	Comment
N105 G96 S210 M3	; Select constant cutting rate.
N110 G0 G42 X50 Z-60 M8	; Set tool with tool radius compensation.
N115 G1 Z-70 F0.12	; Turn diameter 50.
N120 G2 X50 Z-80 I6.245 K-5	; Turn radius 8.
N125 G0 G40 X100 Z50 M9	; Retract tool and deselect tool radius compensation.
N130 G0 G53 X280 Z380 D0 M5	; Approach tool change location.
N135 M30	; End of program.

More information

Calculation of the tool paths

The control requires the following information in order to calculate the tool paths:

Information	Meaning
Tool no. (T...), cutting edge no. (D...)	To calculate the distance between the tool path and the workpiece contour.
Machining direction (G41/G42)	To determine the direction in which the tool path should be shifted.
Machining plane (G17/G18/G19)	To determine the plane and therefore the axis directions in which compensation should be applied.

This is the reason that a tool must be loaded (T function) and the tool cutting edge/tool compensation (D1 to D9) activated no later than in the program block with the tool radius compensation selection.

Negative compensation value

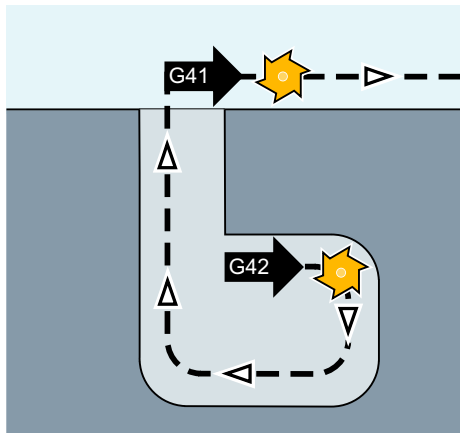
A negative compensation value has the same significance as a change of offset side (G41 ↔ G42).

Tool length compensation

The wear parameter assigned to the diameter axis on tool selection can be defined as the diameter value using an MD. This assignment is not automatically altered when the plane is subsequently changed. To do this, the tool must be selected again after the plane change.

Change in compensation direction (G41 ↔ G42)

A change in compensation direction (G41 ↔ G42) can be programmed without an intermediate G40.



Change of the machining plane

It is **not** possible to change the machining plane (G17/G18/G19) when G41/G42 is active.

Change of tool offset data block (D...)

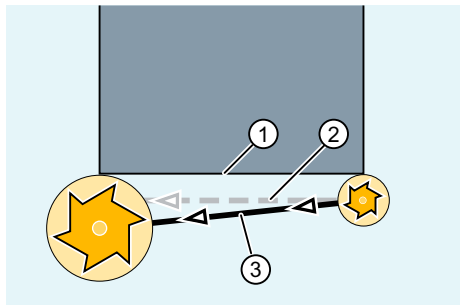
The tool offset data block can be changed in compensation mode.

A changed tool radius already becomes active as from the block containing the new D number.

Note

The radius change or compensation movement is performed across the entire block and only reaches the new equidistance at the programmed end point.

In the case of linear movements, the tool travels along an inclined path between the starting point and the end point:



- ① Programmed contour
- ② Tool path without changing the tool offset data block in the active block
- ③ Tool path when the tool offset data block in the active block changes and the tool radius changes as a result

Circular interpolation produces spiral movements.

Changing the tool radius

The change can be made, e.g. using system variables. The sequence is the same as when changing the tool offset data block (D...).

Note

The modified values only take effect the next time T or D is programmed. The change does not apply until the next block.

Compensation mode

Compensation mode may only be interrupted by a certain number of consecutive blocks or M functions which do not contain traversing commands or positional data in the compensation plane.

Note

The maximum number of consecutive interruption blocks or M commands can be set using machine data.

A block with a path distance of zero also counts as an interruption.

Setting data

The response of 2D tool radius compensation in certain machining situations can be set using the following setting data:

- SD42490 \$SC_CUTCOM_G40_STOPRE (retraction response for a preprocessing stop before deselecting tool radius compensation)
- SD42496 \$SC_CUTCOM_CLSD_CONT (tool radius compensation behavior with closed contour)

For details, see Parameter Manual Machine Data and Parameters.

3.5.3.2 Adapting the approach/retract response (NORM, KONT, KONTC, KONTT)

If tool radius compensation is active (G41/G42), G Group 17 commands (NORM, KONT, KONTC or KONTT) can be used to adapt the approach and retract paths of the tool to the required contour profile or the shape of the blank (unmachined part).

KONTC or KONTT ensure that continuity conditions are maintained in all three axes. It is, therefore, permissible to program a path component perpendicular to the offset plane simultaneously.

Note

A license is required for option "Polynomial interpolation" in order to use KONTC and KONTT.

3.5 2D tool radius compensation

Syntax

G41/G42 NORM/KONT/KONTC/KONTT X... Y... Z...	
...	
G40 X... Y... Z...	

Meaning

NORM	Activate direct approach/retraction to/from a straight line. The tool is oriented perpendicular to the contour point.
KONT	Activate approach/retract with travel around the starting/end point The tool travels around the starting point either along a circular path or over the intersection of the equidistants depending on the programmed corner behavior (G450/G451).
KONTC	Activate approach/retraction with constant curvature. The contour point is approached/exited with constant curvature. There is no jump in the acceleration at the contour point.
KONTT	Activate approach/retraction with constant tangent. The contour point is approached/exited with constant tangent. A jump in the acceleration can occur at the contour point.

Note

Only G1 blocks are permissible as original approach/retract blocks for KONTC and KONTT. The control replaces these with polynomials for the appropriate approach/retract path.

Constraints

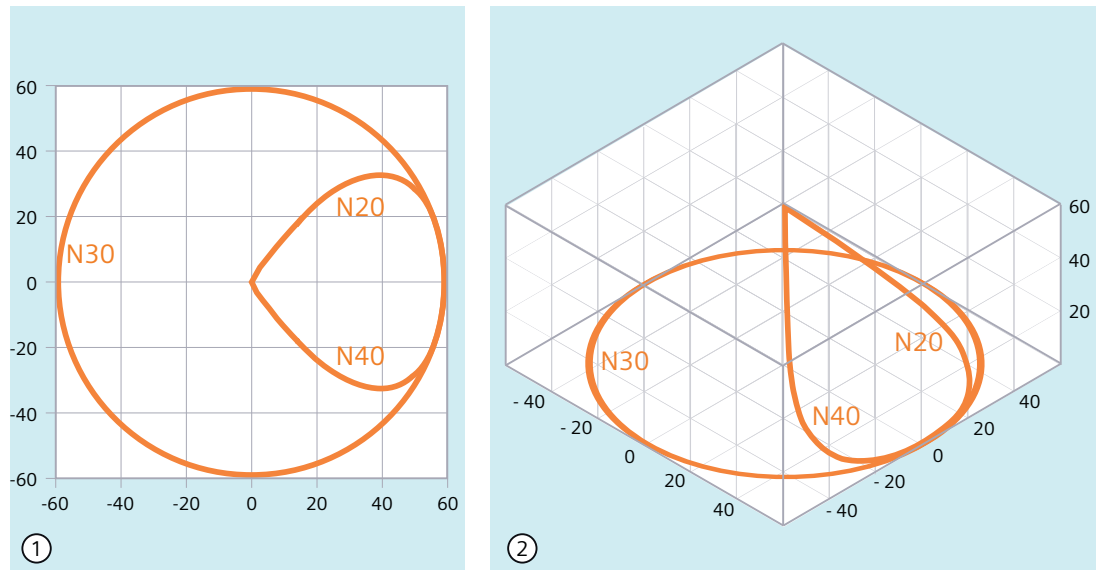
KONTT and KONTC are not available in 3D variants of tool radius compensation (CUT3DC, CUT3DCC, CUT3DF). If they are programmed, the control switches internally to NORM without an error message.

Example

In the following program example, a full circle with a radius of 70 mm is machined in the X/Y plane. The tool approaches/retracts with KONTC:

Program code	Comment
\$TC_DP1[1,1]=121	; Milling tool
\$TC_DP6[1,1]=10	; Radius 10 mm
N10 G1 X0 Y0 Z60 G64 T1 D1 F10000	
N20 G41 KONTC X70 Y0 Z0	; Approach
N30 G2 I-70	; Full circle
N40 G40 G1 X0 Y0 Z60	; Retract
N50 M30	

As the tool has a radius of 10 mm, the resulting tool center point path describes a circle with a radius of 60 mm. Start and end point are at X0 Y0 Z60. When approaching the full-circle with KONTC (N20), the curvature is adapted to the circular path of the full circle. At the same time, the axis traverses from Z60 to the plane of circle Z0. The axis retracts (N40) in the same fashion.



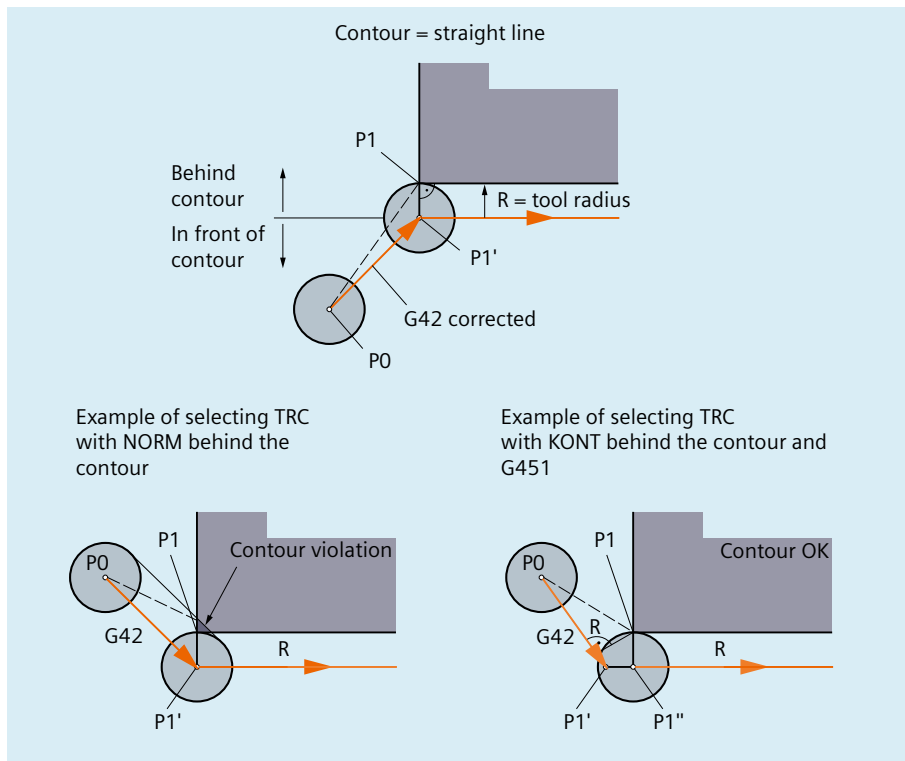
- ① Perpendicular projection
- ② Spatial representation

Figure 3-15 Tool path

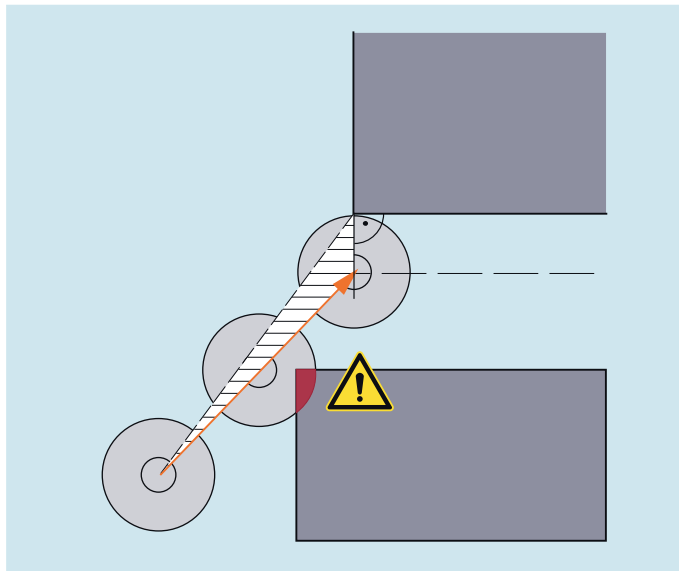
More information

Comparison of NORM and KONT

KONT only differs from NORM when the tool start position is behind the contour:



Modified approach/retract angle



NOTICE

Risk of collision

Modified approach/retract angles as a result of the tool radius compensation must be taken into account during programming in order that collisions are avoided.

3.5.3.3 Defining the response when traveling around outside corners (G450, G451, DISC)

When tool radius compensation (G41/G42) is active, the compensated tool path when traveling around outside corners can be defined using commands of G Group 18 (G450/G451).

Note

G450/G451 is also used to define the approach path when KONT is active and the approach point behind the contour (see "Adapting the approach/retract response (NORM, KONT, KONTC, KONTT) (Page 81)").

Syntax

G450 [DISC=<value>]

G451

Meaning

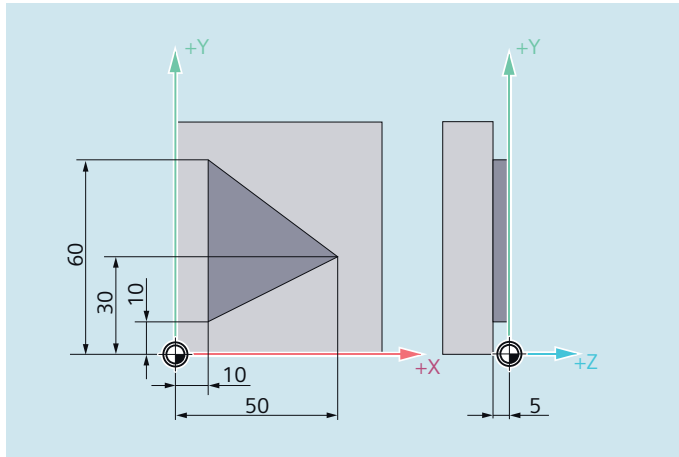
G450	Activating travel around with transition circle With G450, the tool center point travels around the workpiece outside corner along an arc with the tool radius.			
DISC	Flexible programming of the circular path with G450 (optional) No sharp outside contour corners can occur with G450 as the path of the tool center point through the transition circle is controlled so that the cutting edge stops at the outside corner (programmed position). If sharp outside corners are still to be machined with G450, the DISC instruction in the NC program can be used to program an overshoot. Thus, the transition circle becomes a conic section and the tool cutting edge retracts from the outside corner.			
	<value>	Type:	INT	
		Range of values:	0, 1, 2, ... 100	
		Meaning:	0	Transition circle
			100	Intersection of the equidistants (theoretical value)
G451	Activate travel around with intersection point of the equidistants With G451, the tool center point approaches the point of intersection of the two equidistants, which are located at a distance equivalent to the tool radius from the programmed contour. The tool backs off from the workpiece corner. G451 only applies to straight lines and circles.			

Note

DISC only applies with call of G450; however, it can be programmed in a previous block without G450. Both commands are modal.

Example

In the following example, a transition radius is programmed for all outside corners (corresponding to the programming of the corner behavior in block N30). This prevents the tool stopping and backing off at the change of direction.



Program code	Comment
N10 G17 T1 G0 X35 Y0 Z0 F500	; Starting conditions.
N20 G1 Z-5	; Feed in tool.
N30 G41 KONT G450 X10 Y10	; Activate TRC with KONT approach/retract mode and corner behavior G450 .
N40 Y60	; Mill the contour.
N50 X50 Y30	
N60 X10 Y10	
N80 G40 X-20 Y50	; Deactivate compensation mode, retraction on transition circle.
N90 G0 Y100	
N100 X200 M30	

More information

G450/G451

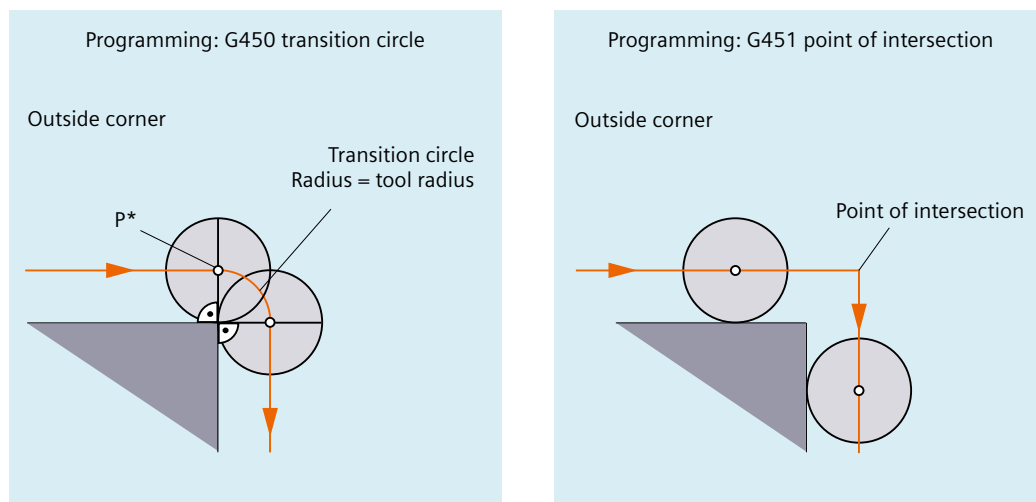


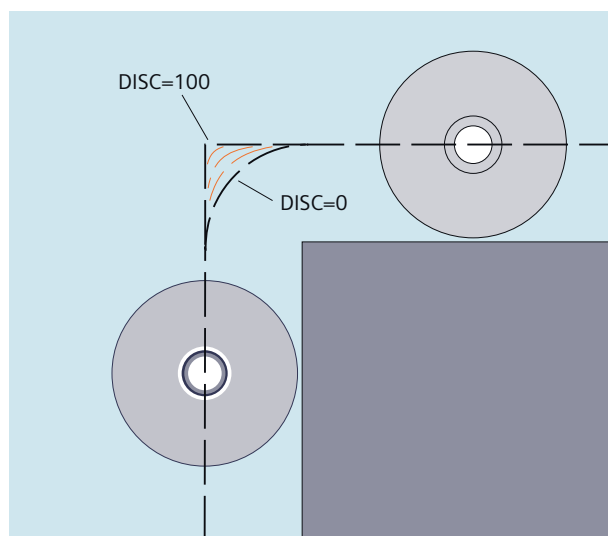
Figure 3-16 Traveling around a 90° outside corner with G450 or G451

At intermediate point P*, the control executes operations such as infeed movements or switching functions. These operations are programmed in blocks inserted between the two blocks forming the corner.

With respect to the data, for G450 the transition circle belongs to the next traversing command.

DISC

When DISC values greater than 0 are specified, transition circles are shown with an increased height – the result is transition ellipses or parabolas or hyperbolas.



3.6 Collision monitoring ("bottleneck detection")

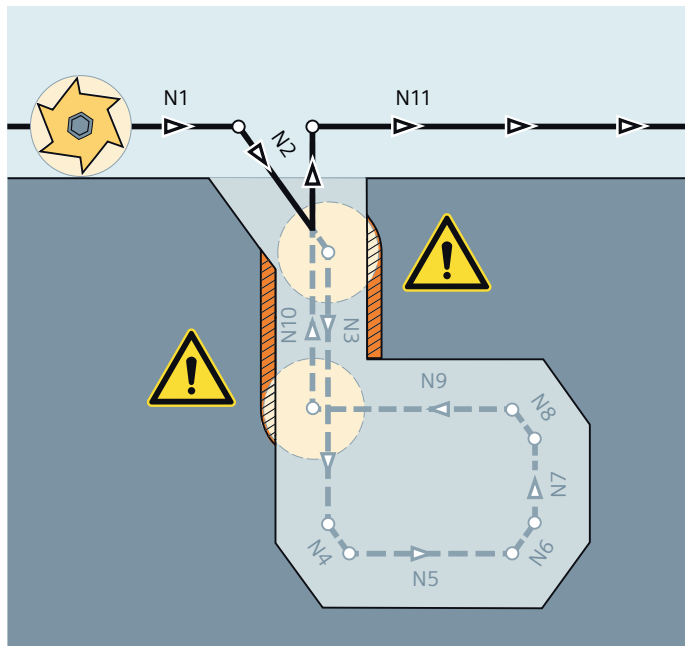
3.6.1 Function

If tool radius compensation is active, collision monitoring ("bottleneck detection") checks by predictive contour calculation across a certain number of blocks whether the equidistant paths of **non-consecutive** blocks intersect. This look-ahead function allows possible collisions to be detected in advance and permits the control to actively avoid them.

The maximum number of blocks that are looked at in advance can be set by channel-specific machine data (see "Parameterization (Page 89)").

If an intersection is detected, the motions programmed between these blocks on the compensation plane are not executed. Alarm 10763 "The path component of the block in the compensation plane will become zero" is displayed. This alarm does not interrupt the NC program, it is only displayed as a warning. The bottleneck is bypassed. All other motions and executable instructions (M commands, traversal of positioning axes, etc.) contained in the omitted blocks are executed at the position of the last intersection found in the sequence in which they are programmed in the NC program.

Example:



When collision monitoring is active with a predictive contour calculation ≥ 8 blocks, blocks N3 to N10 are not executed, only alarm 10763 "The path component of the block in the compensation plane will become zero" is displayed. The control system only runs through blocks N1, N2 and N11.

Activation / deactivation

The function is activated / deactivated in the NC program with commands of G Group 23.

See "Programming (Page 89)".

3.6.2 Parameterization

Maximum number of blocks for predictive contour calculation

The maximum number of blocks to be predictively checked is set in:

MD20240 \$MC_CUTCOM_MAXNUM_CHECK_BLOCKS (blocks for predictive contour calculation for TRC)

Alarm 10763 "The path component of the block in the compensation plane will become zero"

Suppressing an alarm

The display of alarm 10763 "The path component of the block in the correction plane will become zero" (Page 88) can be suppressed by the following setting:

MD11410 \$MN_SUPPRES_ALARM_MASK (mask for support of special alarm outputs), bit 1 = 1

Outputting an alarm and stopping machining

Machining can be stopped by the following setting when alarm 10763 "The path component of the block in the correction plane will become zero" is output:

MD11411 \$MN_ENABLE_ALARM_MASK (mask for activation of warnings), bit 16 = 1

Stopping the machining has the advantage that the user must actively confirm the skipping of blocks by NC Start and thus knows that parts of the machining process may not be processed due to a detected risk of collision.

3.6.3 Programming

Collision monitoring ("bottleneck detection") with active tool radius compensation is activated or deactivated in the NC program with the commands of G group 23.

Syntax

```
G41/G42 CDON
...
CDOF/CDOF2
```

Meaning

CDON:	Activating collision detection ("bottleneck detection") With CDON it is possible to check over an adjustable (MD20240 (Page 89)) number of blocks whether the tool paths of non-adjacent or non-sequential blocks intersect. This look-ahead function allows possible collisions to be detected in advance and permits the control to actively avoid them.
CDOF:	Deactivating collision detection ("bottleneck detection") With CDOF, a search is made in the previous traversing block (at inside corners) for a common intersection for the current block; if necessary the search is extended to even earlier blocks. If an intersection is found, no further blocks are examined. With outside corners, an intersection is always found between two consecutive blocks. Note: CDOF can be used to avoid the faulty detection of bottlenecks which may occur due to lack of information in the NC program, for example.
CDOF2:	Deactivating collision detection for 3D circumferential milling The tool offset direction is determined from adjacent block parts with CDOF2. CDOF2 is only effective for 3D circumferential milling (CUT3Dx) and has the same meaning as CDOF for all other types of machining (e.g. CUT2D or CUT3DFD).

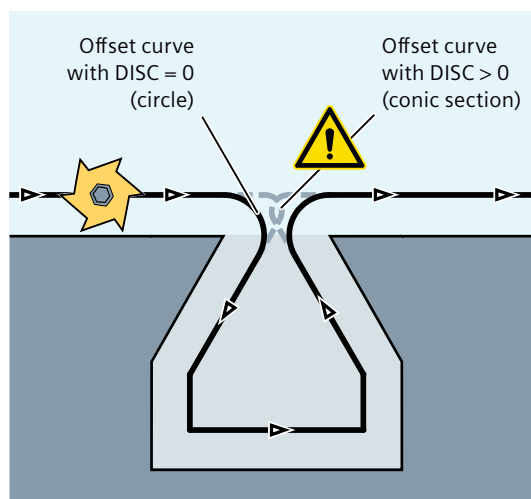
3.6.4 Supplementary conditions

Program test

To avoid program stops, the tool with the largest radius from the range of used tools should always be used during the program test.

"Bottleneck detection" due to overhangs at outside corners

When the intersections of non-consecutive blocks are checked, not the programmed original contours are examined, but the associated calculated equidistant paths. This can result in a "bottleneck" being falsely detected at outside corners. The reason for this is that the calculated tool path does not run equidistant to the programmed original contour when $DISC > 0$.



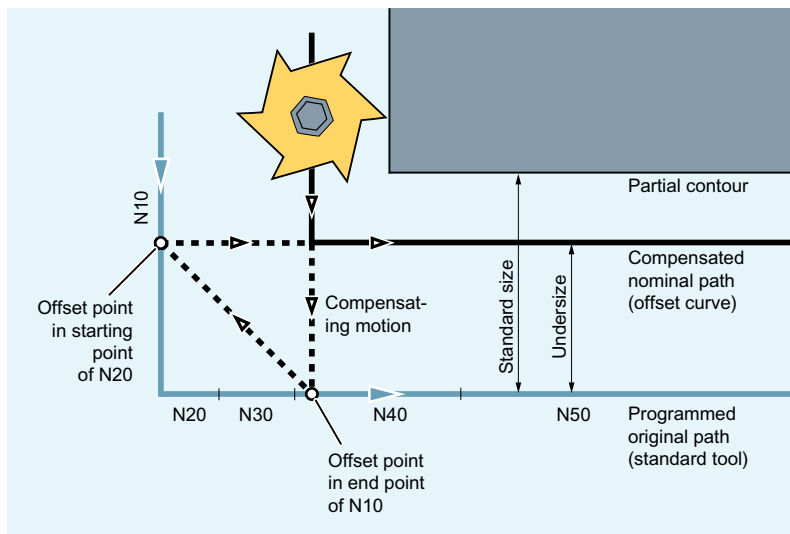
3.6.5 Example

Effect of collision detection using an example

The NC program describes the center point path of a standard tool. The contour for a tool that is actually used results in undersize, which is shown unrealistically large to demonstrate the geometric relationships in the following figure.

The control also only has an overview of three blocks in the example:

MD20240 \$MC_CUTCOM_MAXNUM_CHECK_BLOCKS = 3



Since an intersection exists only between the offset curves of the two blocks N10 and N40, the two blocks N20 and N30 would have to be omitted. In the example, the control does not know in block N40 if N10 has to be completely processed. Only a single block can therefore be omitted.

With active CDOF2, the compensation motion shown in the figure is executed and not stopped. In this situation, an active CDOF or CDON would result in an alarm.

3.7 Keep tool radius compensation constant

Meaning

The "Keep tool radius compensation constant" function is used to suppress tool radius compensation for a number of blocks, whereby a difference between the programmed and the actual traveled tool center path established by tool radius compensation in the previous blocks is retained as the offset.

It can be an advantage to use this method when several traversing blocks are required during line milling in the reversal points, but the contours produced by the tool radius compensation (bypass strategies) are not wanted.

3.7 Keep tool radius compensation constant

Activation

The "Keep tool radius compensation constant" function is activated with the G command CUTCONON (CUTter compensation CONstant ON) and deactivated with the G command CUTCONOF (CUTter compensation CONstant OFF).

CUTCONON and CUTCONOF form a modal G group.

The basic setting is CUTCONOF.

The function is applied independently of the type of tool radius compensation (2D-WRK, ...).

Normal case

Tool radius compensation is normally active before the compensation suppression and is still active when the compensation suppression is deactivated again.

In the last traversing block before CUTCONON, the offset point in the block end point is approached. All following blocks in which compensation suppression is active are traversed without compensation. However, they are offset by the vector from the end point of the last offset block to its offset point. These blocks can have any type of interpolation (linear, circular, polynomial).

The deactivation block of the compensation suppression, i.e. the block that contains CUTCONOF, is compensated normally. It starts in the offset point of the start point. One linear block is inserted between the end point of the previous block, i.e. the last programmed traversing block with active CUTCONON, and this point.

Circular blocks, for which the circle plane is perpendicular to the compensation plane (vertical circles), are treated as though they had CUTCONON programmed. This implicit activation of compensation suppression is automatically canceled in the first traversing block that contains a traversing motion in the compensation plane and is not such a circle. Vertical circle in this sense can only occur during circumferential milling.

Example:

N10	; Definition of tool d1
N20 \$TC_DP1[1,1] = 110	; Type
N30 \$TC_DP6[1,1]=	; Radius
N40	
N50 X0 Y0 Z0 G1 G17 T1 D1 F10000	
N60	
N70 X20 G42 NORM	
N80 X30	
N90 Y20	
N100 X10 CUTCONON	; Activation of the compensation suppression

N110 Y30 KONT	; On deactivation of contour suppression, insert bypass circle, if necessary
N120 X-10 CUTCONOF	
N130 Y20 NORM	; No bypass circle when deactivating the TRC
N140 X0 Y0 G40	
N150 M30	

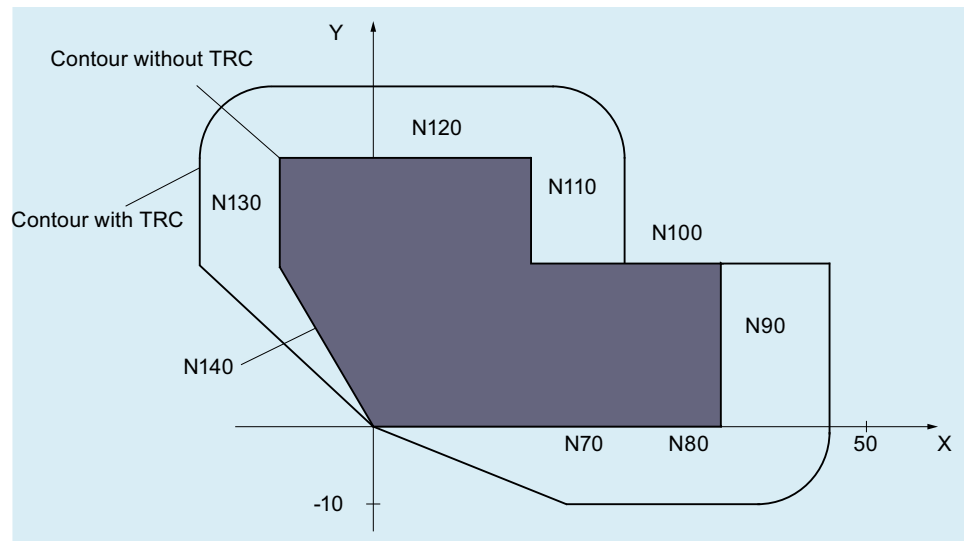


Figure 3-17 Sample program for contour suppression

Special cases

- If tool radius compensation is not active (G40), CUTCONON has no effect. No alarm is produced. The G command remains active, however. This is significant if tool radius compensation is to be activated in a later block with G41 or G42.
- A change from the G command to the 7th G group (tool radius compensation; G40 / G41 / G42) with active CUTCONON is permissible. A change to G40 is active immediately. The offset used for traversing the previous blocks is traveled.
- If CUTCONON or CUTCONOF is programmed in a block without traversing in the active compensation plane, activation is delayed until the next block that has such a traversing motion.
- If CUTCONON is programmed with active tool radius compensation and not canceled before the end of the program, the traversing blocks are traversed with the last valid offset. The same applies for reprogramming of G41 or G42 in the last traversing block of a program.
- If tool radius compensation is activated with G41 or G42 and CUTCONON is also already active, activation of compensation is delayed until the next traversing block with CUTCONOF.

- When repositioning with CUTCONOF, the 17th G group (approach and retraction behavior with tool offset; NORM / KONT) is evaluated, i.e. a bypass circle is inserted if necessary for KONT. A bypass circle is inserted under the same conditions as for activation of tool radius compensation with G41 or G42.
- The number of blocks with suppressed tool radius compensation is restricted: MD20252 \$MC_CUTCOM_MAXNUM_SUPPR_BLOCKS (Maximum number of blocks with compensation suppression)
If it is exceeded, machining is aborted and an error message issued.
The restriction is necessary because the internal block processing in the last block before CUTCONON must be resumed when repositioning.
- The response after reprogramming G41 or G42 when tool radius compensation is already active is similar to compensation suppression.
The following deviations apply:
 - Only linear blocks are permissible
 - A single traversing block that contains G41 or G42 is modified so that it ends at the offset point of the start point in the following block. Thus it is not necessary to insert an intermediate block. The same applies to the last block in a sequence of traversing blocks where each contains G41 or G42.
 - The contour is always reapproached with NORM, independent of the G command of the 17th group (approach and retraction behavior with tool offset; NORM / KONT).
- If G41 / G42 is programmed several times in consecutive traversing blocks, all blocks are machined as for CUTCONON, except for the last one.
- The type of contour suppression is evaluated only in the first traversing block of a sequence of blocks.
If both CUTCONON and G41 or G42 are programmed in the first block, the response to deactivating contour suppression is determined by CUTCONON.
Changing from G41 to G42 or vice versa makes sense in this case as a means of changing the compensation side (left or right of the contour) when restarting.
A change of the compensation side (G41/G42) can also be programmed in a later block, even if contour suppression is active.
- Collision monitoring and bottleneck detection is deactivated for all blocks with active contour suppression.

3.8 Smooth approach and retraction

3.8.1 Function

Description

The SAR (Smooth Approach and Retraction) function is used to achieve a tangential approach to the start point of a contour, regardless of the position of the start point.

The approach behavior can be varied and adapted to special needs using a range of additional parameters.

The two functions, smooth approach and smooth retraction, are largely symmetrical. The following section is, therefore, restricted to a detailed description of approach; special reference is made to differences affecting retraction.

Sub-movements

There are maximum 4 sub-movements in case of soft retraction and approach with the following positions:

- Start point of the movement P_0
- Intermediate points P_1 , P_2 and P_3
- End point P_4

Points P_0 , P_3 and P_4 are always defined. Intermediate points P_1 and P_2 can be omitted, according to the parameters defined and the geometrical conditions.

On retraction, the points are traversed in the reverse direction, i.e. starting at P_4 and ending at P_0 .

3.8.2 Parameters

The response of the smooth approach and retraction function is determined by up to 9 parameters:

Non-modal G command for defining the approach and retraction contour

This G command cannot be omitted.

- G147: Approach with a straight line
- G148: Retraction with a straight line
- G247: Approach with a quadrant
- G248: Retraction with a quadrant

3.8 Smooth approach and retraction

- G347: Approach with a semicircle
- G348: Retraction with a semicircle

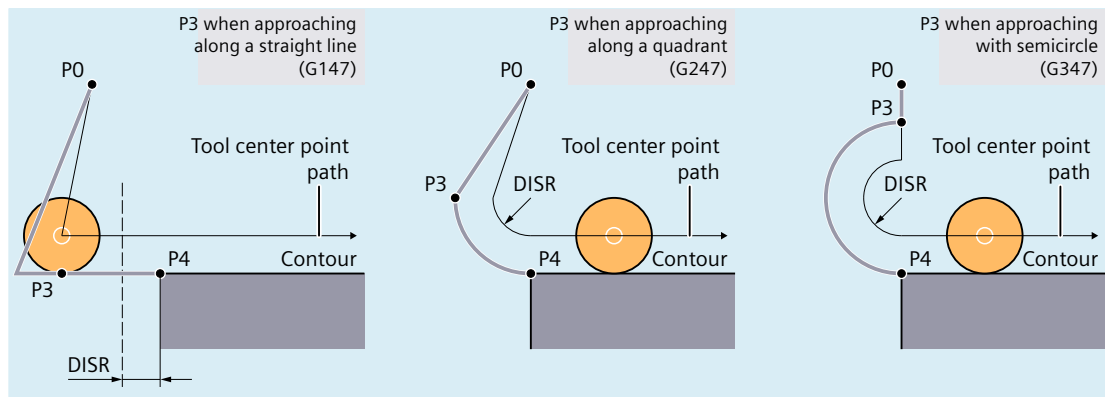


Figure 3-18 Approach behavior depending on G147 to G347 and DISR (with simultaneous activation of tool radius compensation)

Modal G command for defining the approach and retraction contour

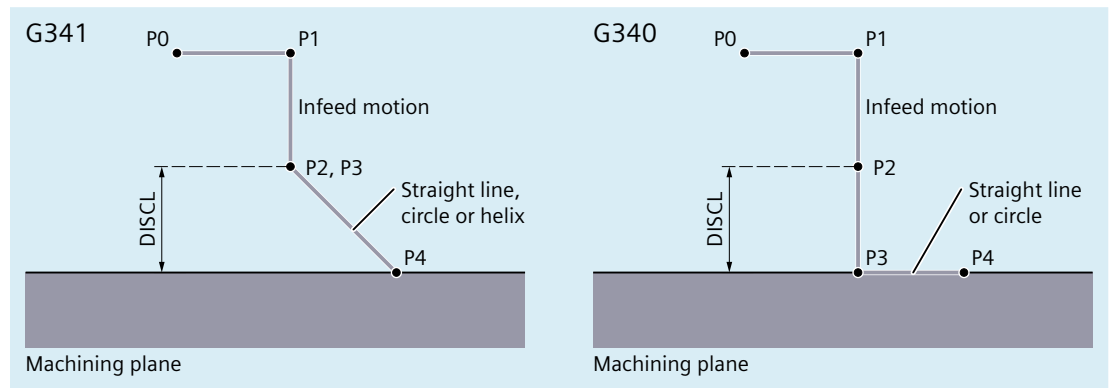
This G command is only relevant if the approach contour is a quadrant or semicircle. The approach and retraction direction can be determined as follows:

- G140:
Defining the approach and retraction direction using active tool radius compensation. (G140 is the basic setting value.)
With positive tool radius:
 - G41 active → approach from left
 - G42 active → approach from right
 If no tool radius compensation is active (G40), the response is identical to G143. In this case, an alarm is not output. If the radius of the active tool is 0, the approach and retraction side is determined as if the tool radius were positive.
- G141:
Approach contour from left, or retract to the left.
- G142:
Approach contour from right, or retract to the right.
- G143:
Automatic determination of the approach direction, i.e. the contour is approached from the side where the start point is located, relative to the tangent at the start point of the following block (P₄).

Note

The tangent at the end point of the preceding block is used accordingly on **retraction**. If the end point is not programmed explicitly on retraction, i.e. if it is to be determined implicitly, G143 is not permitted on retraction, since there is a mutual dependency between the approach side and the position of the end point. If G143 is programmed in this case, an alarm is output. The same applies if, when G140 is active, an automatic switchover to G143 takes place as a result of an inactive tool radius compensation.

Modal G command (G340, G341), which defines the subdivision of the movement into individual blocks from the start point to the end point



G34 The approach characteristic from P_0 to P_4 is shown in the figure.

0: If G247 or G347 is active (quadrant or semicircle) and start point P_3 is outside the machining plane defined by the end point P_4 , a helix is inserted instead of a circle. Point P_2 is not defined or coincides with P_3 .

The circle plane or the helix axis is determined by the plane, which is active in the SAR block (G17 - G19), i.e. the projection of the start tangent is used by the following block, instead of the tangent itself, to define the circle.

The movement from point P_0 to point P_3 takes place along two straight lines at the velocity valid before the SAR block.

G34 The approach characteristic from P_0 to P_4 is shown in the figure.

1: P_3 and P_4 are located within the machining plane, with the result that a circle is always inserted instead of a helix with G247 or G347.

Figure 3-19 Sequence of the approach movement depending on G340/G341

Note

Active, rotating frames are included in all cases where the position of the active plane G17 - G19 (circle plane, helix axis, infeed movements perpendicular to the active plane) is relevant.

DISR

DISR Specifies the length of a straight approach line or the radius of an approach arc.

Retraction/approach with straight lines

On approach/retraction along a straight line, DISR specifies the distance from the cutter edge to the start point of the contour, i.e. the length of the straight line with active TRC is calculated as the total of the tool radius and the programmed value of DISR.

An alarm is displayed:

- If DISR is negative and the amount is greater than the tool radius (the length of the resulting approach line is less than or equal to zero).

Retraction/approach with circles

Approach/retraction with circles

DISR indicates always the radius of the tool center point path. If tool radius compensation is activated, a circle is generated internally, the radius of which is dimensioned such that the tool center path is derived, in this case also, from the programmed radius.

An alarm is output on approach and retraction with circles:

- If the radius of the circle generated internally is zero or negative
- If DISR is not programmed
- If the radius value ≤ 0 .

DISCL

DISCL specifies the distance from point P₂ from the machining plane.

If the position of point P₂ is to be specified by an absolute reference on the axis perpendicular to the circle plane, the value must be programmed in the form DISCL = AC (....).

If DISCL is not programmed, points P₁, P₂ and P₃ are identical with G340 and the approach contour is mapped from P₁ to P₄.

The system checks that the point defined by DISCL lies between P₁ and P₃, i.e. in all movements, which have a component perpendicular to the machining plane (e.g. infeed movements, approach movements from P₃ to P₄), this component must have the same leading sign. It is not permitted to change direction. An alarm is output if this condition is violated.

On detection of a direction reversal, a tolerance is permitted that is defined by the machine data:

MD20204 \$MC_WAB_CLEARANCE_TOLERANCE (direction reversal on SAR).

However, if P₂ is outside the range defined by P₁ and P₃ and the deviation is less than or equal to this tolerance, it is assumed that P₂ is in the plane defined by P₁ and/or P₃.

Example:

An approach is made with G17 starting at position Z=20 of point P₁. The SAR plane defined by P₃ is at Z=0. The point defined by DISCL must, therefore, lie between these two points. MD20204=0.010. If P₂ is between 20.000 and 20.010 or between 0 and -0.010, it is assumed that the value 20.0 or 0.0 is programmed. The alarm is output if the Z position of P₂ is greater than 20.010 or less than -0.010.

Depending on the relative position of start point P₀ and end point P₄ with reference to the machining plane, the infeed movements are performed in the negative (normal for approach) or positive (normal for retraction) direction, i.e. with G17 it is possible for the Z component of end point P₄ to be greater than that of start point P₀.

Programming the end point P₄ (or P₀ for retraction) generally with X... Y... Z...**Possible ways of programming the end point P₄ for approach**

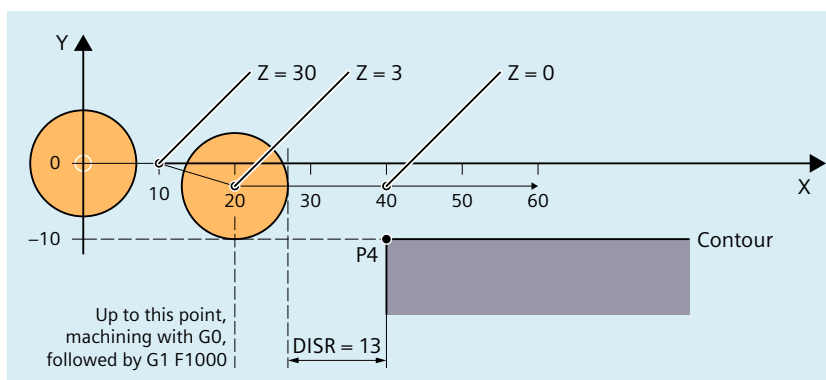
End point P₄ can be programmed in the actual SAR block.

P₄ can be determined by the end point of the next traversing block.

Further blocks (dummy blocks) can be inserted between the SAR block and the next traversing block without moving the geometry axes.

The end point is deemed to have been programmed in the actual SAR block for approach if at least one geometry axis is programmed on the machining plane (X or Y with G17). If only the position of the axis perpendicular to the machining plane (Z with G17) is programmed in the SAR block, this component is taken from the SAR block, but the position in the plane is taken from the following block. In this case, an alarm is output if the axis perpendicular to the machining plane is also programmed in the following block.

Example:



Program code	Comment
\$TC_DP1[1,1]=120	; Milling tool T1/D1
\$TC_DP6[1,1]=7	; Tool with 7mm radius
N10 G90 G0 X0 Y0 Z30 D1 T1	
N20 X10	
N30 G41 G147 DISCL=3 DISR=13 Z=0 F1000	
N40 G1 X40 Y-10	
N50 G1 X50	
...	
...	

N30/N40 can be replaced by:

```
N30 G41 G147 DISCL=3 DISR=13 X40 Y-10 Z0 F1000
```

or:

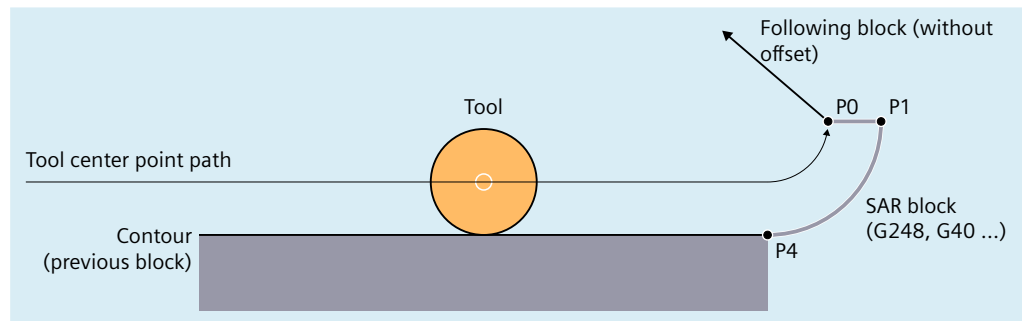
```
N30 G41 G147 DISCL=3 DISR=13 F1000
```

```
N40 G1 X40 Y-10 Z0
```

Possible ways of programming the end point P₀ for retraction

The end position is always taken from the SAR block, no matter how many axes have been programmed. We distinguish between the following situations:

1. No geometry axis is programmed in the SAR block.
In this case, the contour ends at point P_2 (or at point P_1 , if P_1 and P_2 coincide). The position in the axes, which describe the machining plane, is determined by the retraction contour (end point of the straight line or arc). The axis component perpendicular to this is defined by $DISCL$. If, in this case, $DISCL = 0$, the movement takes place completely within the plane.
2. Only the axis perpendicular to the machining plane is programmed in the SAR block.
In this case, the contour ends at point P_1 . The position of the two other axes is determined in the same way as in 1.



Retraction with SAR with simultaneous deactivation of TRC

If the SAR retraction block is also used to deactivate tool radius compensation, in the case of 1. and 2., an additional path from P_1 to P_0 is inserted such that no movement is produced when tool radius compensation is deactivated at the end of the retraction contour, i.e. this point defines the tool center point and not a position on a contour to be corrected.

3. At least one axis of the machining plane is programmed.
The second axis of the machining plane can be determined modally from its last position in the preceding block. The position of the axis perpendicular to the machining plane is generated as described in 1. or 2., depending on whether this axis is programmed or not. The position generated in this way defines the end point P_0 .
No special measures are required for deselection of tool radius compensation, because the programmed point P_0 already directly defines the position of the tool center point at the end of the complete contour.
The start and end points of the SAR contour (P_0 and P_4) can coincide on approach and retraction.

Velocity of the preceding block (typically G0).

All movements from point P_0 to point P_2 are performed at this velocity, i.e. the movement parallel to the machining plane and the part of the infeed movement up to the safety clearance.

Programming the feedrate with FAD

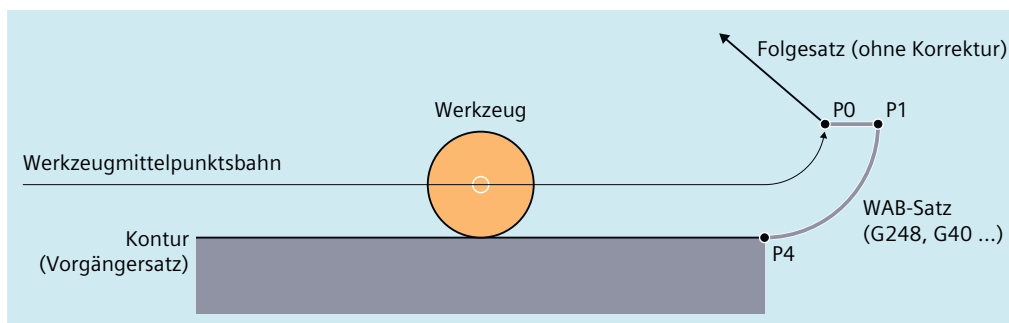
FAD programmed with ...	
G340	Feedrate from P_2 or P_3 to P_4 .
G341	Feedrate of the infeed movement perpendicular to the machining plane from P_2 to P_3 .

If **FAD** is not programmed, this part of the contour is traversed at the velocity, which is active modally from the preceding block, in the event that no **F** command defining the velocity is programmed in the SAR block.

Programmed response:

- FAD=0 or negative → Alarm Output
- FAD=... → Programmed value acts in accordance with the active G command of group 15 (feed type; G93, G94, etc.)
- FAD=PM (...) → Programmed value is interpreted as linear feed (like G94), irrespective of the active G command of group 15
- FAD=PR (...) → Programmed value is interpreted as revolutionary feed (like G95), irrespective of the active G command of group 15

Example:



Program code	Comment
\$TC_DP1[1,1]=120	; Milling tool T1/D1
\$TC_DP6[1,1]=7	; Tool with 7mm radius
N10 G90 G0 X0 Y0 Z20 D1 T1	
N20 G41 G341 G247 DISCL=AC(5) DISR=13FAD 500 X40 Y-10 Z=0 F2000	
N30 X50	
N40 X60	
...	

Programming feed F

This feed value is effective from point P_3 (or from point P_2 , if **FAD** is not programmed). If no **F** command is programmed in the SAR block, the speed of the preceding block is valid. The velocity defined by **FAD** is not used for following blocks.

3.8.3 Velocities

Velocities at approach

In both approach diagrams below, it is assumed that no new velocity is programmed in the block following the SAR block. If this is not the case, the new velocity comes into effect after point P₄.

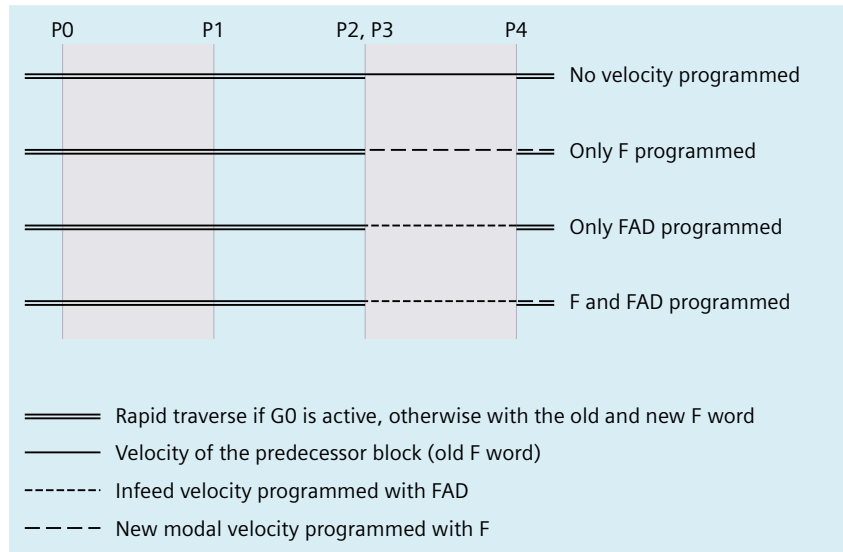


Figure 3-20 Velocities in the SAR subblocks on approach with G340

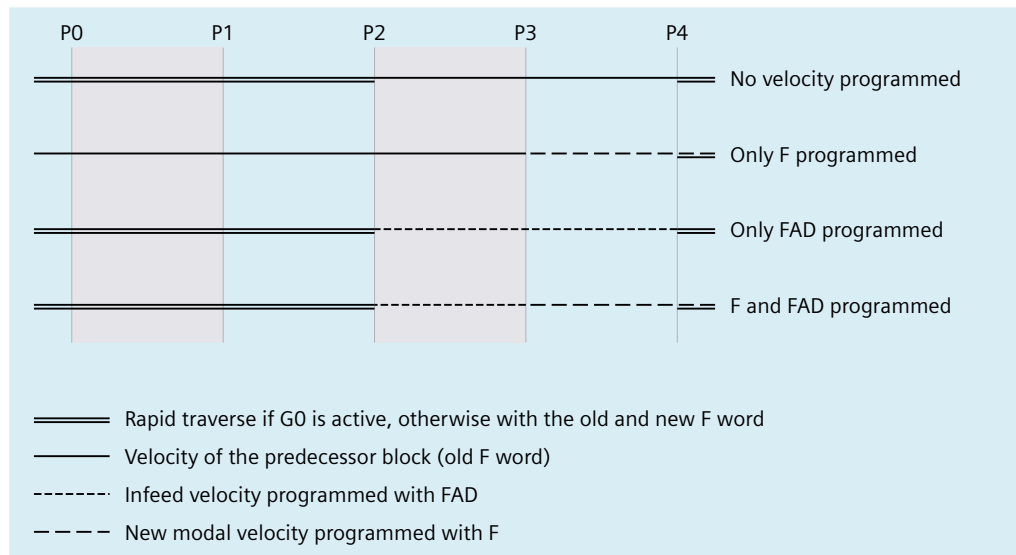


Figure 3-21 Velocities in the SAR subblocks on approach with G341

Velocities at retraction

During retraction, the rolls of the modally active feedrate from the previous block and the programmed feedrate value in the SAR block are interchanged, i.e., the actual retraction contour (straight line, circle, helix) is traversed with the old feedrate value and a new velocity programmed with the F word applies from point P_2 up to P_0 .

If even retraction is active and FAD is programmed, the path from P_3 to P_2 is traversed with FAD, otherwise it is traversed with the old velocity. The last F command programmed in a preceding block always applies for the path from P_4 to P_2 . G0 has no effect in these blocks.

Traversing from P_2 to P_0 takes place with the F command programmed in the SAR block or, if no F command is programmed, with the modal F command from a preceding block. This applies on the condition that G0 is not active.

If rapid traverse is to be used on retraction in the blocks from P_2 to P_0 , G0 must be activated before the SAR block or in the SAR block itself. If an additional F command is programmed in the actual SAR blocks, it is then ineffective. However, it remains modally active for following blocks.

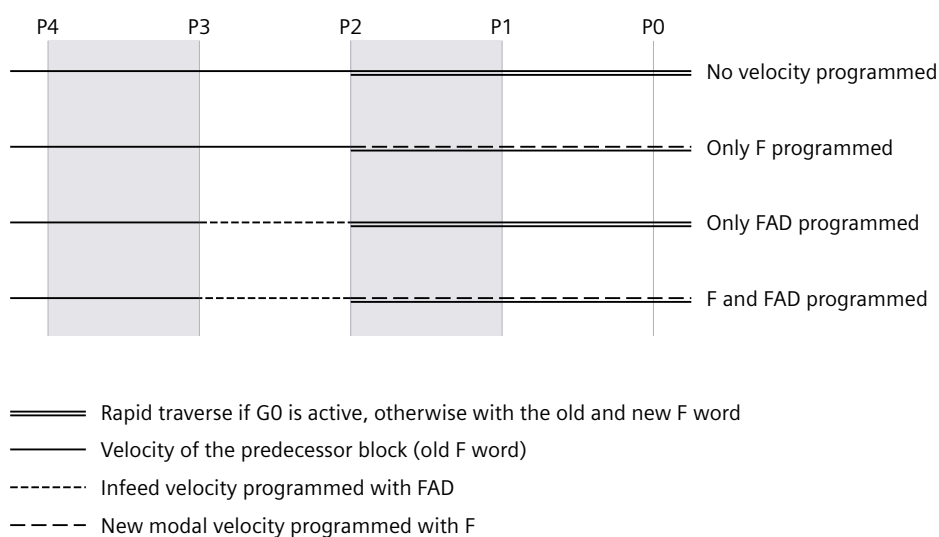


Figure 3-22 Velocities in the SAR subblocks on retraction

3.8.4 System variables

Points P_3 and P_4 can be read in the WCS as system variables during approach.

\$P_APR:	Read P_3 (start point) in WCS	
\$P_AEP:	Read P_4 (contour start point) in WCS	
\$P_APDV	=1	If the content of \$P_APR and \$P_AEP is valid, i.e., if these contain the position values belonging to the last SAR approach block programmed.
	=0	The positions of an older SAR approach block are read.

Changing the WCS between the SAR block and the read operation has no effect on the position values.

3.8.5 Supplementary conditions

Supplementary conditions

- Any further NC commands (e.g. auxiliary function outputs, synchronous axis movements, positioning axis movements, etc.) can be programmed in an SAR block.
These are executed in the first subblock on approach and in the last subblock on retraction.
- If the end point P_4 is not taken from the SAR block but from a subsequent traversing block, the actual SAR contour (straight line, quadrant or semicircle) is traversed in this block.
The last subblock of the original SAR block does not then contain traversing information for geometry axes. It is always output, however, because further actions (e.g. single axes) may have to be executed in this block.
- At least two blocks must always be taken into consideration:
 - The SAR block itself
 - The block, which defines the approach or retraction direction

Further blocks can be programmed between these two blocks.

The number of possible dummy blocks is limited with the machine data:

MD20202 \$MC_WAB_MAXNUM_DUMMY_BLOCKS (maximum number of blocks with no traversing motions with SAR).

- If tool radius compensation is activated simultaneously in an approach block the first linear block of the SAR contour is the block in which activation takes place.
The complete contour generated by the SAR function is treated by tool radius compensation as if it has been programmed explicitly (collision detection, calculation of intersection, approach behavior `NORM/KONT`).
- The direction of the infeed motion and the position of the circle plane or the helix axis are defined exclusively by the active plane ($G17 - G19$) - rotated with an active frame where appropriate.
- On approach, a preprocessor stop must not be inserted between the SAR block and the following block which defines the direction of the tangent.
Whether programmed explicitly or inserted automatically by the control, a preprocessor stop results, in this case, in an alarm.

Behavior with REPOS

If an SAR cycle is interrupted and repositioned, it resumes at the point of interruption on `RMIBL`. With `RMEBL`, the contact point is the end point of the last SAR block; with `RMBBL`, it is the start point of the first SAR block.

If `RMIBL` is programmed together with `DISPR` (reapproach at distance `DISPR` in front of interruption point), the reapproach point can appear in a subblock of the SAR cycle before the interruption subblock.

3.8.6 Examples

Example 1

The following conditions must be true:

- Smooth approach is activated in block N20
- X=40 (end point); Y=0; Z=0
- Approach movement performed with quadrant (G247)
- Approach direction not programmed, G140 is valid, i.e. because TRC is active (G42) and compensation value is positive (10), the contour is approached from the right
- Approach circle generated internally (SAR contour) has radius 20, so that the radius of the tool center path is equal to the programmed value DISR=10
- Because of G341, the approach movement takes place with a circle in the plane, resulting in a start point at (20, -20, 0)
- Because DISCL=5, point P2 is at position (20, -20, 5) and, because of Z30, point P1 is in N10 at (20, -20, 30)

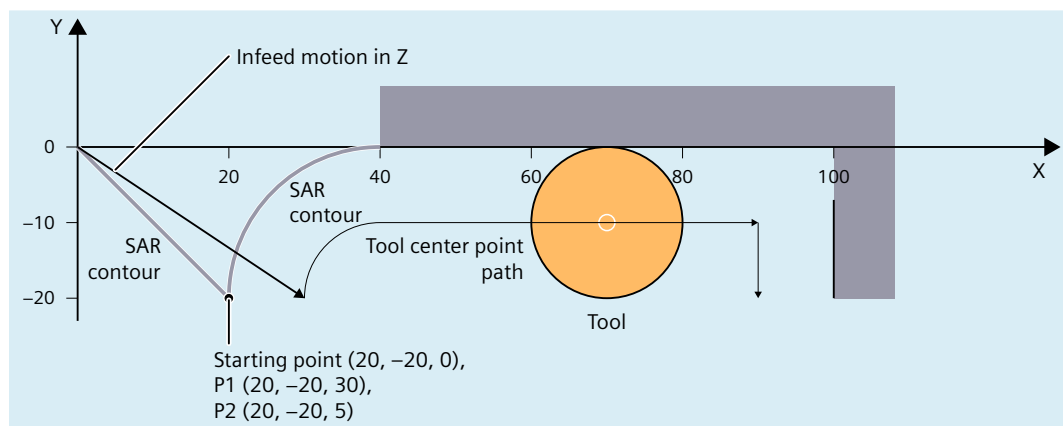


Figure 3-23 Contour example 1

Part program:

Program code	Comment
\$TC_DP1[1,1]=120	; Tool definition T1/D1
\$TC_DP6[1,1]=10	; Radius
N10 G0 X0 Y0 Z30	
N20 G247 G341 G42 NORM D1 T1 Z0 FAD=1000 F=2000 DISCL=5 DISR=10	
N30 X40	
N40 X100	
N50 Y-30	
...	

Example 2

The following conditions must be true for approach:

- Smooth approach is activated in block N20
- Approach movement performed with quadrant (G247)
- Approach direction not programmed, G140 is valid, i.e. because TRC is active (G41), the contour is approached from the left
- Contour offset OFFN=5 (N10)
- Current tool radius=10, and so the effective compensation radius for TRC=15; the radius of the SAR contour is thus equal to 25, with the result that the radius of the tool center path is equal to DISR=10
- The end point of the circle is obtained from N30, since only the Z position is programmed in N20
- Infeed movement
 - From Z20 to Z7 (DISCL=AC (7)) with rapid traverse
 - Then on to Z0 with FAD=200
 - Approach circle in X-Y-plane and following blocks with F1500
(In order that this velocity becomes effective in the following blocks, the active G-code G0 in N30 must be overwritten with G1. Otherwise, the contour would continue to be machined with G0.)

The following conditions must be true for retraction:

- Smooth retraction is activated in block N60
- Retraction movement performed with quadrant (G248) and helix (G340)
- FAD not programmed, since irrelevant for G340
- Z=2 in the start point; Z=8 in the end point, since DISCL=6

- When DISR=5, the radius of SAR contour=20; that of the tool center point path=5
- After the circle block, the retraction movement leads from Z8 to Z20 and the movement is parallel to the XY plane up to the end point at X70 Y0

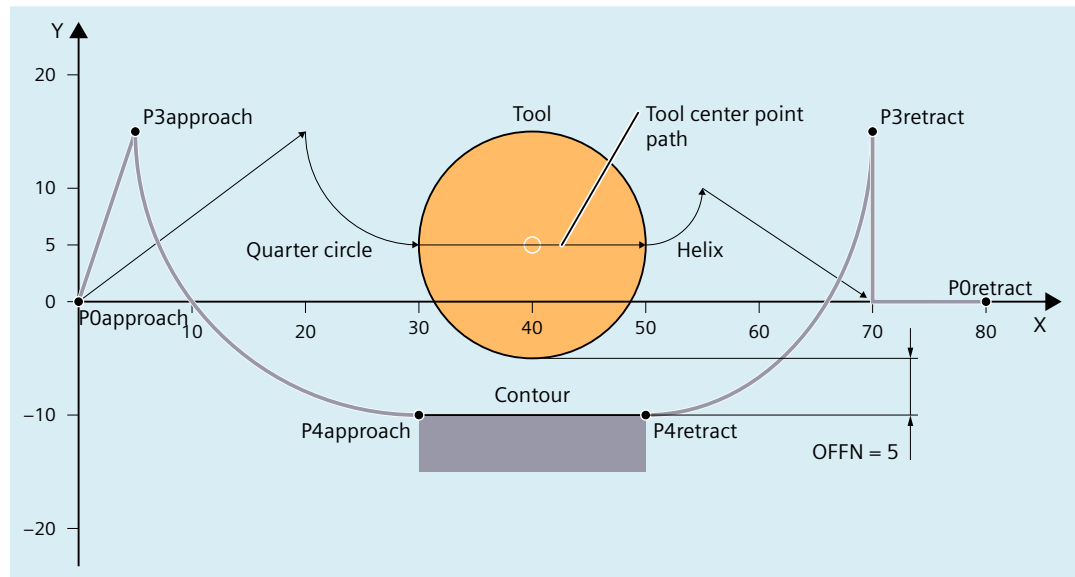


Figure 3-24 Contour example 2

Part program:

Program code	Comment
\$TC_DP1[1,1]=120	; Tool definition T1/D1
\$TC_DP6[1,1]=10	; Radius
N10 G0 X0 Y0 Z20 G64 D1 T1 OFFN = 5	; (P0app)
N20 G41 G247 G341 Z0 DISCL = AC(7) DISR = 10 F1500 FAD=200	; (P3app)
N30 G1 X30 Y-10	; (P4app)
N40 X40 Z2	
N50 X50	; (P4ret)
N60 G248 G340 X70 Y0 Z20 DISCL = 6 DISR = 5 G40 F10000	; (P3ret)
N70 X80 Y0	; (P0ret)
N80 M 30	

Note

The contour generated in this way is modified by tool radius compensation, which is activated in the SAR approach block and deactivated in the SAR retraction block.

The tool radius compensation allows for an effective radius of 15, which is the sum of the tool radius (10) and the contour offset (5). The resulting radius of the tool center path in the approach block is therefore 10, and 5 in the retraction block.

3.9 Extension of the approach and retract strategy

Function

In certain special geometrical situations, extended approach and retraction strategies, compared with the previous implementation, are required in order to activate or deactivate tool radius compensation (see figure below).

Note

The following example describes only the situation for deactivation of tool radius compensation: The response for approach is virtually identical.

Example

```
G42 D1 T1 ; Tool radius 20 mm
...
G1 X110 Y0
N10 X0
N20 Y10
N30 G40 X50 Y50
```

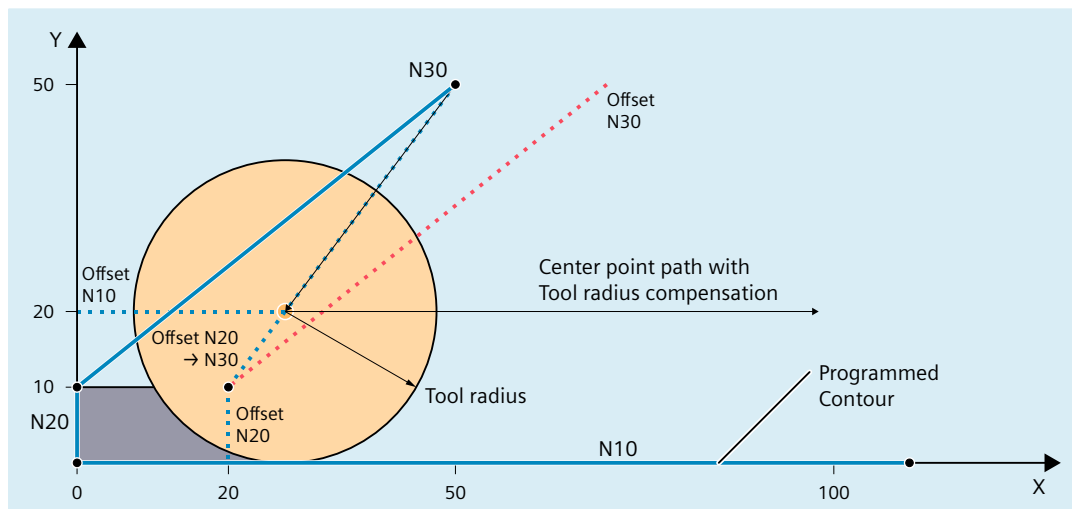


Figure 3-25 Retraction behavior with G460

The last block with active tool radius compensation (N20) is so short that an intersection no longer exists between the offset curve and the preceding block (or a previous block) for the current tool radius. An intersection between the offset curves of the following and preceding blocks is therefore sought, i.e. between N10 and N30 in this example. The curve used for the retraction block is not a real offset curve, but a straight line from the offset point at the end of block N20 to the programmed end point of N30. The intersection is approached if one is found. The colored area in the figure is not machined, although the tool used would be capable of this.

G460

With G460, the approach/retraction strategy is the same as before.

G461

If no intersection is possible between the last TRC block and a preceding block, the offset curve of this block is extended with a circle whose center point lies at the end point of the uncorrected block and whose radius is equal to the tool radius.

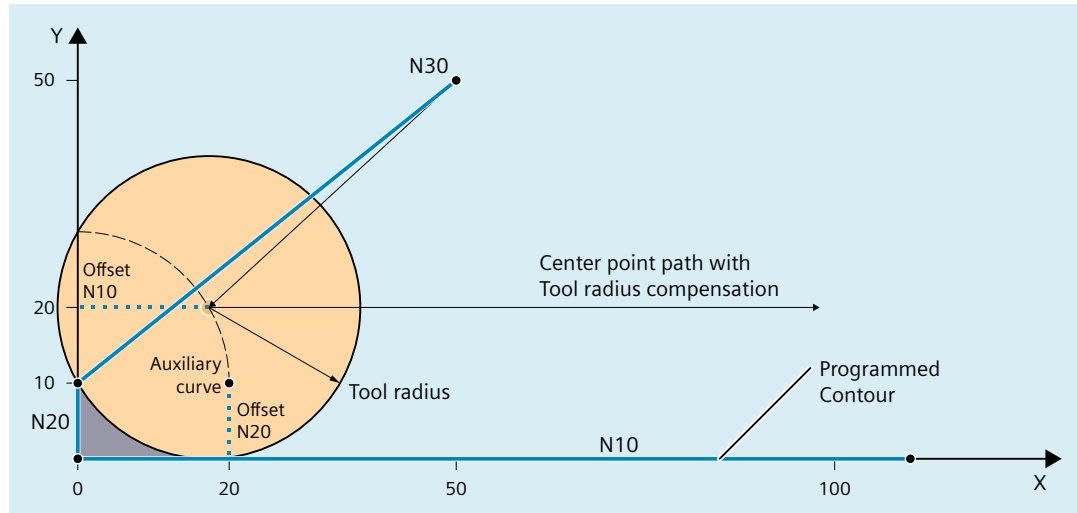


Figure 3-26 Retraction behavior with G461

The control attempts to cut this circle with one of the preceding blocks. If **CDOF** is active, the search is terminated when an intersection is found, i.e. the system does not check for more intersections with even earlier blocks.

If **CDON** is active, the search for more intersections continues after the first intersection is found.

An intersection point, which is found in this way, is the new end point of a preceding block and the start point of the deactivation block. The inserted circle is used exclusively to calculate the intersection and does not produce a traversing movement.

Note

If no intersection is found, the following alarm is output:

Alarm "10751 Collision danger"

G462

If no intersection is possible between the last TRC block and a preceding block, a straight line is inserted, on retraction with G462 (basic setting), at the end point of the last block with tool radius compensation (the block is extended by its end tangent).

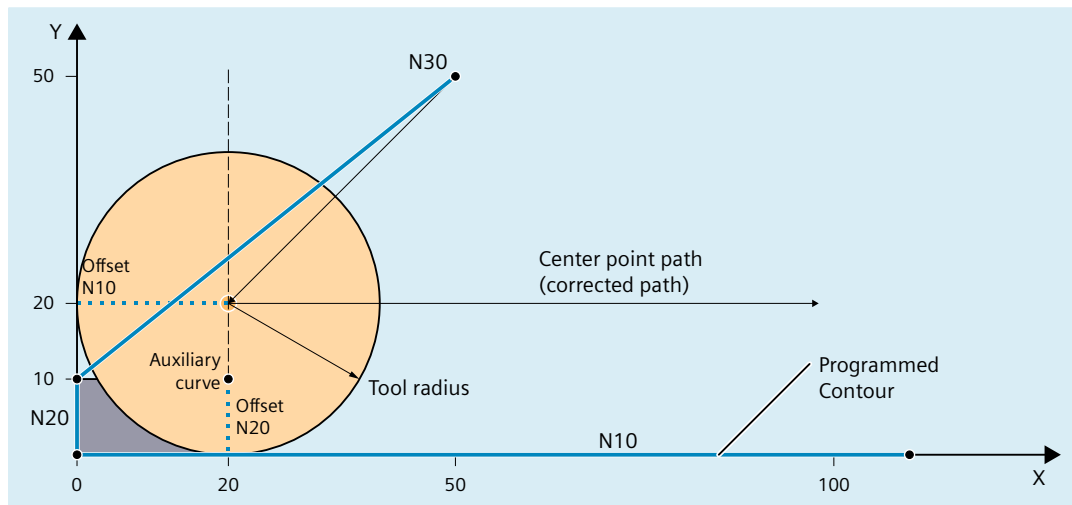


Figure 3-27 Retraction behavior with G462

The search for the intersection is then identical to the procedure for G461.

With G462, the corner generated by N10 and N20 in the sample program is not machined to the full extent actually possible with the tool used. However, this behavior may be necessary if the part contour (as distinct from the programmed contour), to the left of N20 in the example, is not permitted to be violated even with y values greater than 10 mm.

If KONT is active (travel round contour at start or end point), behavior will differ according to whether the end point is in front of or behind the contour.

End point in front of contour

If the end point is located in front of the contour, the retraction behavior is the same as for NORM. This feature does not change, even if the last contour block with G451 is extended with a straight line or a circle. Additional circumnavigation strategies to avoid a contour violation in the vicinity of the contour end point are therefore not required.

End point behind contour

If the end point is behind the contour, a circle or straight line is always inserted depending on G450/G451. In this case, G460-G462 has no effect.

If, in this situation, the last traversing block has no intersection with a preceding block, an intersection with the inserted contour element or with the linear section from the end point of the circumnavigation circle to the programmed end point can result.

If the inserted contour element is a circle (G450), and it intersects with the preceding block, this is the same as the intersection, which would be produced with NORM and G461. In general, however, a remaining section of the circle still has to be traversed. An intersection calculation is no longer required for the linear section of the retraction block.

In the second case (if no intersection is found between the inserted contour element and the preceding blocks), the intersection between the retraction straight line and a preceding block is approached.

Therefore, when G461 or G462 is active, behavior deviating from G460 can only arise if NORM is active or if behavior with KONT is identical to NORM due to the geometrical conditions.

Note

The approach behavior is symmetrical to the retraction behavior.

The approach/retraction behavior is determined by the state of the G command in the approach/retraction block. The approach behavior can therefore be set independently of the retraction behavior.

Example:

Program for using G461 during approach:

```
N10 $TC_DP1[1,1]=120           ; Milling tool type
N20 $TC_DP6[1,1]=10           ; Radius
N30 X0 Y0 F10000 T1 D1
N40 Y20
N50 G42 X50 Y5 G461
N60 Y0 F600
N70 X30
N80 X20 Y-5
N90 X0 Y0 G40
N100 M30
```

3.10 2 1/2 D tool radius compensation

The 2½ D tool radius compensation should be used if, when machining inclined surfaces, the **workpiece** is to be rotated, and not the tool alignment. This function is activated using commands CUT2D, CUT2DD, CUT2DF oder CUT2DFD.

Tool length offset

The tool length compensation is always taken into account referred to the machining plane that is not rotated and is fixed in space.

2½ D tool radius compensation for contour tools

2½ D tool radius compensation for contour tools is activated, if, together with CUT2D, CUT2DD, CUT2DF or CUT2DFD, one of the two commands G41 (tool radius compensation left of the contour) or G42 (tool radius compensation right of the contour) is programmed. It

3.10 2 1/2 D tool radius compensation

is used for automatic cutting-edge selection in the case of non-axially symmetrical tools that can be used for piece-by-piece machining of individual contour segments.

Note

If 2 1/2 D tool radius compensation is not activated, a contour tool behaves like a standard tool, which only has the first cutting edge.

2 1/2 tool radius compensation referred to a differential tool

2 1/2 D tool radius compensation, referred to a differential tool, is activated using the `CUT2DD` or `CUT2DFD` commands. It should be applied if the programmed contour refers to the center point path of a differential tool, and a tool other than a differential tool is used for machining. When calculating the 2 1/2 D tool radius compensation, only the wear of the radius of the active tool (`$TC_DP_15`) and the possibly programmed tool offset `OFFN` and `TOFFR` are taken into account. The basic radius (`$TC_DP6`) of the active tool is **not** taken into account.

Syntax

`CUT2D`
`CUT2DD`
`CUT2DF`
`CUT2DFD`

Meaning

<code>CUT2D:</code>	Activating the 2 1/2 D radius compensation
<code>CUT2DD:</code>	Activating the 2 1/2 D radius compensation referred to a differential tool
<code>CUT2DF:</code>	Activating 2 1/2 D radius compensation, tool radius compensation relative to the current frame and/or inclined plane
<code>CUT2DFD:</code>	Activating 2 1/2 D radius compensation, tool radius compensation relative to the current frame and/or inclined plane

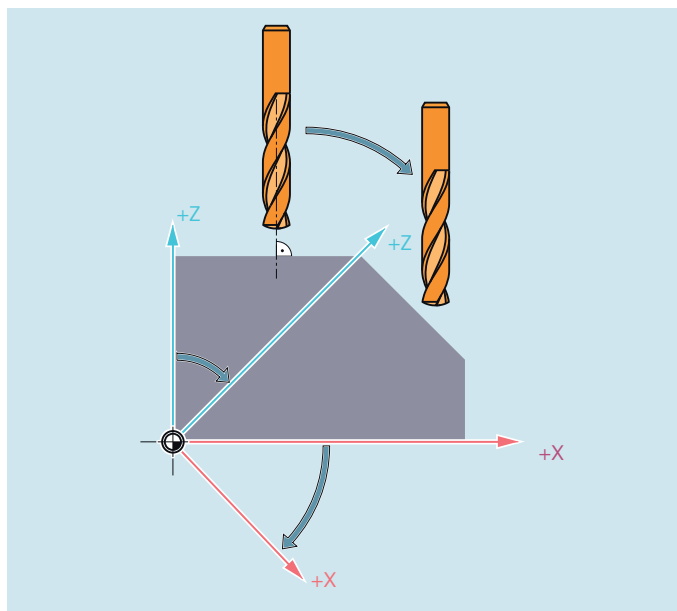
More information

Contour tools

- Enabling
Tool radius compensation for contour tools is enabled on a channel-specific basis using:
MD28290 \$MC_MM_SHAPED_TOOLS_ENABLE
- Tool type
Contour tool types are defined on a channel-specific basis using:
MD20370 \$MC_SHAPED_TOOL_TYPE_NO
- Cutting edge
A number of cutting edges (D numbers) can be assigned to each contour tool in any sequence. The maximum number of cutting edges per tool is parameterized using:
MD18106 \$MN_MM_MAX_CUTTING_EDGE_PERTOOL
The first cutting edge of a contour tool is the cutting edge, which is selected when activating the tool. If, e.g. in a program, using the commands T3 D5, the fifth cutting edge (D5) of the third tool (T3) is activated, then D5 and the following cutting edges define with one part, or altogether, the contour tool. The cutting edges located before D5 are ignored.

2½ D tool radius compensation without rotating the compensation plane (CUT2D, CUT2DD)

If a frame that contains a rotation is programmed, then for CUT2D or CUT2DD, the plane in which the tool radius compensation (compensation plane) takes place **is not rotated at the same time**. The tool radius compensation is taken into account, referred to the **non rotated** machining plane (G17, G18, G19). The tool length compensation acts relative to the compensation plane.

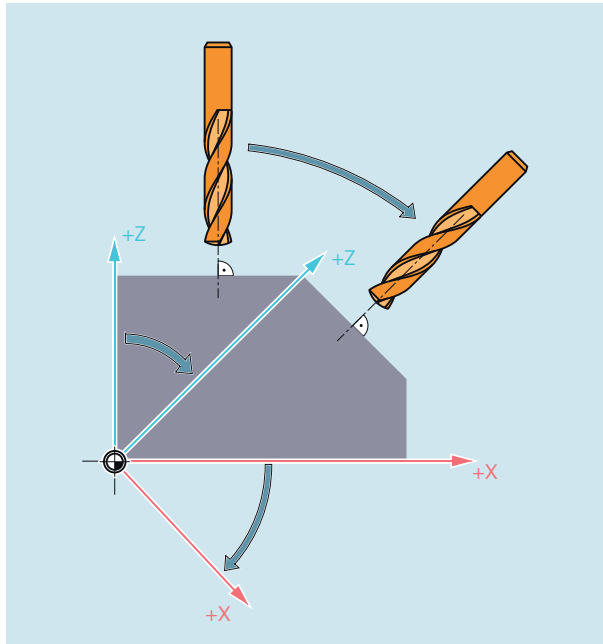


For machining inclined surfaces, the tool offsets must be appropriately defined or calculated based on the functions for "Tool length compensation for tools that can be orientated".

2½ D tool radius compensation with rotation of the compensation plane (CUT2DF, CUT2DFD)

If a frame that contains a rotation is programmed, then for CUT2DF or CUT2DFD, the plane in which the tool radius compensation takes place (compensation plane) **is also rotated**. The tool radius compensation is taken into account, referred to the **rotated** machining plane (G17, G18, G19). However, the tool length compensation still acts relative to the **non-rotated** machining plane.

Requirement: At the machine, the tool orientation must be able to be adjusted perpendicular to the rotated machining plane, and set for machining.

**Note**

The tool length compensation continues to be active relative to the non-rotated working plane.

Approach and retraction for 2½ D tool radius compensation

The approach/retract response when tool radius compensation is active for the cases where the activation and/or deactivation block does not contain any traversing information (only for 2½ D tool radius compensation with CUT2D or CUT2DF) is defined using the setting data:

SD42494 \$SC_CUTCOM_ACT_DEACT_CTRL

The following decimal coding applies:

Thousands position (10 ³)	Hundreds position (10 ²)	Tens position (10 ¹)	Units position (10 ⁰)
Retraction response for tools without cutting edge position (milling tools)	Retraction response for tools with cutting edge position ¹⁾ (turning tools)	Approach response for tools without cutting edge position (milling tools)	Approach response for tools with cutting edge position ¹⁾ (turning tools)
¹⁾ Tools with cutting edge position are tools with tool numbers between 400 and 599 (turning and grinding tools), whose cutting edge position has a value that lies between 1 and 8. Turning and grinding tools with cutting edge position 0 or 9 or other, undefined values, are treated in the same way as milling tools.			

If the relevant position has a 1, then approach and retraction are performed, even if only G41/G42 or G40 are present in the block.

Example:

```
N100 X10 Y0
N110 G41
N120 X20
```

If a tool radius of 10 mm is assumed in the example, then position X10 Y10 is approached in block N110.

If the relevant position has a 2, approach and retraction are only performed if at least one geometry axis is programmed in the activation/deactivation block. If the same result as in the example is to be achieved with this setting, then the program must be changed as follows, for example:

```
N100 X10 Y0
N110 G41 X10
N120 X20
```

If axis data X10 is not included in block N110, then activation of the tool radius compensation is delayed by one block, i.e. the activation block would then be block N120.

If the relevant position has a 3, then retraction is not performed in a deactivation block (G40) if only the geometry axis is programmed perpendicular with respect to the compensation plane. In this case, initially the axis moves perpendicular with respect to the compensation plane. The axis then retracts in the compensation plane. In this case, after G40, the block must contain motion information in the compensation plane. Approach movements for values 2 and 3 are identical.

If the relevant position has a 4 (only in the case of a tool without cutting edge position when retracting, i.e. the thousands digit), then for a deactivation block, the next programmed movement is performed if no motion was programmed in it.

Example:

```
N1040 G41 T1 D1
N1050 X20
N1060 X30
N1070 X50 ; Offset active
```

3.11 Slot shape recognition (option)

```

N1080 UP
N1090 X70 F10000 ; Traverse offset with F10000
N1100 G01
N1110 G90
N1120 X90 F12000
N1130 X100
N1140 M30

```

```

N2000 PROC UP
N2010 G40
N2030 RET

```

If the relevant position has a number other than 1, 2 or 3, i.e. especially a value of 0, then in a block that does not include any traversing information, approach and retraction are not performed.

Note

Tools with cutting edge position

If the value of this setting data is changed in a program, it is recommended that a preprocessing stop (STOPRE) is programmed before writing, as otherwise there is a risk that the new value will be used in upstream parts of the program. The inverse case is not critical. This means that if the setting data is written to, subsequent NC blocks will definitely access the changed value.

See also

Tool length compensation (Page 19)

3.11 Slot shape recognition (option)

If tool radius compensation is active and collision detection (Page 88) is switched on, a programmed slot with a width smaller than the tool diameter is detected as a possible contour violation and it is omitted from machining. In certain technology-dependent application cases (e.g. manufacturing deflection lines with a laser cutting machine), however, it is necessary to machine the slots when the radius of the tool (laser beam) is somewhat larger than the programmed width of the slot would allow. To give the user this option, the tool radius compensation has been expanded to include the function "slot shape recognition".

Note

The "Recognition of slot form" function is an option that requires a license and must be assigned to the hardware via license management!

Function

If the "slot shape recognition" function is active, slots that have a predefined shape are automatically identified and their machining is ensured.

The following are detected:

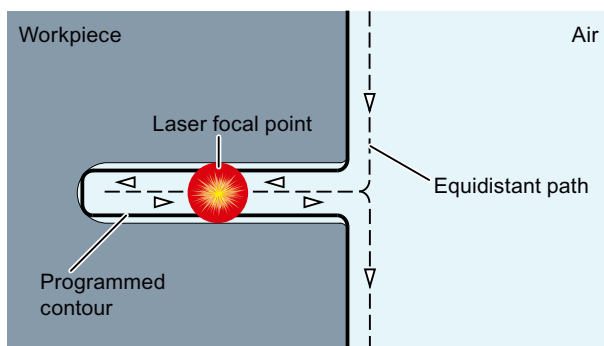
- I-slot



- T-slot



If the slot width is greater than or equal to the tool diameter, the slot will be traversed as programmed. If the slot width is smaller than the tool diameter, the equidistant path is calculated in such a way that the tool traverses along the center of the programmed slot:



The following conditions must be met:

- The radius of a new tool (R2) must not exceed the radius of the original tool (R1) by the factor 2:

$$R2 < 2 * R1$$
- The slot must have an odd number of blocks.
- The central block must comprise a straight line.
- The radius of the rounding at the start of the slot or the length of the block preceding the slot must not be less than the difference $R2 - R1$.

Activation

To activate the function "slot shape recognition", the following channel-specific setting data must be set to "1":

SD42977 \$SC_SLOT_FORM_RECOGN = 1

Parameterization

Number of blocks for predictive feature recognition

The slot shape recognition is only active if the following channel-specific machine data is set to a value greater than zero:

MD28620 \$MC_MM_NUM_FEATURE_BLOCKS > 0

With MD28620, the minimum number of blocks is defined that are considered at any one time in order to detect the slot shape.

Recommended value: 15 (This corresponds to the number of contour elements for a T-slot with an arc between the straight lines.)

Example

In the following part program section, a deflection line is programmed that will be cut with a laser beam.

Program code	Comment
N1 G01 G90 G71 F10000	
N2 \$TC_DP1[1,1] = 120	
N3 \$TC_DP3[1,1] = 0.0	
N4 \$TC_DP6[1,1] = 0.2	; Tool radius = 0.2 mm
N6 G91 G41 G01 T=1 D=1	; Activate TRC.
N7 X100.00	
N8 Y-24.500	
N9 G02 X-0.200 Y-0.200 I-0.200	; Rounding at the start of the deflection line contour.
N10 G01 X-49.700	
N11 G03 X-0.100 Y-0.100 J-0.100	
N12 G01 Y-0.100	
N13 G03 X0.100 Y-0.100 I0.100	
N14 G01 X49.700	
N15 G02 X0.200 Y-0.200 J-0.200	; Rounding at the end of the deflection line contour.
N16 G01 Y-24.500	



If slot shape recognition is inactive (SD42977 \$SC_SLOT_FORM_RECOGN = 0), the tool radius compensation will detect and omit the programmed deflection line as an impermissible

contour for the tool used (laser beam) because the slot width is 0.3 mm but the tool has radius 0.2 mm. On the other hand, if the slot shape recognition is active (SD42977 \$SC_SLOT_FORM_RECOGN = 1), the tool radius compensation will detect the programmed deflection line as an I-slot and the laser beam will cut the deflection line along the center of the programmed slot.

Because, in this example, the slot comprises seven contour elements (4 arcs, 3 straight lines), the following minimum value must be set in MD28620 \$MC_MM_NUM_FEATURE_BLOCK:

MD28620 \$MC_MM_NUM_FEATURE_BLOCK $\geq$ 7

3.12 Tool carriers with orientation capability

3.12.1 General information

The orientation of the tool can vary, e.g. due to retooling, for one single class of machine tools. When the machine is operating, the orientation that has been set is permanent, however, and cannot be changed during traversing. For this reason, kinematic orientation transformation (3, 4 or 5axis transformations, TRAORI) is neither necessary nor does it make sense for such machines. It is however necessary to consider the changes in the tool length components attributable to the change in the orientation.

The following figure shows an example of an orientable tool carrier. It shows a Cardan tool carrier with two axes for the tool orientation:

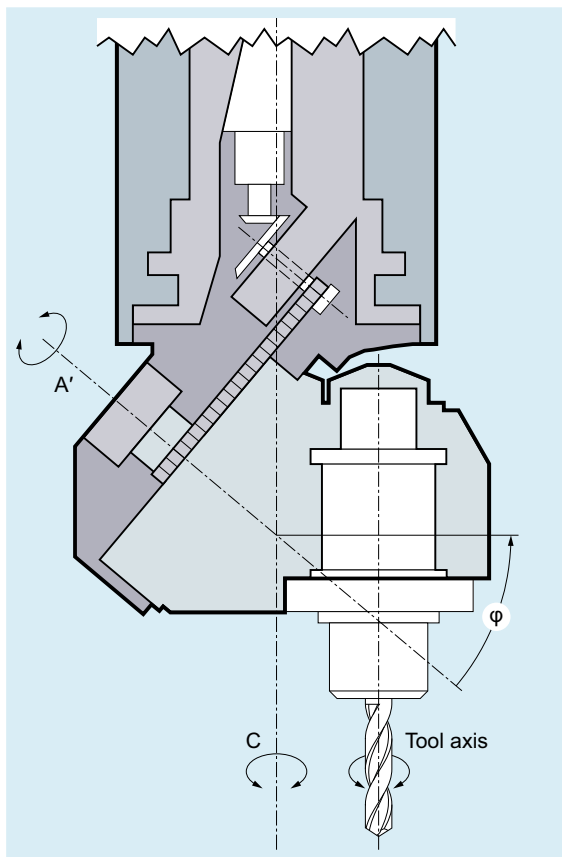


Figure 3-28 Cardan tool carrier with two axes

Prerequisites

The following data must be available so that the control can take tool offsets for orientable tool carriers into account:

- Tool data (geometry, wear, etc.)
- Tool carrier data (data for the geometry of the tool carrier with orientation capability)

Tool carrier selection

The tool carrier is selected by programming the address TCARR in the NC program:

TCARR=<m>

<m>: Number of the tool carrier

The tool carrier has an associated tool carrier data block that describes its geometry.

Selection and activation of the tool carrier and its data block has an immediate effect, i.e. starting with the next traversing block.

Assignment tool/tool carrier

The tool that was active previously is assigned to the new tool carrier.

Note

From the point of view of the control, tool carrier numbers <m> and tool numbers <n> can be freely combined. In the real application, however, certain combinations can be ruled out for machining or mechanical reasons. The control does not check whether the combinations make sense.

Kinematics of the tool carrier

The kinematics of the tool carrier can be configured on the one hand via machine data (TC_CARRx...) or via the system variables of a kinematic chain:

- The kinematics of the tool carrier with orientation capability is described with a total of 33 parameter sets (kinematic chain of the tool carrier).
The data of the data set can be edited by the user (Configuring tool carrier kinematics via machine data (Page 124))
- The kinematics is configured via system variables (kinematic chain of the machine, Configuring tool carrier kinematics with kinematic chain (Page 135))

Processing tool carrier data blocks

The following options are available for editing the tool carrier data blocks:

- Explicit entry in the tool carrier data block from the NC program
- Automatic acceptance of certain values (angles) from a frame
A requirement for this is that the G command TCOFR (Tool Carrier Orientation FFrame) is also specified when the tool carrier is selected.
The tool orientation used to calculate the tool length is determined again from the frame active at this time when a tool carrier is changed.

Orientation in Z direction

The G command TOFRAME defines a frame such that the Z direction in this frame is the same as the current tool orientation.

If no tool carrier or a tool carrier without change in orientation is active, then the following applies for the Z direction in the new frame:

- With G17, the Z direction in the new frame is the same as the old Z direction.
- With G18, the Z direction in the new frame is the same as the old Y direction.
- With G19, the Z direction in the new frame is the same as the old X direction.

Absolute tool carrier orientation with active frame

The absolute tool carrier orientation is set with the G command TCOABS (Tool Carrier Orientation ABSolute).

3.12 Tool carriers with orientation capability

The orientation taken into account for the tool length compensation is **independent** of the orientation of the active frame.

Either TCOABS or TCOFR can be valid.

Frame change

The user can change the frame after selecting the tool. This does not have any effect on the tool length compensation components.

Angles in the tool carrier data

The programmed angles of rotation stored in the tool carrier data is not affected by the angle of rotation defined by the frames. When changing from TCOFR to TCOABS, the original (programmed) angles of rotation in the tool carrier data is reactivated.

Tool radius compensation

Tool radius compensation takes account of the current tool orientation when CUT2D or CUT3DFS is active.

For the other correction types of G group 22, with the exception of CUT3DC and CUT3DF, the behavior with respect to the plane used for correction is unchanged. This is determined independent of the tool orientation from the active frame.

For CUT2DF and CUT3DFF, the compensation plane used for tool radius compensation is determined from the frame **independently** of the current tool orientation. The active working plane (G17/G18/G19) is considered.

CUT3DC and CUT3DF

3D tool offset for circumferential milling and 3D tool offset for face milling with active 5-axis transformation are not affected by the "Tool carriers with orientation capability" function.

The orientation information is determined by the active kinematic 5-axis transformation.

Limited tool carrier orientation

An alarm is output if an orientation that cannot be reached with the defined tool carrier kinematic is specified by the frame.

The situation that not every orientation can be achieved is given in the following cases:

- The two rotary axes which are necessary to define the kinematics are not perpendicular to each other and the tool axis which defines the tool direction is not perpendicular to the second rotary axis.
- Fewer than two axes are defined.

Non-rotary tool carriers

The tool orientation used internally is dependent only on the basic orientation of the tool and the active machining plane (G17/G18/G19).

Ambiguities

With two axes, a particular tool orientation defined by the frame can generally be set with **two** different rotation angle pairs. Of these two, the control selects the setting with which the rotary angle is as close as possible to the programmed rotary angle.

Storing angles in the tool carrier data

In virtually any case where ambiguities may arise, it is necessary to store the approximate angle expected from the frame in the tool carrier data.

3.12.2 Create new tool carrier

Connection tool carrier - TO unit

\$MC_MM_NUM_TOOL_CARRIER_CHAN specifies the total number of tool carrier data sets in the TO unit assigned to the channel. (Definition of TO unit: see Function Manual Tool Management)

- If the value is ≥ 0 , the machine data indicates how many tool carriers can be defined in the assigned TO unit. The remaining carriers are evenly distributed between the other TO units where the value is -1 (default behavior).
- If the sum of all ≥ 0 values in \$MC_MM_NUM_TOOL_CARRIER_CHAN is greater than \$MN_MM_NUM_TOOL_CARRIER, an alarm 17090 "Value greater than upper limit" is output.

Each TO unit is assigned a fixed number of tool carriers. The value is calculated as follows:

3 channels, each assigned a separate TO unit \$MN_MM_NUM_TOOL_CARRIER = 12. Output: \$MC_MM_NUM_TOOL_CARRIER_CHAN values are default = -1

\$MC_MM_NUM_TOOL_CARRIER_CHAN			Number of tool carriers			Remark
CH1	CH2	CH3	CH1	CH2	CH3	
-1	-1	-1	4	4	4	
-1		-1	6	0	6	
-1	-1	2	5	5	2	
3	-1	5	3	4	5	
3	-1	0	3	9	0	
3	2	1	3	2	1	Memory reserved only for 6 carriers. The remaining 6 are not reserved, no alarm
3	4	5	3	4	5	
3	5	7	3	5	4	Alarm 17090 - MD for CH3 not set, remains -1
5	7	9	5	7	0	Alarm 17090 - MD for CH3 not set, remains -1
3	8	10	5	3	3	Alarm 17090 - MD for CH2 and CH3 not set, remains -1
-1	6	9	3	6	3	Alarm 17090 - MD for CH3 not set, remains -1
7	-1	7	7	2	2	Alarm 17090 - MD for CH3 not set, remains -1
-1	3	-1	4	3	4	Memory reserved for 11 carriers only, no alarm

3.12 Tool carriers with orientation capability

\$MC_MM_NUM_TOOL_CARRIER_CHAN			Number of tool carriers			Remark
-1	9	-1	1	9	1	Memory reserved for 11 carriers only, no alarm
-1	11	-1	0	11	0	Memory reserved for 11 carriers only, no alarm

CH = Channel

3.12.3 Configuring tool carrier kinematics via machine data

3.12.3.1 Kinematic interaction and machine design

Representation of the kinematic chain

The concept of the kinematic chain is used to describe the kinematic interaction between a reference point and the tool tip.

The chain specifies all the data required for the tool carrier data set in a schematic. To describe the concrete case with a particular kinematic, the relevant components of the chain must be assigned real vectors, lengths and angles. The chain represents the maximum constellation. In simpler applications, individual components can be zero (e.g. kinematics with one or no axis of rotation).

The machine does not have to have axes that rotate the tool and/or workpiece table. The function can be used even if the orientations are set manually by handwheels or reconfiguration.

The machine design is described by the following parameters:

- Two axes of rotation (v_1 and v_2), each with one angle of rotation (α_1 or α_2), which counts positively for clockwise rotation facing the direction of the rotation vector.
- Up to four offset vectors (l_1 to l_4) for relevant machine dimensions (axis distances, distances to machine or tool reference points).

Zero vectors

Vectors v_1 and v_2 can be zero. The associated angle of rotation (explicitly programmed or calculated from the active frame) must then also be zero, since the direction of the axis of rotation is not defined. If this condition is not satisfied, an alarm is produced when the tool carrier is activated.

Fewer than two axes of rotation

The option not to define an axis of rotation makes sense when the tool carrier to be described can only rotate the tool in one plane. A sensible minimum data set may, therefore, contain only one single entry not equal to 0 in the tool carrier data; namely, a value in one of the components of v_1 or v_2 for describing an axis of rotation parallel to the axis where the angle of rotation α_1 or α_2 is determined from one frame.

Further special cases

Vectors v_1 and v_2 can be colinear. However, the degree of freedom for orientation is lost, i.e. this type of kinematic is the same as one where only one axis of rotation is defined. All possible orientations lie on one cone sheath. The conical sheath deforms to a straight line if tool orientation t and v_1 or v_2 become colinear. Change of orientation is, therefore, not possible in this special case. The cone sheath deforms to a circular surface (i.e. all orientations are possible in one plane), if tool orientation t and v_1 or v_2 are perpendicular to each other.

It is permissible for the two vectors v_1 and v_2 to be zero. A change in orientation is then no longer possible. In this special case, any lengths l_1 and l_2 , which are not equal to zero, act as additional tool length compensations, in which the components in the individual axes are not affected by changing the plane (G17 - G19).

Kinematics data expansions

- Possibility of direct access to existing machine axes in order to define the setting of the tool carriers via the rotary axis positions.
- Extension of the kinematics with rotary workpiece and on kinematics with rotary tool and rotary workpiece.
- Possibility to permit only discrete values in a grid for the rotary axis positions (Hirth joints).

The extensions are compatible with earlier software versions and encompass the kinematic data sets from \$TC_CARR18 to \$TC_CARR23.

Machine with rotary tool

On machines with rotary tool there is no change in the definition of the kinematics compared to older software versions. The newly introduced vector l_4 , in particular, has no significance. Should the contents of l_4 not be zero, this is ignored.

The term "Toolholder with orientation capability" is actually no longer really appropriate for the new kinematic types, with which the table can also be rotated, either alone or additionally to the tool. However, it has been kept for reasons of compatibility.

The kinematic chains used to describe the machine with rotary tool (general case) are shown in the figure below:

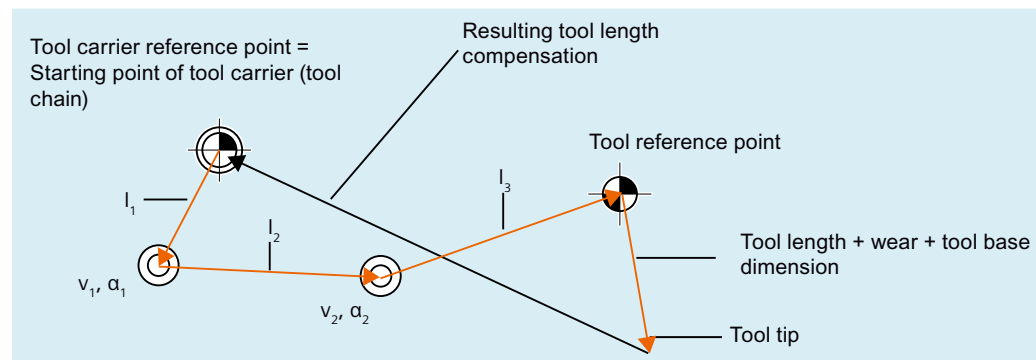


Figure 3-29 Kinematic chain to describe a tool with orientation

Vectors which describe offsets in the rotary head are positive in the direction from the tool tip to the reference point of the tool carrier.

3.12 Tool carriers with orientation capability

The following kinematic type is defined for machines with a rotary tool:

\$TC_CARR23 using letter T

Machine with rotary workpiece

On machines with rotary workpiece, the vector l_1 has no significance. If it contains a value other than zero, this is ignored.

The kinematic chain for machines with rotary workpiece is shown in the figure below.

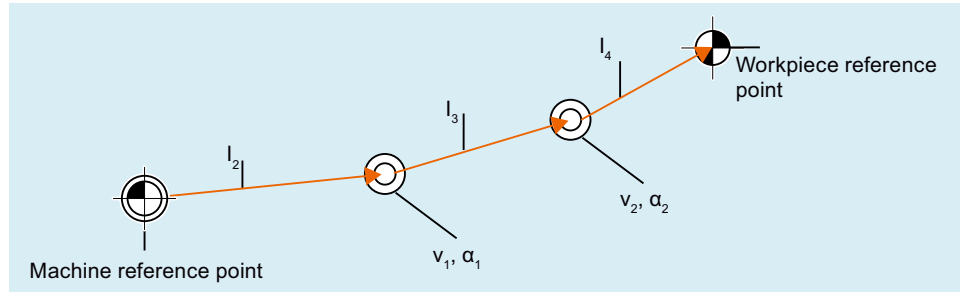


Figure 3-30 Kinematic chain to describe a rotary table

Vectors, which describe offsets in the rotary table, are positive in the direction from the machine reference point to the table.

The following kinematic type is defined for machines with a rotary workpiece:

\$TC_CARR23 using letter P

Note

On machines with rotatable workpiece, it is generally useful if the selected machine reference point and the reference point of the table are identical. Selecting the reference points in this way has the advantage that the position of the workpiece zero in the initial state (i.e. with rotary axes not turned) does not change when the rotary table is activated. The (open) kinematic chain (see figure) is then closed.

In this special case, therefore, the following formula applies: $l_2 = -(l_3 + l_4)$

Machines with extended kinematics

On machines with extended kinematics (both tool and workpiece are rotary), it is only possible to turn each of the components with one axis.

The kinematic of the rotary tool is described with the first axis of rotation (v_1) and the two vectors l_1 and l_2 , that of the rotary table with the second axis of rotation (v_2) and the two vectors l_3 and l_4 . The two kinematic chain components for machines with rotary tool and rotary workpiece are shown in the figure below.

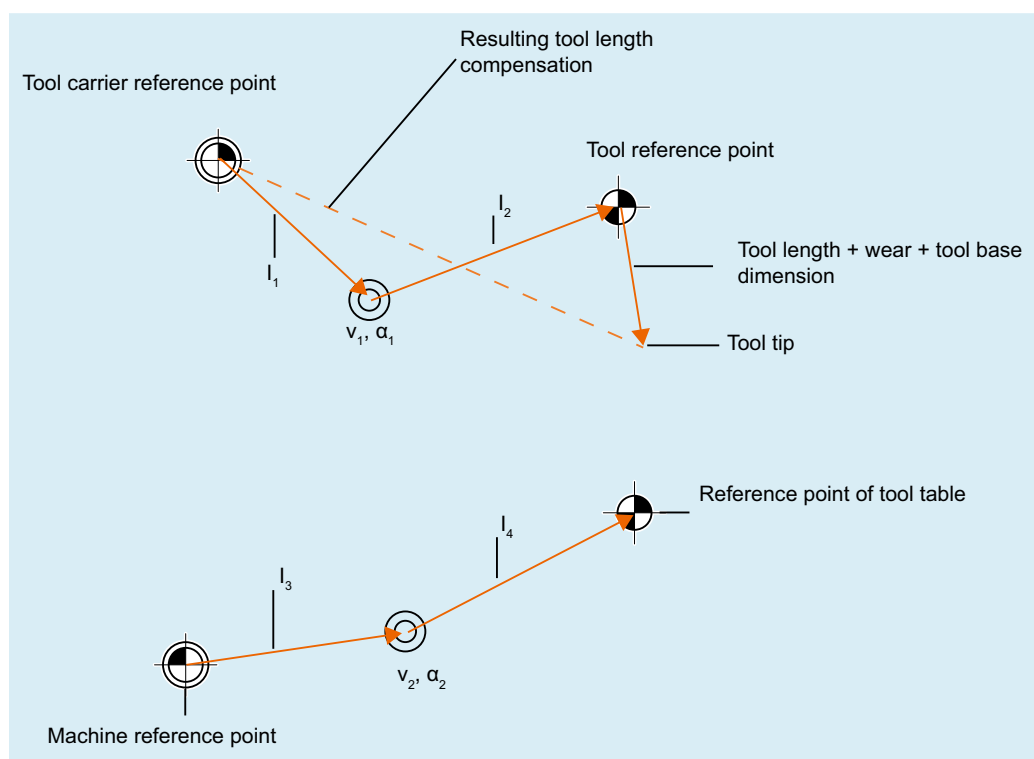


Figure 3-31 Kinematic sequence with extended kinematics

The following kinematic type is defined for machines with a rotary tool and rotary workpiece:

\$TC_CARR23 using letter M (extended kinematics)

Note

On machines with extended kinematics it is generally useful, as with machines where only the table can be rotated, for the machine reference point and the reference point of the table to be identical. The (open) chain component to describe the table (see figure) is then closed.

In this special case, the following formula applies: $l_3 = -l_4$

Rotary tool types T and M

For machine kinematics with a rotary tool (types T and M), the tool carrier component with orientation capability, which describes the tool or head component (as opposed to the table component), acts, in conjunction with the active tool, as a new overall tool.

Fine offset

The offset vectors l_1 to l_4 and the offsets of the axes of rotation v_1 and v_2 can be represented as the sum of a basic value and a fine offset. The fine offset parameters assigned to the basic values are achieved by **adding a value of 40 to the index of the basic value**.

Example:

3.12 Tool carriers with orientation capability

The parameter \$TC_CARR5 is assigned to the fine offset \$TC_CARR45.

Note

For the significance of the system variables \$TC_CARR41 to \$TC_CARR65 available for the fine offset see:

Further information

Programming Manual Advanced; Tool offsets

Activation

The following setting adds the fine offset values to the basic values:

SD42974 \$SC_TOCARR_FINE_CORRECTION = 1 (fine offset TCARR on/off)

Constraints

The amount is limited to the permissible fine offset.

The maximum permissible value is defined:

Components	Machine data
• Components of vectors l_1 to l_4 :	MD20188 \$MC_TOCARR_FINE_LIM_LIN
• Offsets of the two axes of rotation v_1 and v_2 :	MD20190 \$MC_TOCARR_FINE_LIM_ROT

An impermissible fine offset value is only detected upon occurrence of the following conditions:

- A tool carrier capable of orientation which has such a value is activated
- The following setting data is applied at the same time:
SD42974 \$SC_TOCARR_FINE_CORRECTION

Description of a rotation

A data set for describing a rotation comprises one vector v_1/v_2 to describe the direction of rotation of the axis of rotation in its initial state and an angle α_1/α_2 . The angle of rotation is counted positively for clockwise rotation facing the direction of the rotation vector.

The two tool carrier angles α_1 and α_2 are determined using a frame, independent of the active plane currently selected (G17 - G19).

The tool orientation in the initial state (both angles α_1 and α_2 are zero) is (as in the default case):

- G17: Parallel to Z.
- G18: Parallel to Y.
- G19: Parallel to X.

Assigning data to the tool carrier

Example of a machine with rotary tool carrier

The following settings are obtained at the mill head shown for a machine with tool carrier with orientation capability of kinematic type T:

Component of the offset vector $l_1 =$	$(-200, 0, 0)$
Component of the offset vector $l_2 =$	$(0, 0, 0)$
Component of the offset vector $l_3 =$	$(-100, 0, 0)$
Component of axis of rotation $v_1 =$	$(1, 0, 0)$
Component of axis of rotation $v_2 =$	$(-1, 0, 1)$
Tool base dimension of tool reference point	$(0, 0, 250)$

Note

The tool reference point for the tool base dimension is defined by the reference point at the machine.

For further information regarding the reference points in the working area, see Function Manual "Basic Functions", Chapter "Axes, coordinate systems, frames".

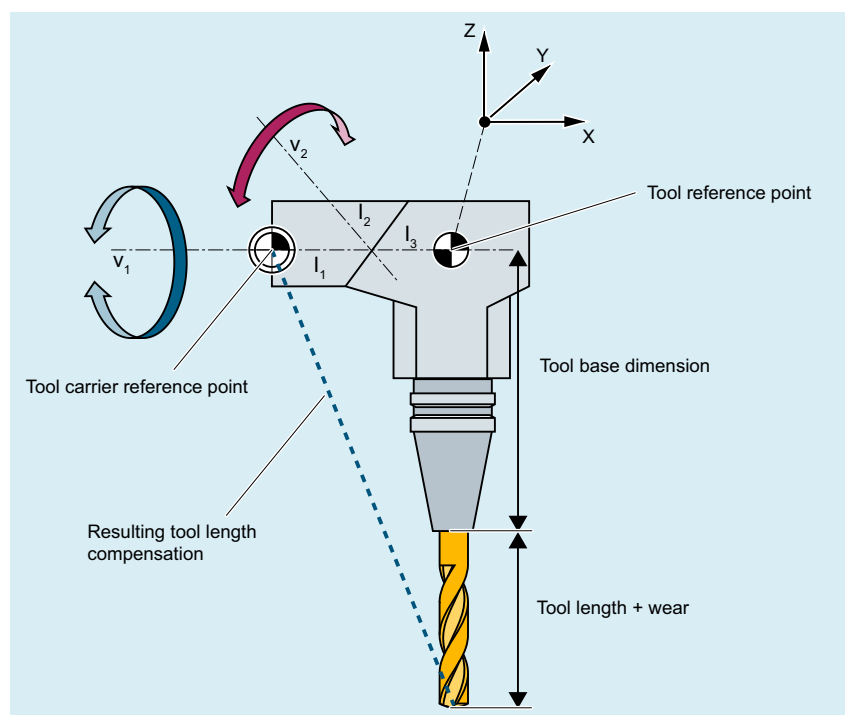


Figure 3-32 Assignment of the tool carrier data

3.12 Tool carriers with orientation capability

Suitable assumptions were made for the following values in the data set:

- The two axes of rotation intersect at one point.
All components of l_2 are therefore zero.
- The first axis of rotation lies in the x/z plane, the second rotary axis is parallel to the x axis.
These conditions define the directions of v_1 and v_2 (the lengths are irrelevant, provided that they are not equal to zero).
- The reference point of the tool carrier lies 200 mm in the negative x direction viewed from the intersection of the two rotary axes.
This condition defines l_1 .

Specify associated data set values

The following associated data set values are specified for the tool carrier shown on a machine with rotary tool carrier:

Description	NC variable	Value
x component of offset vector l_1	\$TC_CARR1	-200
y component of offset vector l_1	\$TC_CARR2	0
z component of offset vector l_1	\$TC_CARR3	0
x component of offset vector l_2	\$TC_CARR4	0
y component of offset vector l_2	\$TC_CARR5	0
z component of offset vector l_2	\$TC_CARR6	0
x component of rotary axis v_1	\$TC_CARR7	1
y component of rotary axis v_1	\$TC_CARR8	0
z component of rotary axis v_1	\$TC_CARR9	0
x component of rotary axis v_2	\$TC_CARR10	-1
y component of rotary axis v_2	\$TC_CARR11	0
z component of rotary axis v_2	\$TC_CARR12	1
Angle of rotation α_1 (in degrees)	\$TC_CARR13	0
Angle of rotation α_2 (in degrees)	\$TC_CARR14	0
x component of offset vector l_3	\$TC_CARR15	-100
y component of offset vector l_3	\$TC_CARR16	0
z component of offset vector l_3	\$TC_CARR17	0

Explanations

The tool carrier kinematic chosen in the example is such that the two axes of rotation form an angle of 45 degrees, which means that the orientation cannot take just any value. In concrete terms, this example does not permit the display of orientations with negative X components.

x component of the tool base dimension:	0
y component of the tool base dimension:	0
z component of the tool base dimension:	250

Note

The required data cannot be determined unequivocally from the geometry of the tool carrier, i.e. the user is free to a certain extent to decide the data to be stored. Thus, for the example, it is possible to specify only one z component for the tool base dimension up to the second axis. In this case, l_2 would no longer be zero, but would contain the components of the distance between this point on the second axis and a further point on the first axis. The point on the first axis can also be selected freely. Depending on the point selected, l_1 must be selected such that the reference point (which can also be selected freely) is reached.

In general: Vector components that are not changed by rotation of an axis can be distributed over any vectors "before" and "after" rotation.

3.12.3.2 Parameter sets for tool carriers

Parameter sets

A complete parameter set for a tool carrier with orientation capability consists of 33 values, which are stored in the following system variables:

- \$TC_CARR1 to \$TC_CARR33

In addition, there are system variables at the free disposal of the user and for fine offsets:

- \$TC_CARR34 to \$TC_CARR65

The significance of the individual parameters is distinguished as follows.

Machine kinematics

\$TC_CARR1 to \$TC_CARR20 and \$TC_CARR23

\$TC_CARR18 to \$TC_CARR20 define a further vector l_4 , which is needed to describe the machine with extended kinematics (both tool and workpiece can be rotated).

\$TC_CARR21 and \$TC_CARR22 contain the channel-axis names of the rotary axes, the positions of which can be used to determine the orientation of the tool carrier with orientation capability, if necessary.

Kinematic type

\$TC_CARR23 = <kinematic type>

<Kinematic type>	Meaning
T	Only the (Tool) can be rotated (basic value).
P	Only the workpiece (Part) can be rotated.
M	Both tool and workpiece can be rotated (Mixed mode).

Parameterization of the axes of rotation

\$TC_CARR24 to \$TC_CARR33

The system variables in \$TC_CARR24 to \$TC_CARR33 can be used to define offsets, angle offsets, Hirth joints, and axis limits.

3.12 Tool carriers with orientation capability

Components and presetting of the chain/data block

The values \$TC_CARR1 to \$TC_CARR20 and \$TC_CARR24 to \$TC_CARR33 in the tool carrier data block are of NC language format type REAL..

The values \$TC_CARR21 and \$TC_CARR22 for the axis name of the first rotary axis (v_1) and the second rotary axis (v_2) are of NC language format type AXIS.

They are all preset to zero.

The value \$TC_CARR23 is initialized with the uppercase letter "T" (only tool can be rotated).

\$TC_CARR[0]= 0 has a special significance.

3.12.3.3 Tool carrier data set**Tool carrier data set**

A complete parameter set for a tool carrier with orientation capability consists of 33 values, which are stored in the following system variables:

\$TC_CARR<n>[<m>]

<n>: Parameters 1 ... 33

<m>: Number of the orientable tool carrier

Value range: 1 ... MD18088 \$MN_MM_NUM_TOOL_CARRIER (maximum number of definable tool carriers)

System variable	Meaning	Data type	Default setting
\$TC_CARR1	x component of offset vector l_1	REAL	0
\$TC_CARR2	y component of offset vector l_1	REAL	0
\$TC_CARR3	z component of offset vector l_1	REAL	0
\$TC_CARR4	x component of offset vector l_2	REAL	0
\$TC_CARR5	y component of offset vector l_2	REAL	0
\$TC_CARR6	z component of offset vector l_2	REAL	0
\$TC_CARR7	x component of rotary axis v_1	REAL	0
\$TC_CARR8	y component of rotary axis v_1	REAL	0
\$TC_CARR9	z component of rotary axis v_1	REAL	0
\$TC_CARR10	x component of rotary axis v_2	REAL	0
\$TC_CARR11	y component of rotary axis v_2	REAL	0
\$TC_CARR12	z component of rotary axis v_2	REAL	0
\$TC_CARR13	Angle of rotation α_1 (in degrees)	REAL	0
\$TC_CARR14	Angle of rotation α_2 (in degrees)	REAL	0
\$TC_CARR15	x component of offset vector l_3	REAL	0
\$TC_CARR16	y component of offset vector l_3	REAL	0
\$TC_CARR17	z component of offset vector l_3	REAL	0
\$TC_CARR18	x component of offset vector l_4	REAL	0
\$TC_CARR19	y component of offset vector l_4	REAL	0

System variable	Meaning	Data type	Default setting
\$TC_CARR20	z component of offset vector l_4	REAL	0
\$TC_CARR21	Tool carrier of the rotary axis v_1	AXIS	0
\$TC_CARR22	Axis name of the rotary axis v_2	AXIS	0
\$TC_CARR23	Kinematic type	CHAR	T
\$TC_CARR24	Offset of rotary axis v_1	REAL	0
\$TC_CARR25	Offset of rotary axis v_2	REAL	0
\$TC_CARR26	Angle offset of rotary axis v_1 (Hirth tooth)	REAL	0
\$TC_CARR27	Angle offset of rotary axis v_2 (Hirth tooth)	REAL	0
\$TC_CARR28	Angle increment of rotary axis v_1 (Hirth tooth)	REAL	0
\$TC_CARR29	Angle increment of rotary axis v_2 (Hirth tooth)	REAL	0
\$TC_CARR30	Minimum position of rotary axis v_1 (SW limit)	REAL	0
\$TC_CARR31	Minimum position of rotary axis v_2 (SW limit)	REAL	0
\$TC_CARR32	Maximum position of rotary axis v_1 (SW limit)	REAL	0
\$TC_CARR33	Maximum position of rotary axis v_2 (SW limit)	REAL	0

Machine kinematics

The system variables \$TC_CARR1 to \$TC_CARR20 and \$TC_CARR23 describe the kinematics of the orientable tool carrier.

\$TC_CARR18 to \$TC_CARR20 define a further vector l_4 , which is needed to describe the machine with extended kinematics (both tool and workpiece can be rotated).

\$TC_CARR21 and \$TC_CARR22 contain the channel-axis names of the rotary axes, the positions of which can be used to determine the orientation of the tool carrier with orientation capability, if necessary.

Kinematic type

The kinematic type of the orientable tool carrier is set with the following system variable:

\$TC_CARR23 = <kinematic type>

<Kinematic type>	Meaning
T	Only the (Tool) can be rotated (default value).
P	Only the workpiece (Part) can be rotated.
M	Both tool and workpiece can be rotated (Mixed mode).

Parameterization of the axes of rotation

The system variables in \$TC_CARR24 to \$TC_CARR33 can be used to define offsets, angle offsets, Hirth joints, and axis limits.

Additional parameters/fine offsets

In addition, there are system variables at the free disposal of the user and for fine offsets:

- \$TC_CARR34 to \$TC_CARR40
Contain parameters, which are freely available to the user.
- \$TC_CARR41 to \$TC_CARR65
Contain fine offset parameters that can be added to the values in the basic parameters. The fine offset value assigned to a basic parameter is obtained when the value 40 is added to the parameter number.
- \$TC_CARR47 to \$TC_CARR54 and \$TC_CARR61 to \$TC_CARR63
Not defined and produce an alarm if read or write access is attempted.

System variable	Meaning	Data type	Default setting
\$TC_CARR34	Tool carrier name	STRING[32]	""
\$TC_CARR35 *)	Axis name 1	STRING[32]	""
\$TC_CARR36 *)	Axis name 2	STRING[32]	""
\$TC_CARR37 *)	Identifier	INT	0
\$TC_CARR38 *)	Position component X	REAL	0
\$TC_CARR39 *)	Position component Y	REAL	0
\$TC_CARR40 *)	Position component Z	REAL	0
\$TC_CARR41	x comp. fine offset of offset vector l_1	REAL	0
\$TC_CARR42	y comp. fine offset of offset vector l_1	REAL	0
\$TC_CARR43	z comp. fine offset of offset vector l_1	REAL	0
\$TC_CARR44	x comp. fine offset of offset vector l_2	REAL	0
\$TC_CARR45	y comp. fine offset of offset vector l_2	REAL	0
\$TC_CARR46	z comp. fine offset of offset vector l_2	REAL	0
\$TC_CARR55	x comp. fine offset of offset vector l_3	REAL	0
\$TC_CARR56	y comp. fine offset of offset vector l_3	REAL	0
\$TC_CARR57	z comp. fine offset of offset vector l_3	REAL	0
\$TC_CARR58	x comp. fine offset of offset vector l_4	REAL	0
\$TC_CARR59	y comp. fine offset of offset vector l_4	REAL	0
\$TC_CARR60	z comp. fine offset of offset vector l_4	REAL	0
\$TC_CARR64	Offset of fine offset of rotary axis v_1	REAL	0
\$TC_CARR65	Offset of fine offset of rotary axis v_2	REAL	0

*) The system variables \$TC_CARR35 to \$TC_CARR40 are used in the measuring cycles as well as ShopMill and ShopTurn.

Accessing the data of a tool carrier

NC program

Write: \$TC_CARR<n> [<m>] =<Value>

Read: <Value>=\$TC_CARR<n> [<m>]

(<Value> must be a variable of data type REAL)

If the referenced tool carrier is not defined, an alarm is displayed.

OPI interface

The parameters of a tool carrier with orientation capability can be read and written with the system variables \$P_TCANG[<n>].

Setting to zero

Setting all values of all tool carrier data sets to zero is possible by programming:

```
$TC_CARR1[0] = 0
```

Individual tool carrier data sets can be deleted selectively using the predefined procedure DELTC or the PI service _N_DELTCAR.

Data backup

The system variables for orientable tool carriers are saved as part of the general NC data backup.

3.12.4 Configuring tool carrier kinematics with kinematic chain

3.12.4.1 Tool carrier with kinematic chains

Modeling tool carriers with kinematic chains

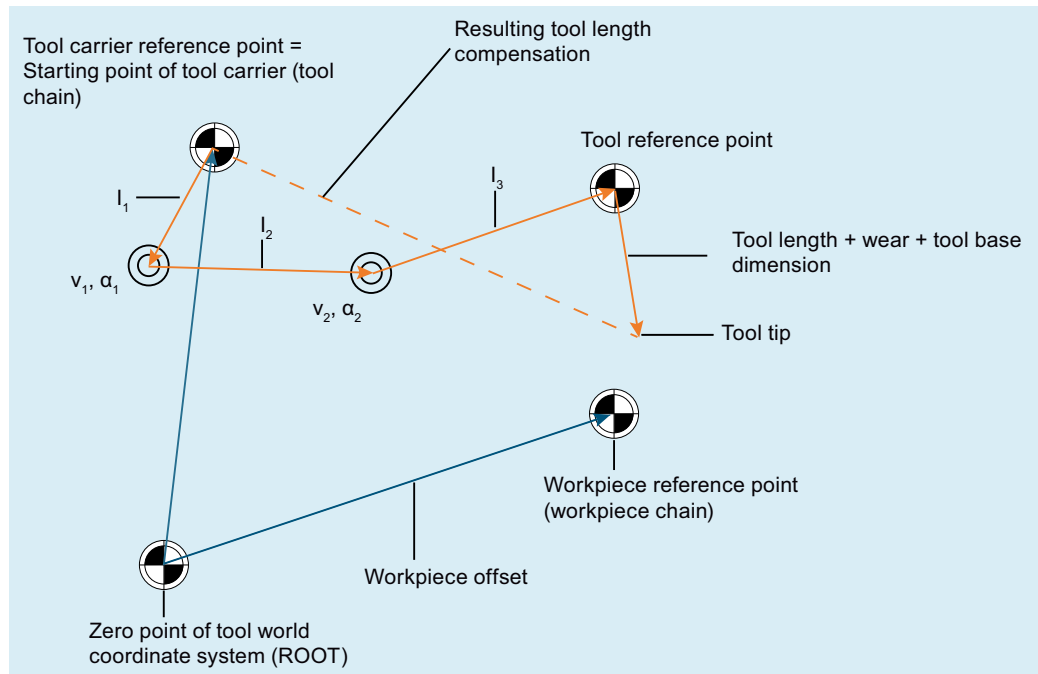
On machines whose structures are defined with kinematic chains, machine parts that act as tool carriers can also be described using kinematic chains. Identical geometry data for kinematic chains and tool carriers can be managed jointly.

Two kinematic chains are required to define the machine kinematics:

- One chain points from machine zero (zero point of world coordination system) to the workpiece reference point (workpiece part).
- The other chain points from machine zero to the tool carrier reference point (tool part).
- You will find information about kinematic chains in the Function Manual "Basic functions".

Three different kinematic types are available for tool carriers, which are parametrized via \$TC_CARR23.

The following graphics show an example of the kinematics of a machine with rotatable tool (type T) and rotatable workpiece (type P) and the resolved kinematics (type M). The resolved kinematics (type M) is a combination of both types.

Example for a rotatable tool

Orange: Kinematic chain of the tool carrier

Blue: Kinematic chain of the machine

Figure 3-33 Tool carrier with rotatable tool

Example of a rotatable workpiece

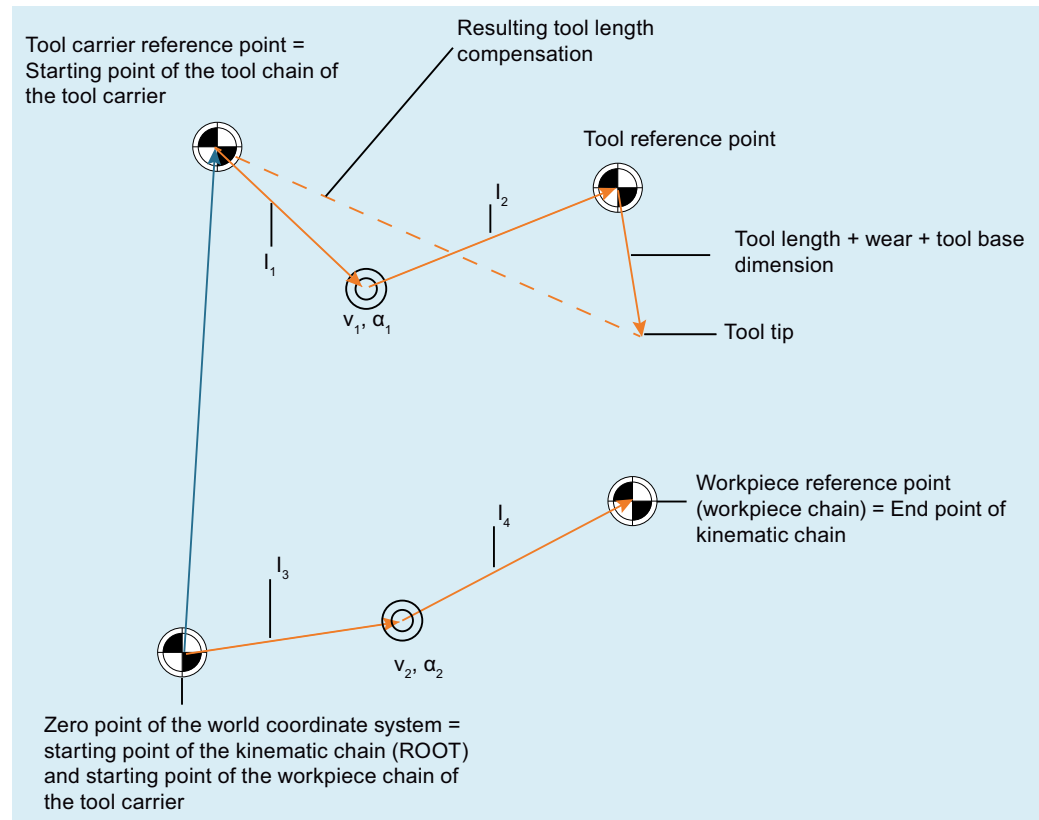
$$\langle \text{Matrix} \rangle = \begin{pmatrix} 0 & \sin 60^\circ & \cos 60^\circ \\ 0 & \cos 60^\circ & -\sin 60^\circ \\ 1 & 0 & 0 \end{pmatrix}$$

Orange: Kinematic chain of the tool carrier

Blue: Kinematic chain of the machine

Figure 3-34 Tool carrier with rotatable workpiece

Example of mixed kinematics



- l_1 and l_2 Offset of the tool chain of the tool carrier
- v_1 Rotary axis of the tool chain
- l_3 and l_4 Offset of the workpiece chain of the tool carrier
- v_2 Rotary axis of the tool chain

Figure 3-35 Mixed kinematics

The kinematics of the tool carrier is defined together with the machine kinematics.

Definition of the tool carrier:

- Definition of the characteristics of tool carrier with `$TC_CARR_KIN_CNTRL`:
 - Close tool carrier via kinematic chain, tool chain, and table chain of the tool carrier.
- The starting point of the kinematic chain of the tool carrier is set via system variables `$TC_CARR_KIN_TOOL_START` and `$TC_CARR_KIN_PART_START` (chain element name).
- Definition of the rotary axes of the tool carrier with `$TC_CARR21` (Page 119) and `$TC_CARR22`
- Definition of the tool carrier type with `$TC_CARR23` (T, P, or M)

Definition of the kinematics of the tool carrier:

The workpiece part of a kinematic chain for a tool carrier can be structured as follows, for example:

- Define the linear axes of the kinematic chain (geometric axes X, Y, and Z).
- Define a kinematic element of type "OFFSET" ($=l_2$) in the X, Y, and Z direction. Corresponds to \$TC_CARR4 - \$TC_CARR6.
- Define a kinematic element of type "AXIS_ROT" ($=v_1$) in the X, Y, and Z direction. Corresponds to \$TC_CARR7 - \$TC_CARR9.
- Define an element of type "OFFSET" ($=l_3$) in the X, Y, and Z direction. Corresponds to \$TC_CARR15 - \$TC_CARR17.
- Define an element of type "AXIS_ROT" ($=v_2$) in the X, Y, and Z direction. Corresponds to \$TC_CARR10 - \$TC_CARR12.
- Define an element of type "OFFSET" ($=l_4$) in the X, Y, and Z direction. Corresponds to \$TC_CARR18 - \$TC_CARR20.
- If required, definition of a rotary axis offset for axes v1 and v2. The offset is entered in the system variable \$NK_A_OFF, which is assigned to the "AXIS_ROT"-type chain element. Corresponds to \$TC_CARR24 - TC_CARR25.

At least one rotary axis must be defined.

Note**Sign for rotary axis offset**

A positive value for the rotary axis offset for the axes of rotation in \$TC_CARR24 or \$TC_CARR25 is entered in the kinematic chain as a negative value.

Selecting the tool carrier:

The tool carrier is selected with the NC program command in the part program:

```
TCARR = <name>
```

Note**Starting point of the kinematic chain**

If a starting point for the kinematic chain of the tool carrier (workpiece or tool) has not been defined via TC_CARR_KIN_PART_START or TC_CARR_KIN_TOOL_START, the ROOT element (world coordinate system) is automatically defined as the starting point of the chains.

\$TC_CARR_KIN_CNTRL

\$TC_CARR_KIN_CNTRL specifies whether the geometry data of a tool carrier are read from kinematic chain elements or from conventional tool carrier data (\$TC_CARRxx). The system variable is bit-coded.

Syntax	Description																
\$TC_CARR_KIN_CNTRL[n]	Controls transfer of the geometry data of a machine model defined with kinematic chains to parameterize a tool carrier. If Bit0 = 1, the data from \$TC_CARRxx are ignored but data are taken from the kinematic chains. It may furthermore be set how the kinematic chain is closed. Data type: INT																
n	Index of the tool carrier; the maximum number of tool carriers can be defined in machine data MM_NUM_TOOL_CARRIER.																
Bit0=1	The tool carrier is parameterized from the kinematic chain elements. The following data of the tool carrier, including the fine offset, are replaced with geometry data from the kinematic chain elements: <table border="1"> <thead> <tr> <th>Conventional tool carrier</th><th>Kinematic chain</th></tr> </thead> <tbody> <tr> <td>\$TC_CARR1 – TC_CARR3 (offset vector l_1)</td><td>X, Y, and Z component of the offset vector l_1</td></tr> <tr> <td>\$TC_CARR4 – TC_CARR6 (offset vector l_2)</td><td>X, Y, and Z component of the offset vector l_2</td></tr> <tr> <td>\$TC_CARR7 – TC_CARR9 (axis of rotation direction v_1)</td><td>X, Y, and Z component of the axis of rotation v_1</td></tr> <tr> <td>\$TC_CARR10 – TC_CARR12 (axis of rotation direction v_2)</td><td>X, Y, and Z component of the axis of rotation v_2</td></tr> <tr> <td>\$TC_CARR15 – TC_CARR17 (offset vector l_3);</td><td>X, Y, and Z component of the offset vector l_3</td></tr> <tr> <td>\$TC_CARR18 – TC_CARR20 (offset vector l_4)</td><td>X, Y, and Z component of the offset vector l_4</td></tr> <tr> <td>\$TC_CARR24 – TC_CARR25 (rotary axis offsets)</td><td></td></tr> </tbody> </table>	Conventional tool carrier	Kinematic chain	\$TC_CARR1 – TC_CARR3 (offset vector l_1)	X, Y, and Z component of the offset vector l_1	\$TC_CARR4 – TC_CARR6 (offset vector l_2)	X, Y, and Z component of the offset vector l_2	\$TC_CARR7 – TC_CARR9 (axis of rotation direction v_1)	X, Y, and Z component of the axis of rotation v_1	\$TC_CARR10 – TC_CARR12 (axis of rotation direction v_2)	X, Y, and Z component of the axis of rotation v_2	\$TC_CARR15 – TC_CARR17 (offset vector l_3);	X, Y, and Z component of the offset vector l_3	\$TC_CARR18 – TC_CARR20 (offset vector l_4)	X, Y, and Z component of the offset vector l_4	\$TC_CARR24 – TC_CARR25 (rotary axis offsets)	
Conventional tool carrier	Kinematic chain																
\$TC_CARR1 – TC_CARR3 (offset vector l_1)	X, Y, and Z component of the offset vector l_1																
\$TC_CARR4 – TC_CARR6 (offset vector l_2)	X, Y, and Z component of the offset vector l_2																
\$TC_CARR7 – TC_CARR9 (axis of rotation direction v_1)	X, Y, and Z component of the axis of rotation v_1																
\$TC_CARR10 – TC_CARR12 (axis of rotation direction v_2)	X, Y, and Z component of the axis of rotation v_2																
\$TC_CARR15 – TC_CARR17 (offset vector l_3);	X, Y, and Z component of the offset vector l_3																
\$TC_CARR18 – TC_CARR20 (offset vector l_4)	X, Y, and Z component of the offset vector l_4																
\$TC_CARR24 – TC_CARR25 (rotary axis offsets)																	
Bit0=0	The tool carrier parameters are assigned with the data from \$TC_CARRxx.																
Bit1=1	The content of offset vector l_1 is changed in such a way that the end point of the tool chain coincides with the zero point of the world coordinate system (chain is closed).																
Bit2=1	The content of offset vector l_4 is changed in such a way that the end point of the workpiece chain coincides with the zero point of the world coordinate system (chain is closed).																

\$TC_CARR_KIN_TOOL_START and \$TC_CARR_KIN_PART_START

System variables \$TC_CARR_KIN_TOOL_START and \$TC_CARR_KIN_PART_START define the starting points of the kinematic chains that are used to parameterize the tool or workpiece part of a tool carrier. If the system variables are empty, the starting point of each kinematic chain is the root element of the kinematic description.

The option of parameterizing only part of the kinematic chain that does not start with the root element makes particular sense and indeed is necessary if the tool carrier is to be

3.12 Tool carriers with orientation capability

deployed in its original function as a "tool extension" and not for the static modeling of the complete transformation kinematics of a machine tool.

Syntax	Description
\$TC_CARR_KIN_TOOL_START[n] = <name>	Defines the name of the starting elements of the tool or workpiece chain.
\$TC_CARR_KIN_PART_START[n] = <name>	Data type: STRING
n	Index of the tool carrier; the maximum number of tool carriers can be defined in machine data MM_NUM_TOOL_CARRIER.

\$TC_CARR_KIN_TOOL_END and \$TC_CARR_KIN_PART_END

System variables \$TC_CARR_KIN_TOOL_END and \$TC_CARR_KIN_PART_END define the end points of the kinematic chains that are used to parameterize the tool or workpiece part of a tool carrier. At least one of these two names must be not equal to the null string.

Syntax	Description
\$TC_CARR_KIN_TOOL_END[n] = <name>	Defines the name of the end elements of the tool or workpiece chain.
\$TC_CARR_KIN_PART_END[n] = <name>	Data type: STRING
n	Index of the tool carrier; the maximum number of tool carriers can be defined in machine data MM_NUM_TOOL_CARRIER.

Simulation with tool carriers capable of orientation (kinematic chain)

Note

Setting machine data for simulation and recording

To ensure that simulation and recording is implemented correctly, set the machine data MD 51226 \$MNS_FUNCTION_MASK_SIM Bit 22.

Constraints and notes

- If the system data of the tool carrier (TC_CARRxx) are replaced with data from kinematic chain elements, the content of system variable \$TC_CARRxx is not changed.
- Tool carriers and transformations that were defined with kinematic chains can be active at the same time. It does not matter how the tool carrier is parametrized (directly from the system data TC_CARR... or from kinematic chains).
- It must be possible to map the kinematic chains that will be used to parameterize a tool carrier onto a tool carrier configuration. If this condition is not met, alarm 14149 is triggered.

Measuring tool carriers from kinematic chains

When measuring the kinematics of a tool carrier and the subsequent offset, the effective offset vectors have to be modified. Because such an offset vector can in principle be formed from any number of partial vectors of the kinematic chain supplying the parameter settings, the chain element to be corrected is not automatically identifiable.

Syntax	Description
\$TC_CARR_CORR_ELEM[n, m] = <string>	Name of the next element in the kinematic chain from which the tool carrier data are to be read. The element is used as the correction element for the offset vector with index m. Data type: STRING
n	Index of the tool carrier; the maximum number of tool carriers can be defined in machine data MM_NUM_TOOL_CARRIER.
m	Index of the correction vector (value range 0...3)

These chain elements can be uniquely identified with system variable \$TC_CARR_CORR_ELEM[m, n]. where m is the index of the tool carrier data set and n is the index of the correction vector (n = 0...3]).

Read out length and direction of the vectors of TCARR

The currently active offset vectors, rotary axis vectors, and rotary axis offsets can be read out via system variables \$PC_TCARR_OFFSET, \$PC_TCARR_AX_VECT, and \$PC_TCARR_AX_OFFSET. For more detailed information about offset vectors, see General information (Page 119).

Syntax	Description
\$PC_TCARR_OFFSET[m,n] = <value>	Current value of the n-th vector component of offset l_m of an active tool carrier. Maximum 4 offset vectors (l1 to l4) are defined for an active tool carrier. Vector components can be read out with this system variable. Data type: DOUBLE
m	Offset vector[m] (where m = 1...4) of the active tool carrier.
n	Vector component n (where $0 \leq n \leq 2$) of the vector that was selected with field index m.

Syntax	Description
\$PC_TCARR_AX_VECT[m,n] = <value>	Current value of the n-th vector component of rotary axis v_m of an active tool carrier. A maximum of 2 rotation axes (v1 to v2) are defined for an active tool carrier. Vector components can be read out with this system variable. Data type: DOUBLE
m	Axis of rotation direction v1, v2, where m = 1...2 of the active tool carrier.
n	Vector component n (where $0 \leq n \leq 3$) of the vector that was selected with m.

3.12 Tool carriers with orientation capability

Syntax	Description
\$PC_TCARR_AX_OFFSET[m] = <value>	Current value of the rotational offset of rotation axes 1 and 2. Maximum 2 rotation axes are defined for an active tool carrier. The positions of the rotary axes in their initial state can be read out with this system variable.
	Data type: DOUBLE
m	Rotary offset where m = 1...2 of the active tool carrier.

Tool carrier with more than two rotary axes

For tool carriers that are defined with more than two rotary axes, the third axis is ignored. It describes a constant angle (knee-point) between the two axes of rotation. Axis v_2 in the example is not taken into account. Instead, offset l_2 is formed.

The relevant rotary axes are defined with the system variable \$TC_CARR_KIN_ROTAX_NAME in the sequence in which they occur in the chain. Up to two rotary axes may be defined.

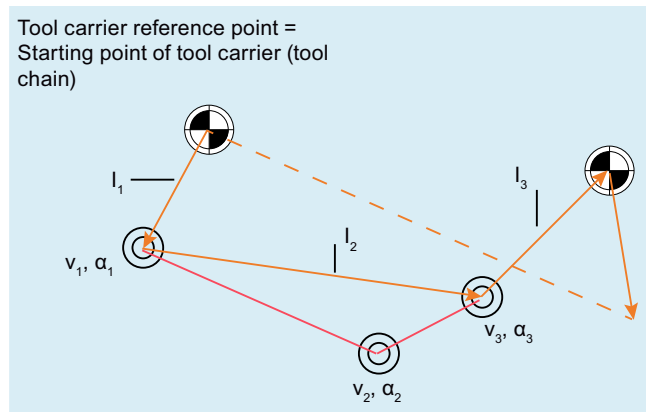


Figure 3-36 Tool carriers with orientation capability with 3 axes of rotation

The rotary axes are not orientation axes, as used for the 5-axis-transformation.

Defining rotation axes (more than two rotary axes)

If a tool carrier comprises more than two rotation axes, this is defined via the relevant axes using the system variable \$TC_CARR_KIN_ROTAX_NAME. The name of the chain element is entered in the system variable in this regard. The maximum two rotary axes are defined in the sequence in which they are defined in the kinematic chain. The entries must begin with the index "0".

Syntax	Description
\$TC_CARR_KIN_ROTAX_NAME[n, m] = <String>	Name of the element in the kinematic chain which is to be defined as the rotary axis.
	Data type: STRING
n	Index of the tool carrier; the maximum number of tool carriers can be defined in machine data MM_NUM_TOOL_CARRIER.
m	Index of rotary axes (value range 0...1)

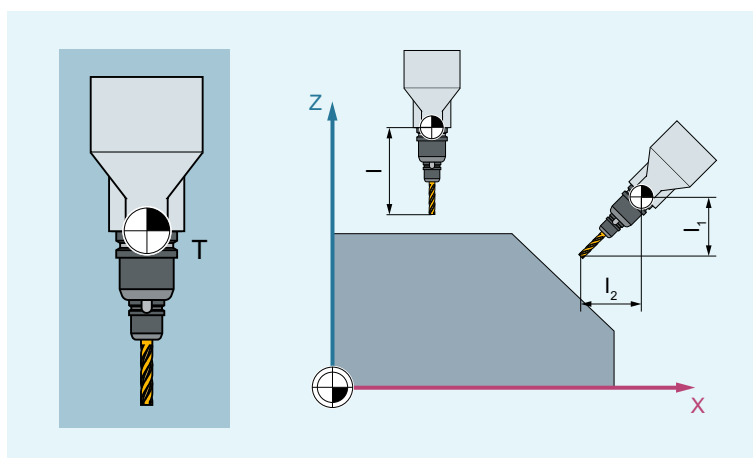
Example

You will find a part program for a tool carrier via a kinematic chain under: Example: Tool carrier with orientation capability via kinematic chain (Page 157).

3.12.5 Programming

3.12.5.1 Tool length compensation for orientable toolholders (TCARR, TCOABS, TCOFR, TCOFRX, TCOFRY, TCOFRZ)

As the spatial orientation of the tool carrier and thus of the tool changes, its tool length components also change:



T Tool carrier reference point

l_1, l_2 Tool length components

After a reset, e.g. through manual setting or change of the tool carrier with a fixed spatial orientation, the tool length components also have to be determined again. The commands of G group 42 "tool carrier" are used for this purpose.

Syntax

TCARR=<m>
TCOABS
TCOFR
TCOFRZ/TCOFRY/TCOFRX

Meaning

Element	Type	Meaning	
TCARR=<m>	Address	Request tool carrier	
		<m>	Number of the tool carrier

3.12 Tool carriers with orientation capability

Element	Type	Meaning
TCOABS	G command	Determine tool length components from the orientation of the current tool carrier
TCOFR	G command	Determine tool length components from the orientation of the active frame
TCOFRZ/TCOFRY/ TCOFRX	G command	The TCOFRX/TCOFRY/TCOFRZ command assumes a tool oriented in the corresponding direction (X/Y/Z) and calculates the setting angles of the orientable tool carrier so that the tool in the active frame is oriented in the same direction.
		TCOFRZ The tool oriented in the Z direction is aligned so that it is also oriented in the Z direction in the active frame.
		TCOFRY The tool oriented in the Y direction is aligned so that it is also oriented in the Y direction in the active frame.
		TCOFRX The tool oriented in the X direction is aligned so that it is also oriented in the X direction in the active frame.

Further information

Tool carrier selection (TCARR)

A tool carrier data block is connected to the tool carrier which describes its geometry. If a tool is active at the time of tool carrier selection, it is assigned to the newly selected tool carrier and the geometry data of the tool carrier take effect immediately. It is not necessary to change or re-program the active tool. When a new tool is activated, it is treated as if it was mounted on the active tool carrier.

With TCARR=0, the active tool carrier is deselected.

Note

The current geometry data of the tool carrier can also be defined in the NC program via the corresponding system variables (see "Access to tool carrier data blocks").

Note

From the point of view of the control, tool carrier numbers <m> and tool numbers <n> can be freely combined. In the real application, however, certain combinations can be ruled out for machining or mechanical reasons. The control does not check whether the combinations make sense.

Access to tool carrier data blocks

The current geometry data of a tool carrier can be accessed from the NC program as follows:

- Write:
\$TC_CARR<n>[<m>]=<Value>
- Read:
<Value>=\$TC_CARR<n>[<m>]
(<Value> must be a variable of data type REAL)

If the referenced tool carrier is not defined, an alarm is displayed.

Setting tool carrier data blocks to zero

All values of all tool carrier data blocks can be deleted from within the part program using one command:

```
$TC_CARR1[0] = 0
```

Individual tool carrier data blocks can be selectively deleted using the predefined procedure DELTC.

Effect of tool carrier selection

A tool carrier becomes effective when both a tool carrier and a tool have been activated. The selection of the tool carrier alone has no effect.

The effect of a tool carrier selection depends on the command from the G group 42 (tool carrier), which was programmed together with TCARR.

Changing the G command from this group causes the tool length components to be recalculated when a tool carrier is active.

Calculating the tool length compensation from the tool carrier orientation (TCOABS)

TCOABS calculates the tool length compensation from the current orientation angles of the tool carrier, stored in the system variables \$TC_CARR13 and \$TC_CARR14. The considered orientation is independent of the orientation of the active frame.

For the definition of the tool carrier kinematics with system variables, see Chapter "Tool carrier kinematics" in the NC Programming Manual.

Determining the tool length components from the orientation of the active frame (TCOFR)

With TCOFR, the orientation angles of the tool carrier are determined from the orientation of the active frame. The values stored in the tool carrier data are not changed, however. These are also used to resolve the ambiguity that can arise when calculating the rotation angles from a frame (see following paragraph).

Ambiguities

With two axes, a particular tool orientation defined by the frame can generally be set with **two** different rotation angle pairs. Of these two possible positions, the control selects the one in which the rotation angles are as close as possible to the programmed rotation angles.

Note

In cases where ambiguity may occur, it is usually necessary to store the angles to be expected from the frame in the tool carrier data.

Tool orientation in the active frame (TCOFRZ/TCOFRY/TCOFRX)

The TCOFRX/TCOFRY/TCOFRZ command assumes an orientation in the X/Y/Z direction for calculating the tool carrier's setting angles, regardless of the active orientation. The selection of the machining plane (G17/G18/G19) does not affect the orientation in the active frame. However, it has an influence on the calculation of the tool length compensation.

Frame change

The user can change the frame after selecting the tool. This does not have any effect on the tool length compensation components.

The rotation angles stored in the tool carrier data are not affected by the rotation angles defined with frames. When changing from TCOFR to TCOABS, the original (programmed) angles of rotation in the tool carrier data is reactivated.

Recalculation of the tool length compensation

To recalculate the tool length compensation after a frame change, the tool must be selected again. When the tool length compensation is calculated, the angle of rotation of the tool carrier is calculated in an intermediate step. Since tool carriers with two axes of rotation generally have two rotation angle pairs with which the tool orientation can be adapted to the active frame, the rotation angle values stored in the tool carrier data must at least approximately correspond to the mechanically set rotation angles.

Note

In virtually any case where ambiguities may arise, it is necessary to store the approximate angle expected from the frame in the tool carrier data.

Note

The tool orientation must be manually adapted to the active frame.

Note

The control system cannot check the angles of rotation calculated via the frame orientation for adjustability on the machine. If the machine is designed in such a way that the axes of rotation of the tool carrier cannot reach the tool orientation calculated by the frame orientation, an alarm is output.

Note

The combination of tool precision compensation and the functions for tool length offset on movable tool carriers is not permissible. If both functions are called simultaneously, an error message is issued.

Note

With TOFRAME it is possible to define a frame based on the orientation direction of the selected tool carrier.

Note

When orientation transformation is active (3, 4 or 5-axis transformation), it is possible to select a tool carrier with an orientation deviating from the zero position without causing output of an alarm.

Note

With an active frame without rotation, orientation with TCOFRX/TCOFRY/TCOFRZ leads to a trivial solution, since the tool orientation already points in the X/Y/Z direction.

Orientation in the Z direction (TOFRAME)

With the TOFRAME command of the G group 53 (tool-related frame rotation), it is possible to define a frame whose Z direction is aligned in parallel to the orientation direction of the selected tool carrier.

If no tool carrier or a tool carrier without orientation change is active, then the Z direction in the new frame is as follows:

- The same as the old Z direction with G17.
- The same as the old Y direction with G18.
- The same as the old X direction with G19.

Transfer parameter from standard and measuring cycles

For the transfer parameter of standard and measuring cycles, the following defined value ranges apply.

For angle values, the following value ranges are defined:

Rotation around ...	Value range
1st geometry axis	$-180^\circ \leq \text{angle} \leq +180^\circ$
2nd geometry axis	$-90^\circ \leq \text{angle} \leq +90^\circ$
3rd geometry axis	$-180^\circ \leq \text{angle} \leq +180^\circ$

Note

When transferring angular values to a standard or measuring cycle, the following should be carefully observed:

Values smaller than the computational accuracy of the NC must be rounded to zero!

The calculation resolution of the NC for angular positions is defined in the machine data:

MD10210 \$MN_INT_INCR_PER_DEG

3.12.5.2 Modifying the orientable tool carrier according to the machine measurement (CORRTC)

Measured kinematic chain elements of a tool carrier can be written to special correction elements with the CORRTC function.

3.12 Tool carriers with orientation capability

Note

The correction values written with the CORRTC function are not immediately effective in the tool carrier. The correction values only become active after deselecting the tool carrier, NEWCONF and selecting the tool carrier.

Syntax

```
<_Corr_Status> = CORRTC(<_Corr_Vect>, <_Corr_Index>, <_Corr_Mode>,
[ <_No_Alarm>])
```

Meaning

CORRTC:	Function call			
<_Corr_Status>:	Function return value			
	Data type:	INT		
	Values:	0	The function was executed without an error.	
		1	No tool carrier is active.	
		2	The active tool carrier was not defined with kinematic chains.	
		10	The <_Corr_Index> call parameter is negative.	
		11	The <_Corr_Mode> call parameter is negative.	
		12	Invalid reference to a section of a subchain (_CORR_INDEX).	
		13	No correction element has been defined in the section referred to by the _CORR_INDEX parameter (\$TC_CARR_CORR_ELEM).	
		20	The hundreds position of <_CORR_MODE> is invalid. Only the values 0 and 1 are permissible.	
		21	The thousands position of <_CORR_MODE> is invalid. Only the values 0 and 1 are permissible.	
		30	For the correction of an offset vector, the deviation from the current value in at least one coordinate is greater than the maximum value specified by the setting data SD41612 \$SN_CORR_TRAFO_LIN_MAX.	
	31	The attempt to write a system variable was rejected because of missing write rights.		
<_Corr_Vect>:	Correction vector			
	The content of the correction vector is defined by the following parameters <_Corr_Index> and <_Corr_Mode>.			
	If <_Corr_Status> = 30, the content of the vector is overwritten (see above).			
	Data type:	REAL		
<_Corr_Index>:	Designates the section for which the direction vector of the correction element is to be corrected.			
	Data type:	INT		

<_Corr_Mode>:	Correction mode		
	Data type:	INT	
	The <Corr_Index> parameter is decimal coded (unit to thousands position):		
	Unit position:	Reserved	
	Tens position:	Specifies how the correction element to which the content of <_Corr_Index> refers, is to be modified.	
		xx0x	The correction vector is written immediately to the correction element. This variant can be used to immediately write the correction element without the index <n> of the relevant system data (\$NK_OFF_DIR[<n>, ...]) having to be known.
		xx1x	As 0, but with the difference that the transferred correction value is interpreted in world coordinates. A difference between variants 0 and 1 can always occur when the kinematic chain in the initial state (positions of all rotary axes equal to 0) contains other rotations.
		xx2x	As 1, but with the difference that the correction value refers to the entire section, i.e. a value is entered in the correction element so that the entire section reaches the length defined by the correction value.
	Hundreds position:	Specifies how the content of the <_Corr_Vect> parameter is to be interpreted.	
		x0xx	The transferred correction vector <_Corr_Vect> contains the entire new length of the correction element or the section to which the <_Corr_Index> in conjunction with the tens position of <_Corr_Mode> refers (absolute correction).
		x1xx	The transferred correction vector <_Corr_Vect> only contains the difference compared to the current length of the correction element or the section to which the <_Corr_Index> in conjunction with the tens position of <_Corr_Mode> refers (incremental correction).
	Thousands position:	Determines whether or not the maximum permissible correction is to be limited by the setting data \$SN_CORR_TOCARR_LIN_MAX.	
		0xxx	Threshold value monitoring is active.
		1xxx	The threshold value monitoring is suppressed.
<_No_Alarm>:	Response in the event of an error (return value > 0) (optional)		
	Data type:	BOOL	
	Value:	FALSE (default)	In the event of an error, the program execution is stopped and an alarm displayed.
		TRUE	In the event of an error, the program processing is not stopped and no alarm is displayed. Application: User-specific response corresponding to the return value

Further information about CORRTC

The kinematic structure of a tool carrier is described by one (type T and type P) or two (type M) kinematic chains (subchains), which start from the associated reference point, machine zero or tool carrier reference point). One of the two chains, the tool chain, ends at the reference point of the tool, the other chain, the workpiece chain ends in the zero point of the basic coordinate system.

The CORRTC function writes axis directions for machines with tool carriers in special correction elements. A kinematic chain is described, for example, with elements of the type OFFSET, which are defined via \$NK_TYPE.

CORRTC works with sections

The two subchains can each be divided into a maximum of four sections:

- Section 1 begins at the starting point of the chain and ends at the first rotary axis.
- Section 2 is the section between rotary axis 1 and rotary axis 2.
- Section 3 is the section between rotary axis 2 and the end of the chain.

The following figure shows an orientable tool carrier with 2 rotary axes.

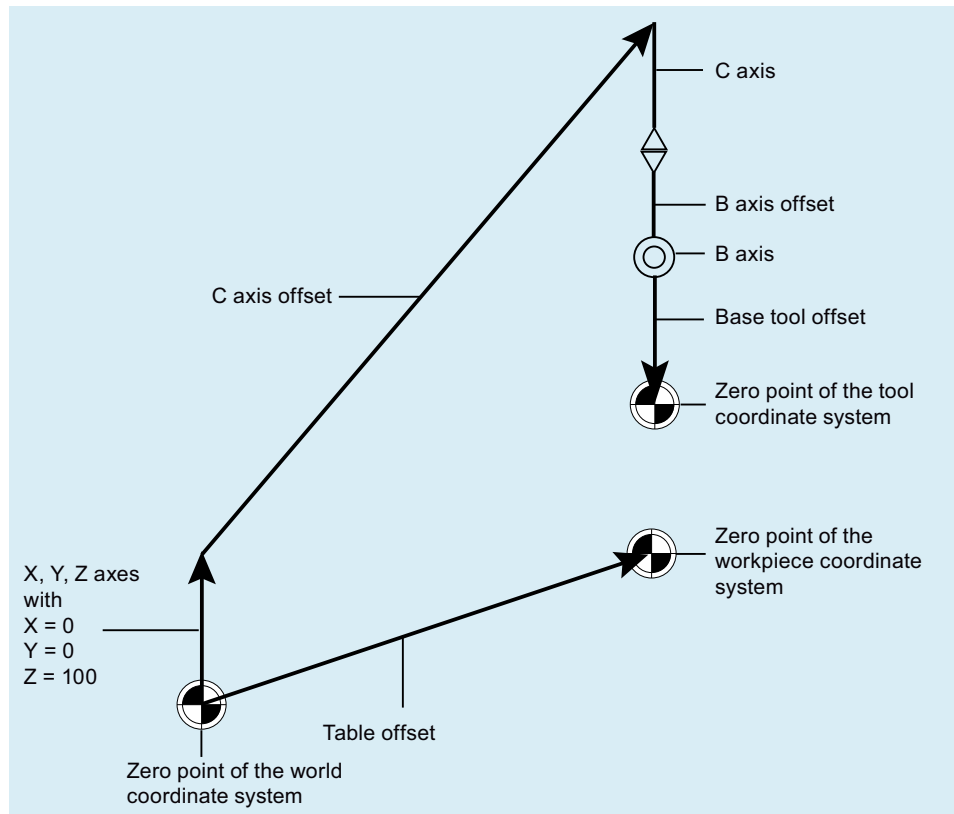


Figure 3-37 CORRTC example

The sections are clearly defined: If the kinematic subchain is executed from its starting point to its end point, then the first section has the index 1, next has index 2, and so on.

Correction elements

A reference can be made to a constant kinematic chain element (chain element of the type \$NK_TYPE[<n>] = "OFFSET") in each of these sections with the \$TC_CARR_CORR_ELEM [, 0 ... 3] system variables. The correction values determined during the machine measurement are written to the so designated elements with the CORRTC function.

The sequence of references in \$TC_CARR_CORR_ELEM[m, 0 ... 3] must correspond to the sections described above, that is there can only be one chain element in \$TC_CARR_CORR_ELEM[m, 0] which belongs to the offset vector I1, etc.

The reference value is always the corresponding value effective in the tool carrier active when CORRTC is called. After selection of the tool carrier, changed contents of the stored kinematic data have no effect on the method of operation of the CORRTC function.

3.12.6 Examples

3.12.6.1 Example: Toolholder with orientation capability

Requirement

The following example uses a toolholder, which is described fully by a rotation about the Y axis. It is therefore sufficient to enter only one value to define the rotary axis (block N20).

Blocks N50 to N70 describe an end mill with radius 5 mm and length 20 mm.

Block N90 defines a rotation of 37 degrees about the Y axis.

Block N120 activates the tool radius compensation and all settings are made to describe the compensation in the following blocks with a rotation of 37 degrees about the Y axis.

```

N10                                ; Definition of toolholder 1
N20 $TC_CARR8[1] = 1               ; Component of first rotary axis in Y di-
                                   ; rection
N30
N40                                ; Definition of tool offset memory T1/D1
N50 $TC_DP1[1,1] = 120             ; End mill
N60 $TC_DP3[1,1] = 20              ; Length 1
N70 $TC_DP6[1,1] = 5               ; Radius
N80
N90 ROT Y37                        ; 37-degree rotation about y axis
N100
N110 X0 Y0 Z0 F10000
N120 G42 CUT2DF TCOFR TCARR = 1 T1 D1 X10
N130 X40
N140 Y40
N150 X0
N160 Y0
N170 M30

```

3.12.6.2 Example of toolholder with orientation capability with rotary table

Use of the MOV T command

For use of the MOV T command it is assumed that the program is running on a 5axis machine, on which the tool rotates about the Y axis in case of a rotation of the B axis:

```

N10 TRAORI()
N20 X0 X0 Z0 B45 F2000           ; Setting the tool orientation
N30 MOV T=-10                     ; Infeed movement 10 mm in tool direc-
                                ; tion
                                ; (under 45 degrees in the Y-Z plane)
N40 MOV T=AC(20)                 ; Retraction in tool direction at dis-
                                ; tance of
                                ; 20 mm from the zero point

```

Machine with rotary table

Complete definition for the use of a toolholder with orientation capability with rotary table:

```

N10 $TC_DP1[1,1]= 120
N20 $TC_DP3[1,1]= 13             ; Tool length 13 mm

; Definition of toolholder 1:

N30 $TC_CARR1[1] = 0             ; X component of 1st offset vector
N40 $TC_CARR2[1] = 0             ; Y component of 1st offset vector
N50 $TC_CARR3[1] = 0             ; Z component of 1st offset vector

N60 $TC_CARR4[1] = 0             ; X component of 2nd offset vector
N70 $TC_CARR5[1] = 0             ; Y component of 2nd offset vector
N80 $TC_CARR6[1] = -15           ; Z component of 2nd offset vector

N90 $TC_CARR7[1] = 1             ; X components of 1st axis
N100 $TC_CARR8[1] = 0            ; Y components of 1st axis
N110 $TC_CARR9[1] = 0            ; Z components of 1st axis

N120 $TC_CARR10[1] = 0           ; X components of 2nd axis
N130 $TC_CARR11[1] = 1           ; Y components of 2nd axis
N140 $TC_CARR12[1] = 0           ; Z components of 2nd axis

N150 $TC_CARR13[1] = 30          ; Angle of rotation of 1st axis
N160 $TC_CARR14[1] = -30         ; Angle of rotation of 2nd axis

N170 $TC_CARR15[1] = 0           ; X components of 3rd offset vector
N180 $TC_CARR16[1] = 0           ; Y component of 3rd offset vector

```


3.12 Tool carriers with orientation capability

```

N190 $TC_CARR17[1] = 0 ; Z component of 3rd offset vector

N200 $TC_CARR18[1] = 0 ; X component of 4th offset vector
N210 $TC_CARR19[1] = 0 ; Y component of 4th offset vector
N220 $TC_CARR20[1] = 15 ; Z component of 4th offset vector

N230 $TC_CARR21[1] = A ; Reference for 1st axis
N240 $TC_CARR22[1] = B ; Reference for 2nd axis
N250 $TC_CARR23[1] = "P" ; Toolholder type

N260 X0 Y0 Z0 A0 B45 F2000
N270 TCARR=1 X0 Y10 Z0 T1 TCOABS
N280 PAROT
N290 X0 Y0 Z0
N300 G18 MOVT=AC(20)
N310 G17 X10 Y0 Z0
N320 MOVT=-10
N330 PAROTOF
N340 TCOFR
N350 X10 Y10 Z-13 A0 B0
N360 ROTS X-45 Y45
N370 X20 Y0 Z0 D0
N380 Y20
N390 X0 Y0 Z20
N400 M30

```

The definition of the toolholder with orientation capability is given in full. The components which contain the value 0 need not actually be given, as they are preset to zero in any case.

The toolholder is activated in N270.

As `$TC_CARR21` and `$TC_CARR22` refer to the machine axes A and B and TCOABS is active, the values in `$TC_CARR13` and `$TC_CARR14` are ignored, i.e. the axis position A0 B45 is used for the rotation.

The rotation of the 4th offset vector (length 15 mm in Z direction) around the B axis causes an offsetting of the zero point by X10.607 [= 15 * sin(45)] and Z-4.393 [= -15 * (1. - cos(45))]. This zero offset is taken into account by an automatically written basic or system frame so that the position X10.607 Y10.000 Z8.607 is approached. In the Z direction the tool selection leads to an additional offset of 13 mm; the Y component is not affected by the table rotation.

N280 defines a rotation in accordance with the rotation of the table of the toolholder with orientation capability. The new X direction thus points in the direction of the bisecting line in the 4th quadrant, the new Z axis in the direction of the bisecting line in the 1st quadrant.

The zero point is approached in N290, i.e. the machine position X10.607 Y0 Z-4.393, since the position of the zero point is not changed by the rotation.

3.12 Tool carriers with orientation capability

N300 traverses in Y to the position Y33.000, since G18 is active and the Y component is not affected by the active frame. The X and Z positions remain unchanged.

The position X17.678 Y0 Z1.536 is approached in N310.

N320 changes only the Z position to the value -8.464 as a result of the MOV_T command. As only the table can be rotated, the tool orientation remains unchanged parallel to the machine Z direction, even if the Z direction of the active frame is rotated by 45 degrees.

N330 deletes the basic or system frame; thus the frame definition from N280 is undone.

In N340, TCOFR specifies that the toolholder with orientation capability is to be aligned according to the active frame. Since a rotation is no longer active in N330 due to the PAROTOF command, the default setting is applied. The frame offset becomes 0.

N350 thus approaches the position X10 X10 Z0 (= Z-13 + tool length). Notice: Through the simultaneous programming of both rotary axes A and B the actual position of the toolholder with orientation capability is made to match that used in N340. The position approached by the three linear axes is dependent on this position, however.

In N360, solid angles are used to define a plane whose intersecting lines in the XZ and in the YZ plane each form an angle of +45 degrees or -45 degrees with the X or Y axis. The plane defined in such a way therefore has the following position: The surface normal points towards the solid diagonals.

N370 traverses to the position X20 Y0 Z0 in the new coordinate system. Since the tool is deselected with D0 at the same time, there is no longer an additional offset in Z. Since the new X axis lies in the old X-Z plane, this block reaches the machine position X14.142 Y0 Z-14.142.

N380 only traverses on the Y axis in the rotated coordinate system. This leads to a motion of all three machine axes. The machine position is X5.977 Y16.330 Z-22.307.

N390 approaches a point on the new Z axis. Relative to the machine axes this is thus on the solid diagonal. All three axes thus reach the position 11.547.

3.12.6.3 Calculation of compensation values on a location-specific and workpiece-specific basis

Tool with adapter

A tool with adapter and tool carrier with orientation capability is defined in the following program example. To simplify the overview, only length L1 is different from zero for the additive and insert offsets and for the adapter in case of the tool itself. The offset vectors of the tool carrier with orientation capability are all zero.

Program code	Comment
N10 \$TC_TP2[1]="MillingTool"	; Name of identifier
N20 \$TC_TP7[1]=9	; Location type
N30 \$TC_TP8[1]=2	; Status: Enabled and not blocked
; D corr. D=1	

3.12 Tool carriers with orientation capability

Program code	Comment
N40 \$TC_DP1[1,1]=120	; Tool type - milling
N50 \$TC_DP3[1,1]=	; Length offset vector
N60 \$TC_DP12[1,1]=	; Wear
N70 \$TC_SCP13[1,1]=0.1	; Sum offset DL=1
N80 \$TC_ECP13[1,1]=0.01	; Insert offset DL=1
N90 \$TC_ADPTT[1]=5	; Adapter transformation
N100 \$TC_ADPT1[1]=0.001	; Adapter dimension
; Magazine data	
N110 \$TC_MAP1[1]=3	; Magazine type: Turret
N120 \$TC_MAP2[1]="Turret"	; Name of a magazine
N130 \$TC_MAP3[1]=17	; Status of magazine
N140 \$TC_MAP6[1]=1	; Dimension - line
N150 \$TC_MAP7[1]=2	; Dimension - column -> 2 positions
N160 \$TC_MPP1[1,1]=1	; Location type
N170 \$TC_MPP2[1,1]=9	; Location type
N180 \$TC_MPP4[1,1]=2	; Location status
N190 \$TC_MPP7[1,1]=1	; Bring adapter into position
N200 \$TC_MPP6[1,1]=1	; T number "MillingTool"
N210 \$TC_MAP1[9999]= 7	; Magazine type: Buffer
N220 \$TC_MAP2[9999]="Buffer"	; Name of a magazine
N230 \$TC_MAP3[9999]=17	; Status of magazine
N240 \$TC_MAP6[9999]=1	; Dimension - line
N250 \$TC_MAP7[9999]=1	; Dimension - column -> 1 position
N260 \$TC_MPP1[9999,1]=2	; Location type
N270 \$TC_MPP2[9999,1]=9	; Location type
N280 \$TC_MPP4[9999,1]=2	; Location status
N290 \$TC_MPP5[9999,1]=1	; Spindle no. 1
N300 \$TC_MDP2[1,1]=0	; Distance from spindle to mag. 1
; Definition of tool carrier 1	
N310 \$TC_CARR10[1]=1	; Component 2. Axis of rotation in the X direction
N320 \$TC_CARR14[1]=45	; Angle of rotation of 2nd axis
N330 \$TC_CARR23[1]="T"	; Tool mode
N340 Stopre	
N350 \$SC_WEAR_TRANSFORM='B101'	
N360 T0 D0 DL=0	
N370 ROT X30	
N380 G90 G1 G17 F10000 X0 Y0 Z0	
N390 T="MillingTool" X0 Y0 Z0 TOWSTD	; X 0.000 Y11.110 Z 0.001

3.12 Tool carriers with orientation capability

Program code	Comment
N400 T="MillingTool" X0 Y0 Z0 TOWMCS	; X 0.000 Y10.100 Z 1.011
N410 T="MillingTool" X0 Y0 Z0 TOWWCS	; X 0.000 Y 9.595 Z 0.876
N420 TCARR=1 X0 Y0 Z0	; X 0.000 Y 6.636 Z 8.017
N430 G18 X0 Y0 Z0	; X10.100 Y-0.504 Z 0.876
N440 M30	

Explanations regarding the example above

Starting at block N390, various methods are used to approach position X0 Y0 Z0. The machine positions reached are specified in the blocks in comments. After the program a description is given of how the positions were reached.

N390: The adapter transformation 5 (block N90) transforms length L1 into length L2. Only the actual adapter dimension is not subject to this transformation. The Y value (L2 with G17) results from the sum of the tool length (10), tool wear (1), sum offset (0.1), and insert offset (0.01). The adapter dimension (0.001) is in Z (L1).

N400: In block N350, bits 0 and 2 are enabled in setting data:

SD42935 \$SC_WEAR_TRANSFORM (transformations for tool components)

This means that the tool wear and the insert offset are not subject to the adapter transformation because of TOWMCS in block N400. The sum of these two compensations is 1.01. The Z position is, therefore, increased by this amount and the Y position is reduced by this amount compared with block N390.

TOWWCS is active in N410. The sum of the tool wear and the insert offset is thus effective in the active workpiece coordinate system. In block N370, a rotation through 30 degrees is activated around the X axis. The original compensation value of 1.01 in the Z direction thus yields a new Z component of 0.875 ($= 1.01 * \cos(30)$) and a new Y component of -0.505 ($= 1.01 * \sin(30)$). This yields the dimension specified in the program comment when added to the sum of the tool length, sum offset and adapter dimension produced in block N390.

In addition, a tool carrier with orientation capability is activated in block N420. This executes a rotation through 45 degrees about the X axis (see N310 - N330). Since all offset vectors of the tool carrier are zero, there is no additional work offset. The tool carrier with orientation capability acts on the sum of the tool length, sum offset and adapter dimension. The resulting vector component is X0 Y7.141 Z7.142. To this, as in block N410, the sum of tool wear and insert offset evaluated in WCS is added.

G18 is activated in N430. The components of the tool length sum, sum offset and adapter dimension are interchanged accordingly. The tool carrier with orientation capability continues to act on this new vector (rotation through 45 degrees about X axis). The resulting vector component thereby is X10.100 Y0.0071 Z0.0071. The vector formed from tool wear and insert offset (X0 Y-0.505 Z0.875) is not affected by the change of plane. The sum of the two vectors yields the dimension specified in the comment in N430.

3.12.6.4 Example: Tool carrier with orientation capability via kinematic chain

Example of a tool carrier with mixed kinematics (rotatable tool and table)

The example shows a tool carrier that is defined via a kinematic chain. The chain is automatically closed at the endpoints of the two separate chains (tool and table).

Program code	Comment
	; Definition tool carrier 1
N1070 _MIXED_1_S:	
N1080 IF (DELOBJ("KIN_CHAIN_ELEM") <> 0)	
N1080 IF (DELOBJ("KIN_CHAIN_ELEM") <> 0)	
N1090 SETAL(62000)	
N1100 ENDIF	
N1110 _KIE_CNT = 0	
N1120 _CARR_CNT = 1	
N1130 TCARR=0	; Deactivate current tool carrier.
N1140 \$TC_CARR1[0] = 0	; Delete all tool carrier data.
N1150 \$TC_CARR_KIN_CNTRL[_CARR_CNT]=7	; Tool carrier via kinematic chain
	; + Close tool and part chain.
N1160 \$TC_CARR_KIN_TOOL_START[_CARR_CNT]="Y_AXIS"	; Start element of the tool chain.
N1170 \$TC_CARR_KIN_TOOL_END[_CARR_CNT]="B_OFFSET_CORR"	; End element of the tool chain.
N1180 \$TC_CARR_KIN_PART_START[_CARR_CNT]="X_AXIS"	; Start element of the part chain.
N1190 \$TC_CARR_KIN_PART_END[_CARR_CNT]="END_PART_CHAIN"	; End element of the part chain.
N1200 \$TC_CARR21[_CARR_CNT]=B	
N1210 \$TC_CARR22[_CARR_CNT]=C	
N1220 \$TC_CARR23[_CARR_CNT]="M"	
N1230 \$NK_NAME[_KIE_CNT] = "ROOT"	
N1240 \$NK_PARALLEL[_KIE_CNT]="X_AXIS"	
N1250 \$NK_TYPE[_KIE_CNT] = "OFFSET"	
N1260 \$NK_NEXT[_KIE_CNT] = "Y_AXIS"	
N1270 _KIE_CNT=_KIE_CNT+1	
	;***** Start tool chain *****
N1280 \$NK_NAME[_KIE_CNT] = "Y_AXIS"	
N1290 \$NK_TYPE[_KIE_CNT] = "AXIS_LIN"	
N1300 \$NK_OFF_DIR[_KIE_CNT,1] = 1	
N1310 \$NK_AXIS[_KIE_CNT] = "Y1"	
N1320 \$NK_NEXT[_KIE_CNT] = "Z_AXIS"	
N1330 _KIE_CNT=_KIE_CNT+1	
N1340 \$NK_NAME[_KIE_CNT] = "Z_AXIS"	
N1350 \$NK_TYPE[_KIE_CNT] = "AXIS_LIN"	
N1360 \$NK_OFF_DIR[_KIE_CNT,2] = 1	
N1370 \$NK_AXIS[_KIE_CNT] = "Z1"	
N1380 \$NK_NEXT[_KIE_CNT] = "CLOSE_HEAD"	
N1390 _KIE_CNT=_KIE_CNT+1	
N1400 \$NK_NAME[_KIE_CNT] = "CLOSE_HEAD"	

3.12 Tool carriers with orientation capability

Program code	Comment
N1410 \$NK_TYPE[_KIE_CNT] = "OFFSET"	
N1420 \$NK_OFF_DIR[_KIE_CNT,0] = 0	
N1430 \$NK_OFF_DIR[_KIE_CNT,1] = 0	
N1440 \$NK_OFF_DIR[_KIE_CNT,2] = 0	
N1450 \$NK_NEXT[_KIE_CNT] = "B_AXIS"	
N1460 _KIE_CNT=_KIE_CNT+1	
N1470 \$NK_NAME[_KIE_CNT] = "B_AXIS"	
N1480 \$NK_TYPE[_KIE_CNT] = "AXIS_ROT"	
N1490 \$NK_OFF_DIR[_KIE_CNT,1] = 1	
N1500 \$NK_OFF_DIR[_KIE_CNT,2] = 1	
N1520 \$NK_A_OFF[_KIE_CNT] = -27	
N1540 \$NK_AXIS[_KIE_CNT] = "B1"	
N1550 \$NK_NEXT[_KIE_CNT] = "B_OFFSET"	
N1560 _KIE_CNT=_KIE_CNT+1	
N1570 \$NK_NAME[_KIE_CNT] = "B_OFFSET"	
N1580 \$NK_TYPE[_KIE_CNT] = "OFFSET"	
N1590 \$NK_OFF_DIR[_KIE_CNT,2] = -30.6	
N1600 \$NK_NEXT[_KIE_CNT] = "B_OFFSET_CORR"	
N1610 _KIE_CNT=_KIE_CNT+1	
N1620 \$NK_NAME[_KIE_CNT] = "B_OFFSET_CORR"	
N1630 \$NK_TYPE[_KIE_CNT] = "OFFSET"	
N1640 \$NK_NEXT[_KIE_CNT] = ""	
N1650 _KIE_CNT=_KIE_CNT+1	
	;***** End tool chain *****
	;***** Start part chain *****
N1660 \$NK_NAME[_KIE_CNT] = "X_AXIS"	
N1670 \$NK_TYPE[_KIE_CNT] = "AXIS_LIN"	
N1680 \$NK_OFF_DIR[_KIE_CNT,0] = 1	
N1690 \$NK_AXIS[_KIE_CNT] = "X1"	
N1700 \$NK_NEXT[_KIE_CNT] = "C_OFFSET"	
N1710 _KIE_CNT=_KIE_CNT+1	
N1720 \$NK_NAME[_KIE_CNT] = "C_OFFSET"	
N1730 \$NK_TYPE[_KIE_CNT] = "OFFSET"	
N1740 \$NK_OFF_DIR[_KIE_CNT,0] = 300	
N1750 \$NK_OFF_DIR[_KIE_CNT,1] = 150	
N1760 \$NK_NEXT[_KIE_CNT] = "C_OFFSET_CORR"	
N1770 _KIE_CNT=_KIE_CNT+1N1780 \$NK_NAME[_KIE_CNT] = "C_OFF-	
SET_CORR"	
N1790 \$NK_TYPE[_KIE_CNT] = "OFFSET"	
N1800 \$NK_OFF_DIR[_KIE_CNT,0] = 0	
N1810 \$NK_OFF_DIR[_KIE_CNT,1] = 0	
N1820 \$NK_OFF_DIR[_KIE_CNT,2] = 0	
N1830 \$NK_NEXT[_KIE_CNT] = "C_AXIS"	
N1840 _KIE_CNT=_KIE_CNT+1	

Program code	Comment
N1850 \$NK_NAME[_KIE_CNT] = "C_AXIS"	
N1860 \$NK_TYPE[_KIE_CNT] = "AXIS_ROT"	
N1870 \$NK_OFF_DIR[_KIE_CNT,2] = -1	
N1890 \$NK_A_OFF[_KIE_CNT] = -58	
N1910 \$NK_AXIS[_KIE_CNT] = "C1"	
N1920 \$NK_NEXT[_KIE_CNT] = "CLOSE_PART"	
N1930 _KIE_CNT=_KIE_CNT+1	
N1940 \$NK_NAME[_KIE_CNT] = "CLOSE_PART"	
N1950 \$NK_TYPE[_KIE_CNT] = "OFFSET"	
N1960 \$NK_OFF_DIR[_KIE_CNT,0] = 0	
N1970 \$NK_OFF_DIR[_KIE_CNT,1] = 0	
N1980 \$NK_OFF_DIR[_KIE_CNT,2] = 0	
N1990 \$NK_NEXT[_KIE_CNT] = "END_PART_CHAIN"	
N2000 _KIE_CNT=_KIE_CNT+1	
N2010 \$NK_NAME[_KIE_CNT] = "END_PART_CHAIN"	
N2020 \$NK_TYPE[_KIE_CNT] = "OFFSET"	
	;***** End part chain *****

3.12.7 Inclined surface machining with 3 + 2 axes

Description of function

Inclined machining with 3 + 2 axes describes an extension of the concept of toolholders with orientation capability and applies this concept to machines with a rotary table, on which the orientation of tool and table can be changed simultaneously.

The "Inclined machining with 3 + 2 axes" function is used to machine surfaces with any rotation with reference to the main planes X/Y (G17), Z/X (G18) and Y/Z (G19).

It is possible to produce any orientation of the tool relative to the workpiece by rotating either the tool, the workpiece or both the tool and the workpiece.

The software automatically calculates the compensation movements resulting from the tool lengths, lever arms and rotation axis angles. It is always assumed that the required orientation is set first and not modified during a machining process such as pocket milling on an inclined plane.

Furthermore, the following 3 functions are described, which are required for oblique machining:

- **Position programming** in the direction of the tool orientation independent of an active frame
- Definition of a **frame rotation** by specifying the solid angle
- Definition of the **component of rotation in tool direction** in the programmed frame while maintaining the remaining frame components

Demarcation to 5-axis transformation

If the required functionality specifies that the TCP (Tool Center Point) does not vary in the event of reorientation with reference to the workpiece, even during interpolation, the 5axis software is required.

For more explanations on 5-axis transformations, see:

Further information

Function Manual Transformations; Multi-axis transformation

Specification of the toolholder with orientation capability

The toolholder with orientation capability is represented by a general 5axis kinematic sequence described by a data block in the tool offset memory with a total of 33 REAL values. For toolholders that have two rotary axes for setting the orientation (e.g. a millhead), 31 of these values are constant.

In the current SW version, a data block in the tool offset memory is described with a total of 47 REAL values. For toolholders that have two rotary axes for setting the orientation, 45 of these values are constant.

The remaining two values are variable and are used to specify the orientation. The constant values describe offsets and directions and setting options for the rotary axes; the variable values describe the angles of the rotary axes.

3.12.8 Machine with rotary work table

System variables

To date, the angles stored in \$TC_CARR13 and \$TC_CARR14 were used for the calculation of the active tool length with TCOABS. This still applies if \$TC_CARR21 and \$TC_CARR22 do not refer to rotary axes. If \$TC_CARR21 or \$TC_CARR22 contains a reference to a rotary axis in the channel, the axis position of the relevant axis at the start of the current block is used as the angle, rather than the entry in \$TC_CARR13 or \$TC_CARR14.

A mixed operating mode is permissible, i.e. the angles can be determined from the entry in the system variables \$TC_CARR13 or \$TC_CARR14 for one axis, and from the position of a channel axis for the other.

This makes it possible for machines, on which the axes used to set the tool carrier with orientation capability are known within the NC, to access their position directly, whereas it was previously necessary, for example, to read system variable \$AA_IM[axis] and write the result of the read operation to \$TC_CARR13/14. In particular, this removes the implicit preprocessing stop when reading the axis positions.

MD20180

The rotary axis position is used with its programmed or calculated value, when the machine data:

MD20180 \$MC_TOCARR_ROT_ANGLE_INCR[i] = 0 (rotary axis increment of the tool carrier that can be oriented)

If the machine data is not zero however, the position used is the nearest grid point obtained for a suitable integer value n from the equation:

$$\varphi = \$MC_TOCARR_ROT_ANGLE_OFFSET[i] + n * \$MC_TOCARR_ROT_ANGLE_INCR[i]$$

This functionality is required if the rotary axes need to be indexed and cannot, therefore, assume freely-defined positions (e.g. with Hirth tooth systems). System variable $\$P_TCANG[i]$ delivers the approximated value and system variable $\$P_TCDIFF[i]$ the difference between the exact and the approximated value.

Frame orientation TCOFR

With $TCOFR$ (determination of the angle from the orientation defined by an active frame), the increments are scaled after determination of the angle from the active frame rotation. If the requested orientation is not possible due to the machine kinematic, the machining is aborted with an alarm. This also applies if the target orientation is very close to an achievable orientation. In particular the alarm in such situations cannot be prevented through the angle approximation.

TCARR frame offset

A frame offset as a result of a tool carrier change becomes effective immediately on selection of $TCARR=.$. . . A change in the tool length, on the other hand, only becomes effective immediately if a tool is active.

TCOFR/TCOABS frame rotation

A frame rotation does not take place on activation and a rotation which is already active is not changed. As in case T (only the tool can be rotated), the position of the rotary axes used for the calculation is dependent on the G command $TCOFR/TCOABS$ and determined from the rotation component of an active frame or from the entries $\$TC_CARRn$.

Activation of a frame changes the position in the workpiece coordinate system accordingly, without compensating motion by the machine itself. The ratios are shown in the figure below:

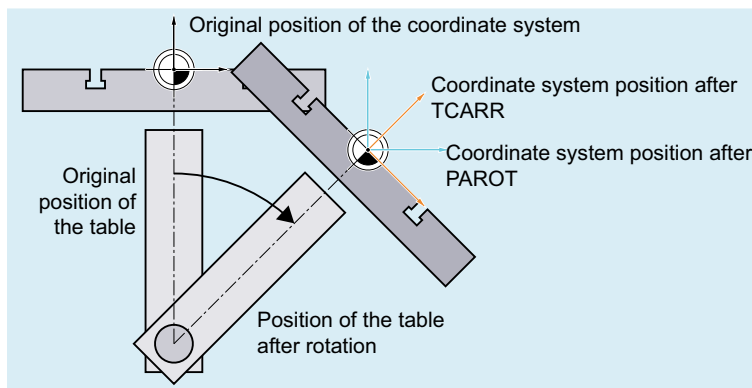


Figure 3-38 Work offset on activation of a rotary table with $TCARR$

Example

On the machine in the figure, the axis of rotation of the table is pointing in the positive Y direction. The table is rotated by +45 degrees. `PAROT` defines a frame, which similarly describes a rotation of 45 degrees about the Y axis. The coordinate system is not rotated relative to the actual environment (marked in the figure with "Position of the coordinate system after `TCARR`"), but is rotated by -45 degrees relative to the defined coordinate system (position after `PAROT`). If this coordinate system is defined with `ROT Y-45`, for example, and if the tool carrier is then selected with active `TCOFR`, an angle of +45 degrees will be determined for the axis of rotation of the tool carrier.

Rotary table

With rotary tables (kinematic types P and M), activation with `TCARR` similarly does not lead to an immediate rotation of the coordinate system (see figure), i.e. even though the zero point of the coordinate system is offset relative to the machine, while remaining fixed relative to the zero point of the workpiece, the orientation remains unchanged in space.

Activation of kinematic types P and M

With kinematics of type P and M the selection of a tool carrier activates an additive frame (table offset of the tool carrier with orientation capability), which takes into account the zero point offset as a result of the rotation of the table.

The work offset can be written to a dedicated system frame `$P_PARTFR`. For this, the bit 2 must be set in the machine data:

MD28082 `$MC_MM_SYSTEM_FRAME_MASK` (system frames (SRAM))

The basic frame identified by following machine data is then no longer required for the work offset:

MD20184 `$MC_TOCARR_BASE_FRAME_NUMBER` (number of the basic frames for taking the table offset)

Activation of kinematic type M

With kinematics of type M (tool and table are each rotary around one axis), the activation of a tool carrier with `TCARR` simultaneously produces a corresponding change in the effective tool length (if a tool is active) and the work offset.

Rotations

Depending on the machining task, it is necessary to take into account not only a work offset (whether as frame or as tool length) when using a rotary tool carrier or table, but also a rotation. However, the activation of an orientable tool carrier never leads directly to a rotation of the coordinate system.

TOROT

If only the tool can be rotated, a frame whose Z axis points in the direction of the tool can be defined with `TOFRAME` or `TOROT`.

PAROT

If the coordinate system needs to be fixed relative to the workpiece, i.e. not only offset relative to the original position but also rotated according to the rotation of the table, then `PAROT` can be used to activate such a rotation in a similar manner to the situation with a rotary tool.

With `PAROT`, the translations, scaling and mirroring in the active frame are retained, but the rotation component is rotated by the rotation component of a tool carrier with orientation capability corresponding to the table.

`PAROT` and `TOROT` take into account the overall change in orientation in cases where the table or the tool are oriented with two axes of rotation. With mixed kinematics, only the corresponding component caused by an axis of rotation is considered. It is thus possible, for example, when using `TOROT`, to rotate a workpiece such that an oblique plane lies parallel to the XY plane fixed in space, whereby rotation of the tool must be taken into account in machining where any holes to be drilled, for example, are not perpendicular to this plane.

Language command `PAROT` is not rejected if no orientable tool carrier is active. This causes no changes in the programmed frame.

Note

For further information about the `TCARR` and `TOROT` as well as `PAROT` functions with regard to channel-specific system frames, see "Basic Functions" Function Manual.

3.12.9 Procedure when using toolholders with orientation capability

Creating a tool carrier

Setting the number of available tool carrier data sets

The number of available tool carrier data sets in the control system is defined with machine data:

MD18088 \$MN_MM_NUM_TOOL_CARRIER (maximum number of definable tool carriers)

The number must be set according to the following rule:

MD18088 \$MN_MM_NUM_TOOL_CARRIER =	
	<highest number of tool carriers required in one channel> *
	<number of parameterized channels>

The number of tool carriers parameterized in the machine data is then distributed equally across the parameterized channels and TO units.

<available number of tool carriers per channel or TO unit> =	
	MD18088 \$MN_MM_NUM_TOOL_CARRIER /
	<number of parameterized channels>

Note

For further explanations on definition and assignment of a TO unit to a channel with machine data MD28085 by machine data \$MC_MM_LINK_TOA_UNIT, see:

Further information

Function Manual Basic Functions; Memory configurations

Zero setting of tool carrier data:

You can use the command `$TC_CARR1[0] = 0` to zero all values of all data sets.

Individual tool carrier data sets can be deleted selectively with the NC command `DELTC` or the PI service `_N_DELTCAR`.

Accessing the data of a tool carrier:

- Part program
 - Write: `$TC_CARR<n>[<m>] = <value>`
Value `<value>` is written to parameter `<n>` of tool carrier `<m>` .
 - Read: `<value> = $TC_CARR<n>[<m>]`
Parameter `<n>` of tool carrier `<m>` is read. If the referenced tool carrier is not defined, an alarm is displayed.
- OPI interface
The parameters of a tool carrier with orientation capability can be read and written with the system variables `$P_TCANG[<n>]`.

Data backup

The system variables specified above are saved as part of the general NC data backup.

Selecting the tool carrier

A tool carrier with number `<m>` is selected with the `TCARR = <m>` NC program command (**TCARR**Tool **C**arrier).

`TCARR = 0` deselects an active tool carrier.

New tool or new tool carrier

When a new tool is activated, it is always treated as if it was mounted on the active tool carrier.

A new tool carrier is activated immediately when it is programmed. It is not necessary to change tools or reprogram the active tool. The tool carrier (number) and tool (number) are independent and can be used in any combination.

Tool carrier from G command of group 42

Absolute tool orientation **TCOABS** (**T**ool **C**arrier **O**rientation **ABS**olute):

Tool orientation is determined explicitly if the corresponding values are entered in system variable `$TC_CARR13` or `$TC_CARR14` and G command **TCOABS** is activated in G group 42.

Frame tool orientation **TCOFR** (Tool Carrier Orientation **FR**ame):

Tool orientation can also be determined automatically from the current orientation of an active frame when selecting a tool, if one of the following G commands is active in G group 42 when the tool carrier is selected:

- **TCOFR** or **TCOFRZ**
The tool carrier with orientation capability is set so that the tool points in the Z direction.
- **TCOFRX**
The tool carrier with orientation capability is set so that the tool points in the X direction.
- **TCOFRY**
The tool carrier with orientation capability is set so that the tool points in the Y direction.

The effect of **TCOFR** is such that, when machining on an inclined surface, tool offsets are considered implicitly as if the tool were standing vertically on the surface.

Note

The tool orientation is not bound strictly to the frame orientation. When a frame is active and G command **TCOABS** is active, you can select a tool, whereby the orientation of the tool is independent of the orientation of the active frame.

Following tool selection, you can change the frame, which does not affect the components of tool length compensation. It is then no longer certain that the tool is positioned perpendicular to the machining plane. You should therefore first check that the intended tool orientation is maintained on an inclined surface.

When **TCOFR**, etc., is active, the tool orientation used in the tool length calculation is always determined from the active frame each time the tool carrier is changed.

Tool carrier from G command of group 53

The G commands of group 53 (**TOFRAME**, **TOROT**, etc.) can be used to define a frame such that an axis direction (Z, Y or X) in this frame is equal to the current tool orientation.

The G command of group 6 (G17 - G19), which is active at the time **TOFRAME** is called, determines the tool orientation.

If no tool carrier is active, or if a tool carrier is active but does not cause the tool orientation to change, the Z direction in the new frame is:

- The same as the old Z direction with G17.
- The same as the old Y direction with G18.
- The same as the old X direction with G19.

These directions are modified accordingly for rotating tool carriers. The same applies to the new X and Y directions.

Instead of **TOFRAME** or **TOROT**, one of the G commands **TOFRAMEX**, **TOFRAMEY**, **TOROTX**, or **TOROTY** can be used. The meanings of the axes are interchanged accordingly.

Group change

Changing the G command from group 42 (TCOABS, TCOFR, etc.) causes recalculation of the tool length components.

The (programmed) angles of rotation stored in the tool carrier data is not affected, with the result that the angles originally stored in the tool carrier data is reactivated on a change from TCOFR to TCOABS.

Read rotary angle (α_1 or α_2):

The angles currently used to calculate the orientation can be read via system variable \$P_TCANG[n] where $n = 1$ or $n = 2$.

If two permissible solutions (i.e. a second valid pair of angles) are available for a particular orientation, the values can be accessed with \$P_TCANG[3] or \$P_TCANG[4]. The number of valid solutions 0 to 2 can be read with \$P_TCSOL.

Tool radius compensation with CUT2D or CUT3DFS:

The current tool orientation is included in the tool radius compensation if either CUT2D or CUT3DFS is active in G group 22 (tool offset type).

For nonrotating tool carriers, the behavior depends solely on the active plane of the G command of group 6 (G17 - G19) and is, therefore, identical to the previous behavior.

All other tool offset types:

The behavior for all other tool offset types is unchanged.

For CUT2DF and CUT3DF in particular, the compensation plane used for TRC is determined from the active frame, independent of the current tool orientation. Allowance is made for the active plane (G17 - G19) and the behavior is, therefore, the same as before.

The two remaining G commands of group 22, CUT3DC and CUT3DF, are not affected by the tool carrier functionality because the tool orientation information in these cases is made available by the active kinematic transformation.

Two rotary axes

Two general solutions exist for two rotary axes. The control itself chooses these two solution pairs such that the orientation angles resulting from the frame are as close as possible to the specified angles.

The two following options are available for specifying the angles:

1. If \$TC_CARR21 or \$TC_CARR22 contains a reference to a rotary axis, the position of this axis at the start of the block in which the tool carrier is activated is used to specify the angle.
2. If \$TC_CARR21 or \$TC_CARR22 does not contain a reference to a rotary axis, the values contained in \$TC_CARR13 or \$TC_CARR14 are used.

Example

The control first calculates an angle of 10 degrees for one axis. The specified angle is 750 degrees. 720 degrees ($= 2 * 360$ degrees) are then added to the initial angle, resulting in a final angle of 730 degrees.

Rotary axis offset

Rotary axis offsets can be specified with system variables \$TC_CARR24 and \$TC_CARR25. A value not equal to zero in one of these parameters means that the initial state of the associated rotary axis is the position specified by the parameter (and not position zero). All angle specifications then refer to the coordinate system displaced by this value.

When the machining plane is changed (G17 - G19), only the tool length components of the active tool are interchanged. The components of the tool carrier are not interchanged. The resulting tool length vector is then rotated in accordance with the current tool carrier and, if necessary, modified by the offsets belonging to the tool carrier.

The two tool carrier angles α_1 and α_2 are determined using a frame, independent of the active plane currently selected (G17 - G19).

Limit values

Limit angles (software limits) can be specified for each rotary axis in the system variable set (\$TC_CARR30 to \$TC_CARR33) used to describe the tool carrier with orientation capability. These limits are not evaluated if both the minimum and maximum value is zero.

If at least one of the two limits is not equal to zero, the system checks whether the previously calculated solution is within the permissible limits. If this is not the case, an initial attempt is made to reach a valid setting by adding or subtracting multiples of 360 degrees to or from the invalid axis position. If this is impossible and two different solutions exist, the first solution is discarded and the second solution is used. The second solution is treated the same as the first with reference to the axis limits.

If the first solution is discarded and the second used instead, the contents of \$P_TCANG[1/2] and \$P_TCANG[3/4] are swapped, hence the solution actually used is also stored in \$P_TCANG[1/2] in this case.

The axis limits are monitored even if the axis angle is specified instead of being calculated. This is the case if TCOABS is active when a tool carrier with orientation capability is activated.

3.12.10 Supplementary conditions and control system response for orientation

Full orientation

For a given data set that describes a certain kinematic, all the conceivable special orientations can only be displayed when the following conditions are satisfied:

- The two vectors v_1 and v_2 that describe the axes of rotation must also be defined (i.e. both vectors must not be equal to zero).
- The two vectors v_1 and v_2 must be perpendicular to each other.
- The tool orientation must be perpendicular to the second axis of rotation.

Non-defined orientation

If these conditions are not satisfied and an orientation that cannot be achieved by an active frame is requested with TCOFR, an alarm is output.

Vector/angle of rotation dependencies

If vector v_1 or v_2 , which describes the direction of an axis of rotation, is zero, the associated angle of rotation α_1 or α_2 must also be zero. Otherwise, an alarm is output. The alarm is not output until the tool carrier is activated, i.e. when the tool carrier is changed.

Tool fine compensation combined with orientation

Tool fine compensations and toolholders **cannot** be combined. The activation of tool fine compensation when a tool carrier is active, and vice versa the activation of the tool carrier when tool fine compensation is active, produces an alarm.

Automatic tool carrier selection, RESET

For **RESET** or at program start, a tool carrier can be selected automatically via the machine data:

MD20126 \$MC_TOOL_CARRIER_RESET_VALUE (Active tool carrier on RESET)

It is handled similar to the controlled selection of a tool via the machine data:

MD20120 \$MC_TOOL_RESET_VALUE (Tool length comp. power up (RESET/TP-End))

The behavior at **RESET** or at program start is controlled as in the case of tool selection via the same bit 6 in the machine data:

MD20110 \$MC_RESET_MODE_MASK (definition of initial control settings after RESET/TP-End)

Or:

MD20112 \$MC_START_MODE_MASK (definition of initial control system settings at NC-START)

For further information, see Function Manual "Basic Functions", Section "Mode group, channel, program operation, reset behavior".

As of SW 6.3

If **TCOABS** was active for the last selection before reset, the behavior is unchanged compared to previous versions. A different active G code causes the tool carrier with orientation capability to be activated with the frame that was active before the last reset. Modified tool carrier data (\$TC_CARR...) are also considered. If these data are unchanged, the tool carrier is activated in exactly the same state as before reset. If the tool carrier data were changed after the tool carrier selection before reset, selection corresponding to the last frame is not always possible. In this case, the tool carrier with orientation capability is selected according to the G code values of group 42 valid at this time and the active frame.

MD22530 output of auxiliary functions to PLC

That, optionally, a constant or an M code is output when the tool carrier is selected, whose number of the code is derived from the tool carrier number. Can be set with the machine data:

MD22530 \$MC_TOCARR_CHANGE_M_CODE (M code at change of tool carrier)

For further information, see Function Manual "Basic Functions", Section "Mode group, channel, program operation, reset behavior".

MD20360 \$MC_TOOL_PARAMETER_DEF_MASK

Definition of the effect of the tool parameters:

Bit10: The tool portion of an active orientable tool carrier becomes effective even if no tool is active.

Bit19: The change in orientation of a tool caused by an orientable tool carrier is effective even if no tool is active. This bit is only effective if bit 10 is also set.

Bit23: The orientable tool carrier is suppressed when the tool lengths are suppressed.

Tool carrier kinematics

The following constraints must be adhered to for tool carrier kinematics:

- Tool orientation in initial state, both angles α_1 and α_2 zero, as per default setting, even if:
 - G17 parallel to Z
 - G18 Parallel to Y
 - G19 parallel to Z
- A permissible position in terms of the axis limits must be achievable.
- For any possible orientation to be set, the two rotary axes must be perpendicular to each other.

For machines, on which the table is rotated by both axes, the tool orientation must also be perpendicular to the first axis of rotation.

For machines with mixed kinematics, the tool orientation must be perpendicular to the axis which rotates the tool, i.e. also the first axis of rotation.

The following applies to orientations specified **in a frame**:

- The orientation specified in a frame must be achievable with the defined tool carrier kinematics, otherwise an alarm is output.

This situation can occur if the two axes of rotation required to define the kinematics are not perpendicular to each other.

This applies if fewer than two axes of rotation are defined and is the case:

 - With **kinematic type T with rotary tool**, if the tool axis, which defines the tool direction, is not perpendicular to the **second** axis.
 - With **kinematic types M and P with rotary workpiece**, if the tool axis, which defines the tool direction, is not perpendicular to the **first** axis.
- Axes of rotation which require a frame with a defined tool orientation in order to reach a specific position are only determined unambiguously in the case of one axis of rotation. Two general solutions exist for two axes of rotation.
- In all cases where ambiguities have to be considered, it is particularly important that the approximate angles expected from the frame are stored in the tool data and that the axes of rotation are in the vicinity of the expected positions.

Response with ASUB, REPOS

The tool carrier can be changed in an asynchronous subprogram (ASUB). When the interrupted program is resumed with REPOS, the approach motion of the new tool carrier is taken into account and the program continues with this motion. The treatment here is analogous to tool change in an ASUB.

For further information, see Function Manual "Basic Functions", Section "Mode group, channel, program operation, reset behavior".

3.13 Online tool length compensation

3.13.1 Function

In the automatic mode, using function "Online tool length compensation", users can override the effective tool lengths of a tool aligned via an orientation transformation/kinematic chain or an orientable tool carrier in all three tool directions in realtime. The online tool length compensations do not change the tool parameters, but are taken into account within the orientation transformation/kinematic chain or the tool carrier that can be oriented, so that offsets are obtained in the tool coordinate system (TCS).

Activating the online tool length compensation

The online tool length compensation can be activated in two ways:

- While executing the NC program by activating the function with TOFFON(...) and assigning compensation values using system variable \$AA_TOFF[<n>]
→ More information: Chapter "Programming (Page 175)"
- Using handwheel override in tool direction
The online tool length compensation using handwheel override in the tool direction is a **DRF offset in the tool coordinate system (TCS)**.
→ Detailed information: Function Manual Axes and Spindles
Using this function, the effective tool length in the longitudinal direction (L1 direction, for G17 Z direction) of a tool that is oriented can be overridden in the following states:
 - When the NC program is stopped (by NC stop, M0 or active single-block processing)
 - At cycle stop
 - With feedrate override = 0
 - During machining

Depending on the parameterization, online tool length compensation using handwheel override in tool direction is also possible without prior activation by programming TOFFON.

Preconditions

The following requirements best be met to be able to apply function "Online tool length compensation":

- An orientation transformation (TRAORI) / kinematic chain or orientable tool carrier (TCARR) is active.
- The "AUTOMATIC" or "MDI" operating mode is active.
- The following applies for handwheel override in tool the direction:
 - The handwheel, from which the increments for the online tool length compensation can be derived, is assigned to the machine axis on which the third geometry axis is emulated.
 - The tool coordinate system (TCS) and function "DRF offset" are active.
→ More information: Function Manual Axes and Spindles

Effectiveness

The following applies for online tool length compensation:

- It is possible in the "AUTOMATIC" and "MDI" modes.
- When changing the tool or changing the tool orientation, the active tool offset (SAA_TOFF) is retained.

Note

Before a subsequent machining operation or a tool change, the active tool offset (SAA_TOFF) must be cleared using CORROF or DRFOF, if necessary.

- With end of program/channel reset, the tool offset (SAA_TOFF) is deselected by default. If the tool offset (SAA_TOFF) is to remain effective beyond end of program/channel reset, the corresponding function parameter must be adjusted.
- The tool offset (SAA_TOFF) is deleted by a warm restart.

Display

The current tool offset (SAA_TOFF) of the axis can be displayed in the user interface.

The actual axis position display (ACTUAL POSITION) does not change due to the online tool length compensation.

Parameterization

The function parameters are set via machine and setting data.

→ More information: Chapter "Parameterization (Page 172)".

NC/PLC interface

As long as the online tool length compensation is active, the following NC/PLC interface signal has the value "1":

```
<Chan>.basic.in.toffonActive = 1 (TOFF active)
```

3.13 Online tool length compensation

While a correction motion is active, the following NC/PLC interface signal has the value "1":

`<Chan>.basic.in.toffonMotionActive = 1` (TOFF motion active)

Constraints

Constraints must be observed when using the online tool length compensation.

→ More information: Chapter "Constraints (Page 181)"

PLC signals

Basic Program Plus	Basic Program	
<code><Chan>.basic.in.toffonActive</code>	<code>LBP_Chan*.E_TOFF</code>	DB21, ...DBX318.2
<code><Chan>.basic.in.toffonMotionActive</code>	<code>LBP_Chan*.E_TOFFmovem</code>	DB21, ...DBX318.3

3.13.2 Parameterization

Online tool length compensation

Operating principle of the online tool length compensation

The operating principle of the online tool length compensation is set in the machine data:

MD21190 \$MC_TOFF_MODE

Bit	Meaning
0	Tool offset \$AA_TOFF[<n>]: Behavior at end of program/channel reset
	= 0 The tool offset \$AA_TOFF[<n>] is deselected.
	= 1 The tool offset \$AA_TOFF[<n>] is retained.
1 ... 3	Tool offset \$AA_TOFF[<n>]: Effect of the value assignment Bits 1 ... 3 define whether the specified offset value for the respective tool direction is to be calculated absolutely or incrementally (integrating).
	= 0 Absolute
	= 1 Incremental Note: The incremental offset of \$AA_TOFF[<n>] enables 3D clearance control.
4	Offset motion: behavior at NC stop
	= 0 Offset motion is interrupted
	= 1 Offset motion remains active
5	Offset motion: Behavior at end of program/channel reset
	= 0 Offset motion is interrupted
	= 1 Offset motion remains active

Bit	Meaning
6	Tool offset \$AA_TOFF[<n>]: Effectiveness of SD42972 \$SC_TOFF_LIMIT_MINUS[<n>] (lower limit of online tool length compensation)
	= 0 Not active
	= 1 Active
7	Tool offset \$AA_TOFF[<n>]: Block-by-block suppression with SUPA
	= 0 No
	= 1 Yes
8	Handwheel override in tool direction: Behavior when limit value is reached SD42970 \$SC_TOFF_LIMIT[<n>] or SD42972 \$SC_TOFF_LIMIT_MINUS[<n>]
	= 0 Interruption of traversing movement ⇒ Collection of the displacements entered via the handwheel
	= 1 Canceling of traversing movement ⇒ No collection of the displacements entered via the handwheel
9	Handwheel override in tool direction: Override also possible without activation of online tool length compensation by TOFFON
	= 0 Only possible after activation by TOFFON(...)
	= 1 Possible without activation by TOFFON(...)
10	Tool offset \$AA_TOFF[<n>]: Axis assignment when changing plane
	= 0 The axis components are exchanged cyclically on plane change from G17 to G18 or G19. This means that all three axes of the rotated coordinate system point in the positive axis directions of the initial coordinate system.
	= 1 The coordinate system is rotated -90° around the X axis on plane change from G17 to G18, and 90° around the Y axis on plane change from G17 to G18. In both cases, the rotated Z axis points in the same direction as when bit 10 is not set. The other two axes are exchanged in comparison to the setting with bit 10 not set, and one of the two axes points in the negative direction. This setting should only be used if compatibility with older software versions is required (SW 4.5 and older).
11	Offset motion: Taking overrides into account
	= 0 Overrides (e.g. DRF offsets) in the orientation axes are not taken into account for the direction of the offset motion.
	= 1 Overrides (e.g. DRF offsets) in the orientation axes are taken into account for the direction of the offset motion.

Velocity and acceleration of the offset motion

The velocity and acceleration with which the superimposed movements of the online tool length compensation are traversed out can be set via the following machine data:

- MD21194 \$MC_TOFF_VELO (feedrate for online correction in tool direction)
- MD21196 \$MC_TOFF_ACCEL (acceleration for online correction in tool direction)

Acceleration margin for overlaid movements

In the default setting, path axes can use up to 80% of the maximum path acceleration and velocity. The remaining 20% are reserved for overlaid movements of the online tool length

3.13 Online tool length compensation

compensation. This means that 100% of the maximum path acceleration and velocity can only be utilized when an overlaid movement is requested.

The share of the acceleration reserve in the channel can be changed if required via the following machine data:

MD20610 \$MC_ADD_MOVE_ACCEL_RESERVE

Upper limit of the online tool length compensation

The following setting data can be used to specify an upper limit for the tool offset \$AA_TOFF[<n>]:

SD42970 \$SC_TOFF_LIMIT

This limit value influences the absolutely effective amount of offset through \$AA_TOFF[<n>].

Lower limit of the online tool length compensation

The following setting data can be used to specify a lower limit for the tool offset \$AA_TOFF[<n>]:

SD42972 \$SC_TOFF_LIMIT_MINUS

This limit value influences the effective amount of offset through \$AA_TOFF[<n>] in the negative axis direction. It is only effective if \$MC_TOFF_MODE bit 6 = 1 is set in the machine data MD21190.

Handwheel

To be able to use the "Handwheel override in tool direction" function, the handwheels must have been commissioned.

→ More information: "Axes and Spindles" Function Manual

Tool offset

The offset value resulting from turning the handwheel by one detent position depends on the following parameters:

- Pulse evaluation of the handwheel:
MD11320 \$MN_HANDWH_IMP_PER_LATCH (handwheel pulses per detent position)
- Active increment size:
 - Fixed increment size INC1, INC10, INC100, INC1000 or INC10000:
MD11330 \$MN_JOG_INCR_SIZE_TAB[1 ... 5] = <Number of increments 1 ... 5> (Fixed increment size INC1, INC10, INC100, INC1000 or INC10000)
 - Variable increment size INCvar:
SD41010 \$SN_JOG_VAR_INCR_SIZE = <number of increments>
 - Limitation of the size of the selected increment for geometry axes:
MD20620 \$MC_HANDWH_GEOAX_MAX_INCR_SIZE
 - Limitation of the size of the selected increment for orientation axes:
MD20621 \$MC_HANDWH_ORIAX_MAX_INCR_SIZE
- Distance of an increment:
MD31090 \$MA_JOG_INCR_WEIGHT (evaluation of an increment for INC/handwheel)

Velocity and acceleration

The acceleration and velocity settings of the online tool length compensation are effective for the offset movements resulting from the handwheel override in tool direction:

- MD21194 \$MC_TOFF_VELO (feedrate for online correction in tool direction)
- MD21196 \$MC_TOFF_ACCEL (acceleration for online correction in tool direction)

Velocity override

Overrides of path or orientation velocity can be set via the machine data:

MD20622 \$MC_HANDWH_GEOAX_MAX_INCR_VSIZE (path velocity override)

MD20623 \$MC_HANDWH_ORIAX_MAX_INCR_VSIZE (orientation velocity overlay)

Response to special channel-specific NC/PLC interface signals

How the handwheel override in tool direction should react to special channel-specific NC/PLC interface signals is set in the machine data:

MD20624 \$MC_HANDWH_CHAN_STOP_COND

Response to special axis-specific NC/PLC interface signals

How the machine axis moved by the handwheel should react to special axis-specific NC/PLC interface signals is set in the machine data:

MD32084 \$MA_HANDWH_STOP_COND

Display

The override movements caused by the online tool length compensation (tool offset SAA_TOFF) can be displayed in the user interface.

The following machine data settings are required for this:

- For display in the "Machine" operating area in the "Override" window:
MD52210 \$MCS_FUNCTION_MASK_DISP, Bit 15 = 1
In addition to the tool offset SAA_TOFF, limits defined via the setting data SD42970 \$SC_TOFF_LIMIT and SD42972 \$SC_TOFF_LIMIT_MINUS are also displayed.
- For display in the "Parameters" operating area in the "Work offset - Overview" window:
MD52211 \$MCS_FUNCTION_MASK_DISP_ZOA, Bit 24 = 1

3.13.3 Programming**3.13.3.1 Online tool length compensation (TOFFON, TOFFOF)**

With the "Online tool length compensation" function, it is possible to override the effective tool lengths in all three tool directions in real time during machining **with active orientation transformation (TRAORI)/kinematic chain** or **active orientable tool carrier (TCARR)**.

Predefined procedures TOFFON() and TOFFOF() are used for activation and deactivation.

3.13 Online tool length compensation

Already during activation an offset value can be specified for the specified direction of compensation and this is immediately traversed.

Online tool length compensations can be applied via the \$AA_TOFF[<n>] system variable. This is possible in all three tool directions of an active orientation transformation/or an active orientable tool carrier. The three geometry axis names X, Y, Z are used as index <n>. The number of active compensation directions must not exceed the number of active geometry axes. All three offset directions can be active at the same time.

Syntax

```
...
TOFFON(<CorrDir>[,<Offset>])
WHEN TRUE DO $AA_TOFF[<CorrDir>]=<CorrVal>
...
TOFFOF(<CorrDir>)
...
```

Meaning

TOFFON	Activate online tool length compensation	
	<CorrDir>	Correction direction Tool direction (X, Y, Z), in which the on-line tool length compensation should be active.
	<Offset>	Offset value (optional) Offset value to be traversed immediately in the specified offset direction when activated.
WHEN TRUE DO ...	Synchronous action that starts as soon as the block is changed in.	
\$AA_TOFF[<CorrDir>]=<CorrVal>	<CorrDir>	Correction direction Tool direction (X, Y, Z) in which the specified offset value is to be interpolated.
	<CorrVal>	Correction value
TOFFOF	Deactivate online tool length compensation	
	<CorrDir>	Correction direction Tool direction (X, Y, Z) in which the on-line tool length compensation should be deactivated.
After deselecting online tool length compensation in the specified compensation direction, the value of the corresponding system variable \$AA_TOFF[<n>] or \$AA_TOFF_VAL[<n>] is reset to "0". A preprocessing stop is triggered.		

Note

Repeated programming of the instruction TOFFON() with a new offset value causes the new offset value to be traversed. The offset value is added to the variable \$AA_TOFF[<n>] as an absolute value.

Note

Before switching on or switching off a transformation, an overlaid motion must be deactivated out using TOFFOF().

Examples**Example 1 and 2: Online tool length compensation with active orientation transformation (TRAORI)**

Program code	Comment
\$MC_TOFF_MODE=1	; The offset value \$AA_TOFF[<n>] is to be absolutely calculated and retained beyond reset.
\$MC_TOFF_VELO[0]=1000	; Velocity for online offset motion in tool direction X, Y, Z
\$MC_TOFF_VELO[1]=1000	
\$MC_TOFF_VELO[2]=1000	
\$MC_TOFF_ACCEL[0]=1	; Acceleration for the online offset motion in tool direction X, Y, Z
\$MC_TOFF_ACCEL[1]=1	
\$MC_TOFF_ACCEL[2]=1	
N5 DEF REAL XOFFSET	
N10 TRAORI(1)	; Orientation transformation ON
N20 TOFFON(Z)	; Activate online tool length compensation in tool direction Z.
N30 WHEN TRUE DO \$AA_TOFF[Z]=10 G4 F5	; Tool length compensation for tool direction Z: 10 mm
...	
N100 XOFFSET=\$AA_TOFF_VAL[X]	; Assign the position offset in tool direction X to the variable XOFFSET.
N120 TOFFON(X,-XOFFSET) G4 F5	; Deactivate online tool length compensation for tool direction X. ; Return the position offset in tool direction X to 0.

Program code	Comment
N10 TRAORI(1)	; Orientation transformation ON
N20 TOFFON(X)	; Deactivate online tool length compensation for tool direction X.
N30 WHEN TRUE DO \$AA_TOFF[X]=10 G4 F5	; Tool length compensation for tool direction X: 10 mm

3.13 Online tool length compensation

Program code	Comment
...	
N80 TOFFOF(X)	; Deactivate online tool length compensation for tool direction X. ; Position offset of tool direction X is deleted: \$AA_TOFF[X]=0 ; No axis is traversed. ; The position offset is added to the actual position in the WCS corresponding to the actual orientation.

Example 3: Online tool length compensation with active orientable tool carrier (TCARR)

Program code	Comment
...	
DEF REAL XOFFSET	
\$MC_TOFF_MODE=1	; The offset value \$AA_TOFF[<n>] is to be absolutely calculated and retained beyond reset.
\$MC_TOFF_VELO[0]=10000	; Velocity for online offset motion in tool direction X, Y, Z
\$MC_TOFF_VELO[1]=10000	
\$MC_TOFF_VELO[2]=10000	
\$MC_TOFF_ACCEL[0]=1	; Acceleration for the online offset motion in tool direction X, Y, Z
\$MC_TOFF_ACCEL[1]=1	
\$MC_TOFF_ACCEL[2]=1	
NEWCONF	
; CYCLE800()	
TCARR=0	; Deactivate tool carrier.
\$P_WPFR=CTRANS()	
\$P_WPFRAME=\$P_WPFR	
G0 X0 Y0 Z0 A0 C0	
M0	
; CYCLE800(0,"HEAD",100000,57,0,0,0,30,0,0,0,0,0,-1,100,101)	
G17	
\$P_WPFR=CTRANS()	
\$P_WPFR=CROT(X,30)	
\$P_WPFRAME=\$P_WPFR	
CUT2DF TCARR=2	; Activate tool carrier.
TCOFR	
PAROT	
G0 C=\$P_TCANG[3] A=\$P_TCANG[4]	
M0	
TOFFON(Z)	; Activate online tool length compensation for tool direction Z.

3.13 Online tool length compensation

Program code	Comment
WHEN TRUE DO \$AA_TOFF[Z]=10	; Tool length compensation for tool direction Z: 10 mm
M0	
G4 F5	
M0	
WHEN TRUE DO \$AA_TOFF[Z]=0	; Tool length compensation for tool direction Z: 0 mm
M0	
G4 F5	
M0	
TOFFOF(Z)	; Deactivate online tool length compensation for tool direction Z.
M0	
; CYCLE800()	
TCARR=0	; Deactivate tool carrier.
\$P_WPFR=CTRANS()	
\$P_WPFRAME=\$P_WPFR	
M30	

More information

Block preparation

In the case of block preparation in forward motion, the tool length offset currently active in the main run is considered. In order to utilize the maximum permissible axis velocities as far as possible, it is necessary to halt the block preparation with a preprocessing stop (STOPRE) while a tool length offset is being generated.

The tool length offset is always known at the prerun time even if the tool length offsets are not changed after the program start. Or if, after a change in tool length offsets, more blocks have been processed than the IPO buffer can accommodate between the prerun and the main run. In this way, correct axis velocities are applied quickly for short blocks.

3.13 Online tool length compensation

System variables

Information on online tool length compensation can be queried via the following system variables:

System variable	Meaning for online tool length compensation	
\$AA_TOFF_VAL[<n>]	Integrated value of the overlaid compensation movement in the corresponding tool direction	
\$AA_TOFF_LIMIT[<n>]	Status of the limitation by: - SD42970 \$SC_TOFF_LIMIT (upper limit of online tool length compensation) - SD42972 \$SC_TOFF_LIMIT_MINUS (lower limit of online tool length compensation)	
	= 0	Limit value not reached
	= 1	Limit value reached in positive axis direction
	= -1	Limit value reached in negative axis direction
\$AA_TOFF_PREP_DIFF[<n>]	The difference between the currently active compensation in the interpolator and the compensation that was active at the time of block preparation.	

Tool change / orientation changes

When changing the tool or changing the tool direction, the active online tool length compensations \$AA_TOFF[<n>] are retained.

Note

Changing the effective tool length using online tool length compensation produces changes in the compensatory movements of the axes in the event of changes in orientation. The resulting velocities can be higher or lower depending on machine kinematics and the current axis position.

TOFFOF

After deselecting online tool length compensation in the specified compensation direction, the accumulated tool length compensations are removed and transferred to the base coordinate system. The prerun is synchronized with the current position in main run. Since no axes are traversed here, the values of the system variable \$AA_IM[<Ax>] (current MCS setpoint of an axis) do not change. Only the values of the system variables \$AA_IW[<Ax>] (actual WCS setpoint of an axis) or \$AA_IB[<Ax>] (actual BCS setpoint of an axis) are changed. These variables now contain the deselected share of tool length compensation.

3.13.4 Constraints

Mode change

The online tool length compensation \$AA_TOFF[<n>] also remains active for an operating mode change and can be executed in all operating modes.

If a tool length compensation is interpolated on account of \$AA_TOFF[<n>] during mode change, the mode change cannot take place until the interpolation of the tool length compensation has been completed. Alarm 16907 "Channel %1 action %2 <ALNX> possible only in stop state" is issued.

Alarm 21670

In the following cases, the existing tool length compensation must first be deleted using TOFFOF() in order to suppress the output of Alarm 21670 "Channel %1 block %2 inadmissible change to the tool direction because \$AA_TOFF active":

- Deactivating the transformation using TRAFOOF
- Switchover from CP to PTP travel
- Geometry axis replacement (if a tool length offset exists in the direction of the geometry axis)
- Plane change
- Changing from axis-specific manual travel in the JOG mode to PTP travel
A change to PTP travel is not executed as long as a tool length compensation is active.

Referencing

Tool length compensation is not considered during reference point approach REF in the JOG mode.

Block search

The instructions TOFFON() and TOFFOF() are not collected and output in an action block during block search.

Deviations from the exact offset surface

If the tool is not perpendicular to the workpiece surface during processing or the contour contains curvatures whose radius is smaller than the compensation dimension, deviations compared to the actual offset surface are produced. It is not possible to produce exact offset surfaces with one tool length compensation alone.

3.14 Modification of the offset data for rotatable tools

3.14.1 Introduction

Function

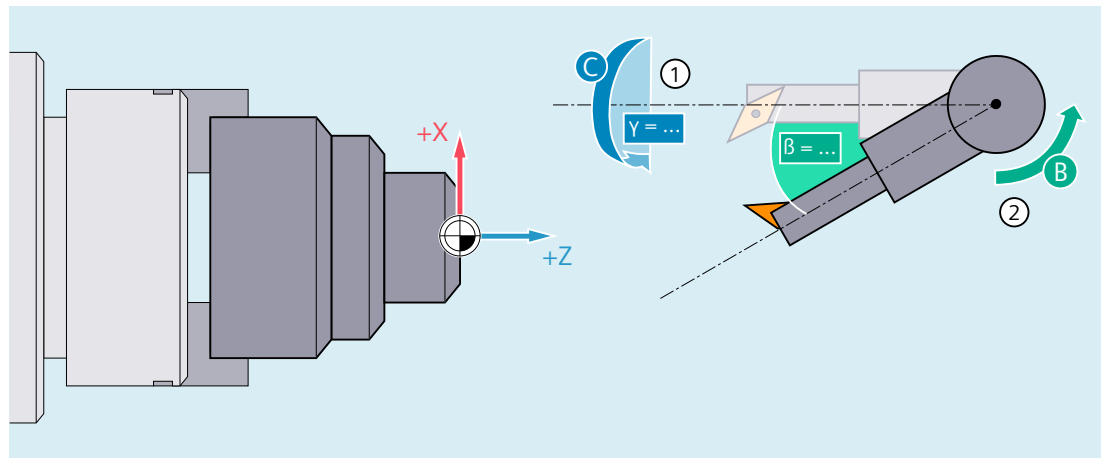
With function "Modification of the offset data for rotatable tools", the modified geometrical relationships that result when a tool is rotated (predominantly turning tools, but also drilling and milling tools) relative to the workpiece being machined can be taken into account.

The actual tool rotation is always determined from a currently active tool carrier with orientation capability (see Chapter "Tool carriers with orientation capability (Page 119)") or from an orientation transformation defined with kinematic chains (see Transformations Function Manual; Chapter "Transformation definition with kinematic chains").

The angle of rotation of the tool carrier with orientation capability is normally (but not necessarily) defined with the TCOFR command from an active frame. This method can be used to define the tool orientation independently of the actual kinematics with which the tool is rotated, identically in each case with the help of two angles.

In an active kinematic transformation, the required angles can be determined with the function ORISOLH (see "Calculating orientations (ORISOLH) (Page 198)").

The two machine-independent orientation angles β (Beta) and γ (Gamma) are used to define the tool rotation. β is the angle of rotation and the applicator (typically a B axis in G18) and γ a rotation around the abscissa (Typically a C axis in G18). The rotation is first executed around Y, finally around β . I.e., the γ axis is rotated by the β axis:



- ① Rotation around the abscissa
- ② Rotation around the applicator

Figure 3-39 Tool rotation

Commissioning

The function is commissioned using the machine and setting data.

See Chapter "Commissioning (Page 195)".

Activation

Modification of the offset data for rotatable tools is activated in the NC program with the language commands CUTMOD (in combination with tool carriers with orientation capability) or CUTMODK (for orientation transformations that were defined with kinematic chains).

See Chapter "Activating the modification of the offset data for rotatable tools (CUTMOD, CUTMODK) (Page 206)".

Results

When the function and tool rotation are active, the modified data are made available in system variables and OPI variables.

See Chapter "Activating the modification of the offset data for rotatable tools (CUTMOD, CUTMODK) (Page 206)".

3.14.2 Rotating turning tools

3.14.2.1 Cutting edge position, cut direction, and angle for turning tools

Turning tools

Turning tools means the following tools whose tool type (\$TC_DP1) has values in the range of 500 to 599. Grinding tools (tool types 400 to 499) are equivalent to turning tools.

Tools are treated independently of tool type such as turning tools if:

SD42950 \$SC_TOOL_LENGTH_TYPE = 2

Cutting edge position and cut direction

Turning tools are limited by their main and secondary cutting edges. The cutting edge position is defined via the position of the primary and secondary cutting edges relative to the coordinate axes. Two different cutting directions can be assigned to each cutting edge position.

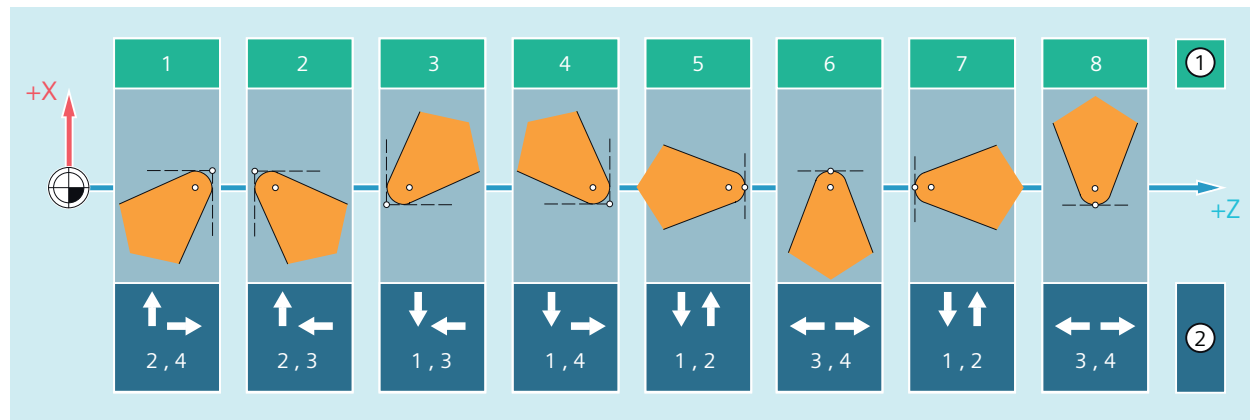


Figure 3-40 Cutting edge position and cut direction

Cutting edge position

For cutting edge position 1 - 4, the primary and secondary cutting edge are in the same quadrant.

For a cutting edge position of 5 - 8, the primary and secondary cutting edge are in adjacent quadrants or there is a coordinate axis between the two cutting edges.

The cutting edge position is stored in the tool parameter \$TC_DP2.

Cut direction

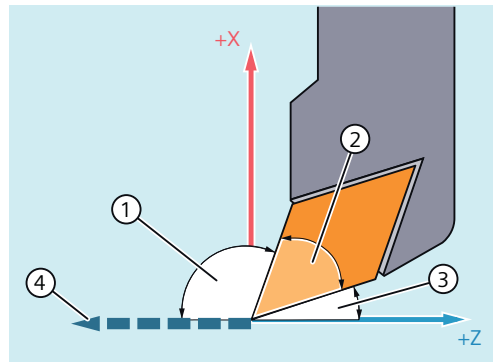
The cut directions 1 - 4 denote a positive or negative direction of the coordinate axes:

- 1: Ordinate -
- 2: Ordinate +
- 3: Abscissa -
- 4: Abscissa +

The cut direction is stored in the tool parameter \$TC_DP11.

Holder angle and clearance angle

The following figure shows the holder angle and clearance angle for a rotary tool with cutting edge position 3. The machining plane is G18 (Z/X). The cut direction is 3 (negative Z or abscissa direction).



- ① Holder angle
- ② Wedge or plate angle = $180^\circ - \text{holder angle} - \text{clearance angle}$
- ③ Clearance angle
- ④ Cut direction

Figure 3-41 Angle and cut direction for a turning tool

Cut direction

The cut direction specifies the reference direction of the holder angle. The clearance angle is the angle measured between the inverse cut direction and the adjacent cutting edge (positive).

Holder angle and clearance angle are stored in the tool parameters \$TC_DP10 or \$TC_DP24.

Note

Cut direction and tool angle are relevant only in the cutting edge positions 1 - 8.

3.14.2.2 Modifications during the rotation of turning tools

Description of tool orientation

Unlike milling tools, turning tools are not rotation-symmetric. This means that normally 3 degrees of freedom or three rotary axes are required to describe the tool orientation. The concrete kinematics therefore, is independent of the machine only to the extent the desired orientation can be set. If necessary, the third degree of freedom can be substituted by a rotation of the workpiece coordinate system.

Note

The division of the orientation into one component created by the tool carrier with orientation capability and a second component achieved via a rotation of the coordinate system is the responsibility of the application. The control does not provide any further functionality in this regard.

If the tool is oriented using a kinematic transformation, the user can use function ORISOLH to calculate the orientation. (see Chapter "Calculating orientations (ORISOLH) (Page 198)").

Angle between tool insert and machining plane

If a turning tool turns by an angle against the machining plane (i.e. around an axis in the machining plane, typically a C axis) that is not a multiple of 180°, then the configuration of the (circular) tool cutting edge in the machining plane becomes an ellipse. It is assumed that the deviations from the circular form arising on account of such rotations is so insignificant that they can be ignored (tilt angle < 5°). i.e. the control always ignores the tool orientation and assumes a circular cutting edge.

This also means that with reference to the active plane, the control accepts only a rotation by 180° as a setting deviating from the initial position. This limitation is valid for the shape of cutting edge only. The tool lengths are always considered correctly in random spatial rotations.

A rotation by 180° around an axis in the machining plane means that while using the tool at the same position, the spindle rotation direction with reference to the use of the unturned tool must be inverted.

A maximum value allowed for the angle between the tool insert and the machining plane can be specified in setting data SD42998 \$SC_CUTMOD_PLANE_TOL (see also Chapter "Parameter assignment (Page 195)").

Cutting edge position, cutting direction, and tool angle

Cutting edge position and cut direction are also not modified like the cutting edge reference point (see below) if the tool is rotated from the plane by $\pm 90^\circ$ (with a tolerance of app. 1°) because then the configuration of the cutting edge is not defined in the current plane.

If a transformation is active, the active complete frame is always included in the calculation of the actual tool rotation relative to the workpiece. If a tool carrier with orientation capability is active, the frame is only included in the calculation if bit 21 is set in machine data MD20360 \$MC_TOOL_PARAMETER_DEF_MASK (Page 195). In this case, any table

component of the tool carrier is no longer treated separately because it must be assumed that this frame component is already included in the complete frame (system frame PAROT).

When orientation transformation is active, bit 17 of machine data MD20360 \$MC_TOOL_PARAMETER_DEF_MASK can be used, should the cutting edge no longer be in the active machining plane, either to project the cutting edge into the machine plane (bit 17 = 0) or alternatively to rotate the cutting edge into the machining plane (bit 17 = 1) in order to calculate the modified cutting data.

Note

MD20360, bit 17

If this bit is set, it is the responsibility of the user to ensure, using an additional frame or by changing the orientation, to ensure that the geometric conditions during machining match those that exist during calculation.

If bit 20 of machine data MD20360 \$MC_TOOL_PARAMETER_DEF_MASK is set to value "0", standard values that deviate from zero can be used to calculate the modified cutting edge position and the modified cutting direction for when the cutting edge parameters for the holder angle and cutting direction (\$TC_DP10 or \$TC_DP24) have value 0. The following rules apply:

- A holder angle of 0° has the same effect as an angle of 112.5° on cutting edge positions 1 to 4.
- A holder angle of 0° has the same effect as an angle of 67.5° on cutting edge positions 5 to 8.
- A clearance angle of 0° has the same effect as an angle of 22.5° on cutting edge positions 1 to 4.
- A clearance angle of 0° has the same effect as an angle of 67.5° on cutting edge positions 5 to 8.

The holder angle and clearance angle might only be temporarily mapped onto the standard values specified above to be able to calculate the modified cutting edge position and modified cutting direction. I.e., reading system variables \$TC_DP10 or \$TC_DP24 still returns value "0".

If the tool rotates in the plane (rotation around an axis vertical to the machining place or around the Y axis for G18), the cutting edge position is determined from the resulting angle for the clearance and holder angles. If these two angles are not specified for the tool (i.e. \$TC_DP10 and \$TC_DP24 are both zero), the new cutting edge position is determined just from the turning angle. A special feature here is that the cutting edge position only changes in steps of 90°. I.e. the cutting edge position remains independent of the initial state either in the value range 1 to 4 or 5 to 8. The new cutting edge position is then determined exclusively from the angle of rotation if the specified values for holder angle and clearance angle are impermissible (negative values, resulting tool insert angle negative or greater than 90°). Impermissible angles cause an alarm (see MD20125 \$MC_CUTMOD_ERR (Page 195)). Clearance angle and holder angle are not modified in all these cases.

Depending on the rotation, the cut direction is modified in such a way that the resulting clearance angle remains less than 90°. If the original cut direction and the original cutting edge position do not match, then the cut direction is not modified during rotation of the tool. This situation can be set to output an alarm (see MD20125 \$MC_CUTMOD_ERR).

3.14 Modification of the offset data for rotatable tools

The angle of rotation in the plane, as it was determined from the tool carrier with orientation capability or from the active kinematic transformation, is provided in the OPI variables pTCutMod or ptCut- ModS and in system variables \$P_CUTMOD_ANG or \$AC_CUTMOD_ANG. This angle is the original angle without any final rounding to multiples of 45° or 90°.

Limit cases

If, for a turning tool, the cutting edge position, cut direction, clearance and holder angles have valid values so that all cutting edge positions (1 to 8) are possible through suitable rotations in the plane, then the cutting edge positions 1 to 4 are preferred to cutting edge positions 5 to 8 in the cases in which one of the cutting edges (main or secondary cutting edge) is away from the coordinate axis by less than half the input increment (0.0005° for an input specification of 3 decimal digits).

The following is applicable in all other cases (milling tools or turning tools without valid cutting edge parameters) in which rotation is possible only in 90° steps: If the amount of the rotation angle is smaller than 45° + 0.5 input increments (corresponds to 45.0005° for an input specification of 3 decimal places), the cutting edge position and cut direction are not changed. I.e. these cases are treated as rotations that are smaller than 45°. Rotations, the amount of which deviates from 180° by less than 45° + 0.5 input increments are treated identically as rotations in the range of 135° to 225°.

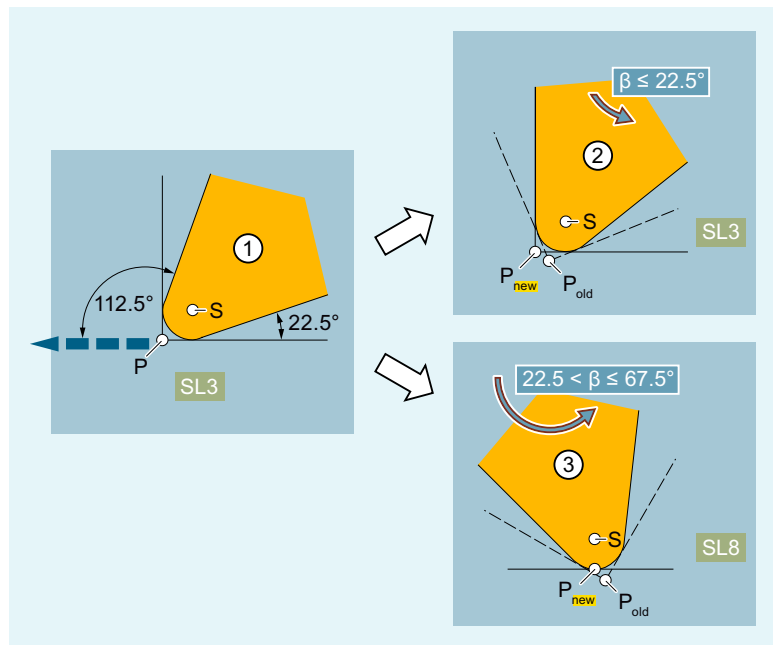
Cutting edge reference point

The cutting edge center point and the cutting edge reference point are defined for turning tools. The position of these two points relative to each other is defined by the cutting edge position.

The distance of the two points for cutting edge positions 1 to 4 is equal to $\sqrt{2}$ times the cutting edge radius; for cutting edge positions 5 to 8 it is equal to 1 times the cutting edge radius. In the first case, the cutting edge reference point relative to the cutting edge center point lies in the machining plane on a bisecting line, while in the second case it lies on a coordinate axis.

If you rotate the tool by a random angle around an axis vertical to the machining plane, the cutting edge reference point would also rotate if it had a fixed position relative to the tool. The above-mentioned condition (position on an axis or a bisecting axis) is not fulfilled in most cases. This is not desirable. Instead, the cutting edge reference point should always be modified in such a way that the distance vector between cutting edge reference point and cutting edge center point has one of the mentioned 8 directions. The cutting edge position must be modified for this if necessary.

The ratios are shown with examples in the figure below:



S: Cutting edge center point

P: Cutting edge reference point

SL: Cutting edge position

① Tool with cutting edge position 3, clearance angle 22.5° , and holder angle 112.5°

② For rotations of the tool up to 22.5° , the cutting edge position is maintained, the position of the cutting edge reference point relative to the tool however, is compensated in such a way that the relative position of both points is maintained in the machining plane.

③ For bigger tool rotations (up to 67.5°), the cutting edge position changes to value 8.

Figure 3-42 Cutting edge reference point and cutting edge position for tool rotation

Note

As the cutting edge reference point is defined by the tool length vector, modifying the cutting edge reference point changes the effective tool length.

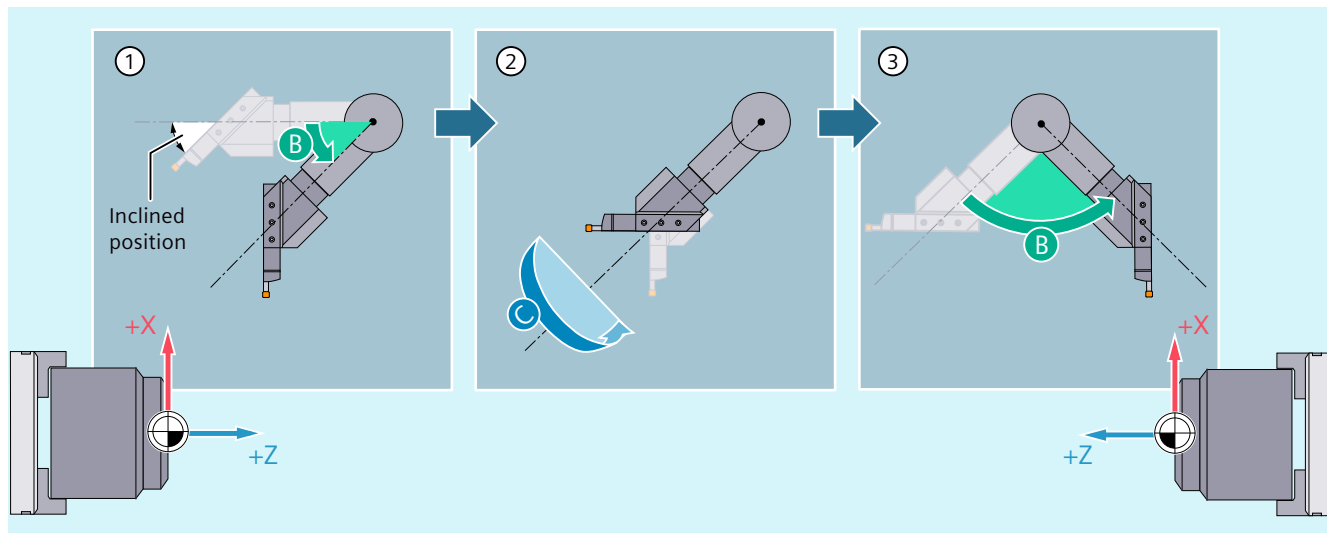
Tool parameter \$TC_DP19: Inclined position

On some turning tools, the tool cutting edge is mounted obliquely. This enables machining closer to the spindle.

The angle of the inclined position is entered in tool parameter \$TC_DP19. The cutting edge position, the reference direction, and the holder and clearance angles are entered for an inclined position of 0° .

For machining, the tool cutting edge must be put into a paraxial position via the orientation angles beta and gamma:

3.14 Modification of the offset data for rotatable tools



- ① Rotation of the B axis about beta for machining on the main spindle
- ② Rotation of the C axis about gamma
- ③ Rotation of the B axis about beta for machining on the counterspindle

Tool parameter \$TC_DPROT: Clamping angle

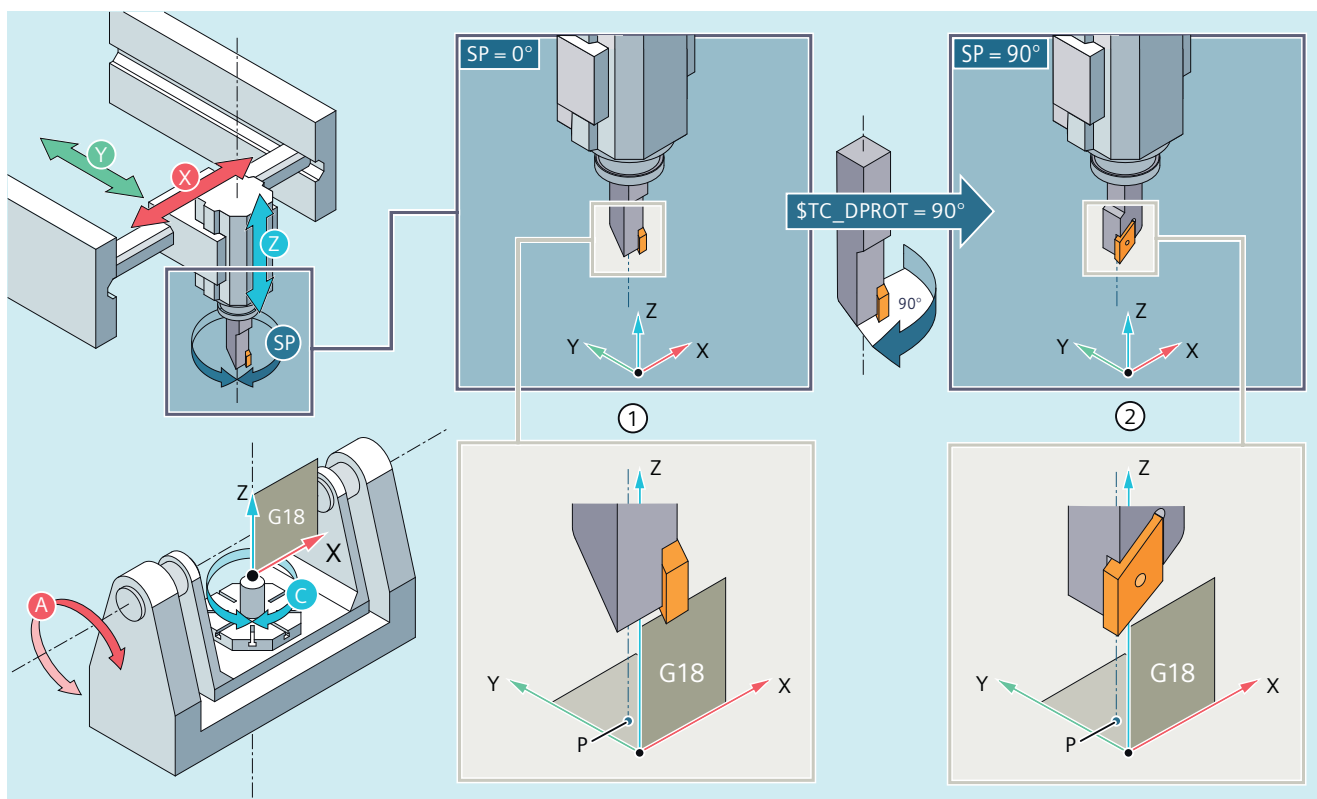
In certain applications, it is necessary to specify the clamping angle via the tool parameter \$TC_DPROT. This is the angle by which the tool spindle must rotate out of the spindle zero position to bring the clamped turning tool into the position in which the tool is to be dimensioned.

Application: Conventional turning tools in combination with the function "Turning on milling machine (TRAORI_STAT)" or "Interpolation turning (TRAINT)"

\$TC_DPROT must be set when the clamped turning tool in the spindle zero position is not in the G18 plane. The angle must be specified through which the tool spindle has to be rotated out of the spindle zero position to move the tool into the correct position for machining in the G18 plane.

The following figure illustrates this using the example "Interpolation turning (TRAINT)":

3.14 Modification of the offset data for rotatable tools



- ① In the spindle zero position ($SP = 0^\circ$), the cutting edge of the clamped turning tool is not in the G18 plane.
 - ② Only after rotation of the tool spindle through the clamping angle specified in tool parameter $\$TC_DPROT$ (in this case: $+90^\circ$) is the tool cutting edge in the correct position for machining in the G18 plane.
- P Imaginary point of intersection of the tool spindle axis with the XY plane

Application: Y-axis turning tools on the turning machine in conjunction with CYCLE805

$\$TC_DPROT$ must be set when the clamped Y turning tool in the basic position is not in the position in which the tool is to be dimensioned. Specify the angle by which the tool must be rotated from the basic position in order to bring it into the correct position for dimensioning.

3.14 Modification of the offset data for rotatable tools

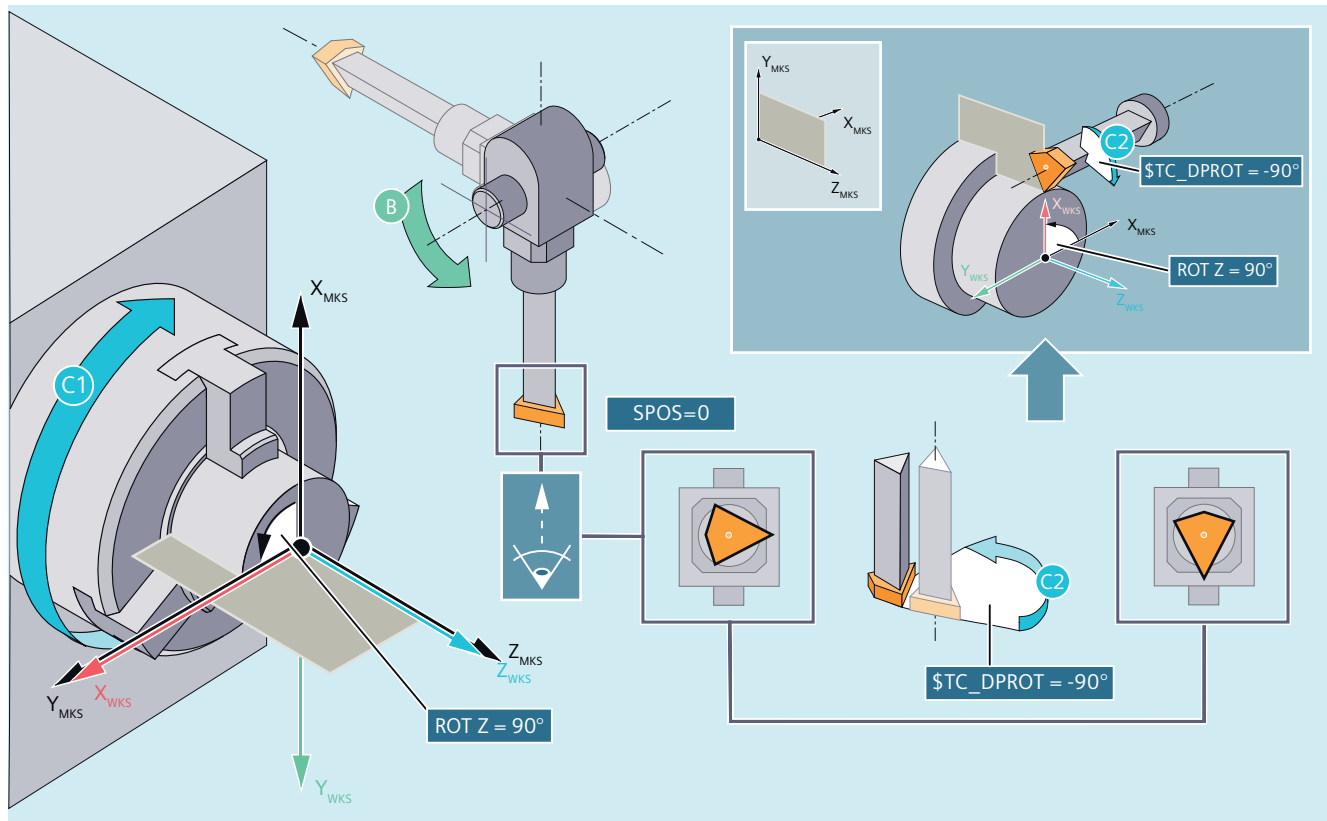


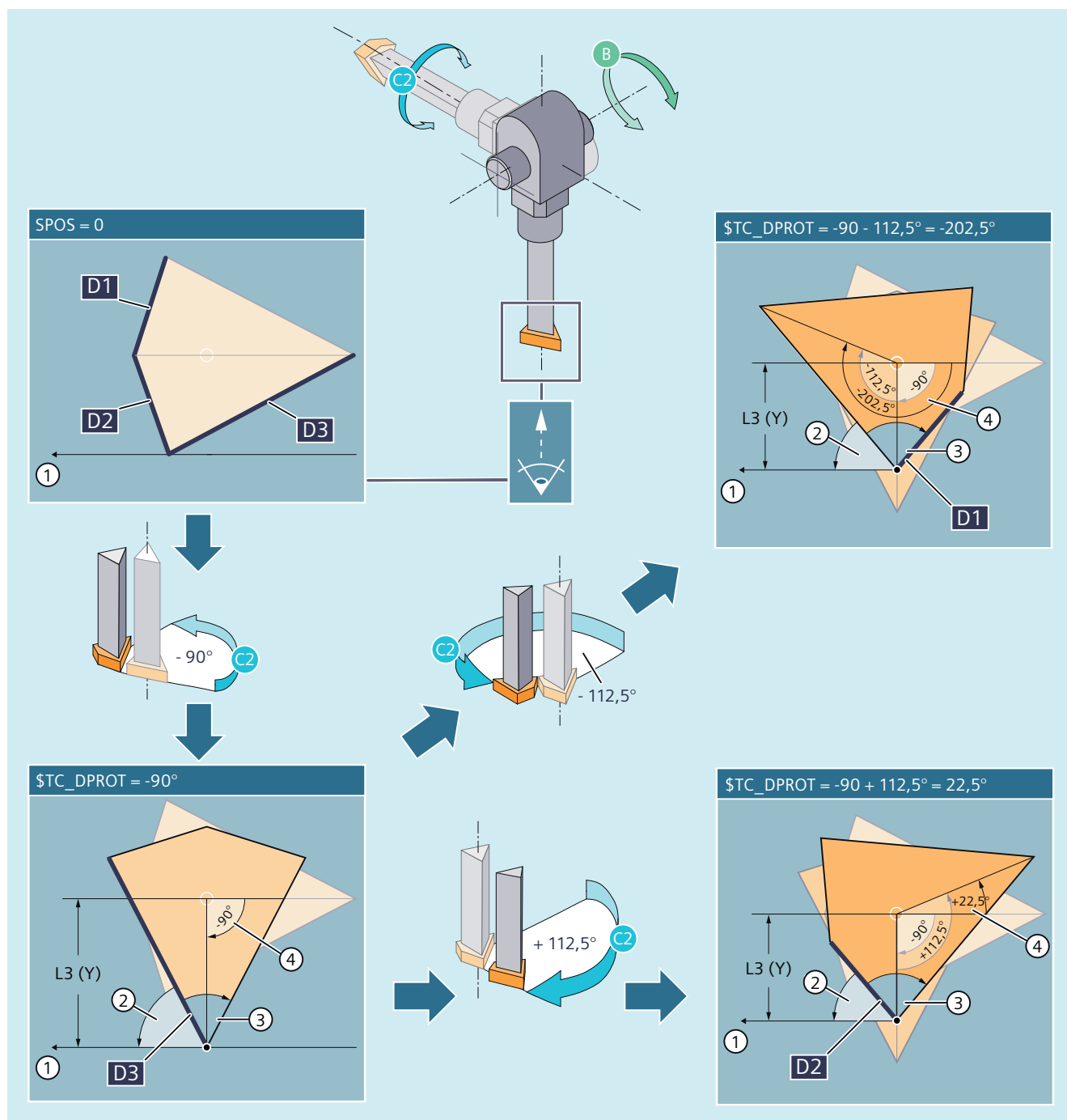
Figure 3-43 Y turning in working range 2 (WCS rotation around Z: +90°): Tool positioning via the setting angle $\$TC_DPROT$

Example:

On a Y axis turning tool with three cutting edges (D1 ... D3), all three cutting edges are to be used. For each cutting edge, the clamping angle $\$TC_DPROT$ required for dimensioning must be entered:

Cutting edge	Clamping angle $\$TC_DPROT$	
D1	$-90^\circ + (-112.5^\circ)$	$= -202.5^\circ$
D2	$-90^\circ + 112.5^\circ$	$= 22.5^\circ$
D3	$-90^\circ + 0^\circ$	$= -90^\circ$

3.14 Modification of the offset data for rotatable tools

Figure 3-44 Clamping angle $\$TC_DPROT$ of a Y turning tool with three cutting edges

3.14.3 Rotation of milling and drilling tools

3.14.3.1 Cutting edge position for milling and tapping tools

Milling and drilling tools

Milling and drilling tools means the following tools whose tool type (\$TC_DP1) has values in the range of 100 to 299.

Tools are treated independently of tool type such as milling and tapping tools if:

SD42950 \$SC_TOOL_LENGTH_TYPE = 1

Cutting edge position

A cutting edge position which is modified accordingly has also been introduced for the milling and tapping tools defined as specified above (see Chapter "Modifications during rotation of milling and tapping tools (Page 195)").

Note

A cutting edge position that is declared for tools that are not milling or tapping tools, or turning tools according to the definitions stated above, is not evaluated.

The cutting edge position of the tapping and milling tools is stored in tool parameter \$TC_DP2 as in the case of turning tools. Based on the definition of the cutting edge position for turning tools, this parameter can assume the values 5 to 8. Here, the cutting edge position specifies the orientation (the direction of the rotation axis) of the tool:

Cutting edge position	Direction of rotation axis of tool
5	Abcissa +
6	Ordinate +
7	Abcissa -
8	Ordinate -

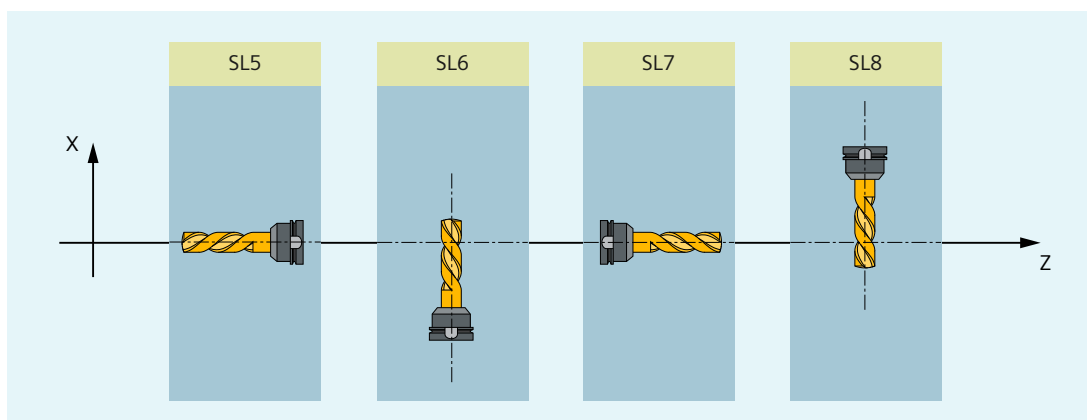


Figure 3-45 Cutting edge position 5 - 8 of a milling tool

3.14.3.2 Modifications during rotation of milling and tapping tools

The cutting edge position is recalculated appropriately during a rotation of a milling or tapping tool. Cut direction and tool angle (clearance angle or holder angle) are not defined for milling and tapping tools so that the change in cutting edge position is derived exclusively from the rotation. Thus, for milling and tapping tools, the cutting edge position always changes when the amount of rotation with reference to the zero setting is more than 45°.

3.14.4 Commissioning

3.14.4.1 Parameter assignment

Response in the event of an error

Different fault conditions can occur during the activation of the function "Modification of the offset data for rotatable tools" (by an explicit call in the part program or by tool selection). For errors with error numbers < 100 (see Diagnostics Manual), it is possible to define whether the error will trigger output of an alarm and, if so, whether such an alarm will only be displayed or will also trigger program stop. The setting is made in the following machine data:

MD20125 \$MC_CUTMOD_ERR

Two bits of the machine data are assigned to each fault condition:

Fault status	Bit	Meaning
No valid cutting direction is defined for the active tool.	0	Alarm output
	1	Program stop
The cutting edge angle (clearance angle and holder angle) of the active tool are both zero.	2	Alarm output
	3	Program stop
The clearance angle of the active tool has an impermissible value (< 0° or > 180°).	4	Alarm output
	5	Program stop

3.14 Modification of the offset data for rotatable tools

Fault status	Bit	Meaning
The holder angle of the active tool has an impermissible value ($< 0^\circ$ or $> 90^\circ$).	6	Alarm output
	7	Program stop
The tool insert angle of the active tool has an impermissible value ($< 0^\circ$ or $> 90^\circ$).	8	Alarm output
	9	Program stop
The cutting edge position - holder angle combination of the active tool is not permitted. (The holder angle must be $\leq 90^\circ$ for cutting edge position 1 to 4; for cutting edge positions 5 to 8 it must be $\geq 90^\circ$.)	10	Alarm output
	11	Program stop
The tool insert is not in the machining plane and angle γ between the tool insert and the machining plane exceeds the upper limit specified with setting data SD42998 \$SC_CUTMOD_PLANE_TOL.	12	Alarm output
	13	Program stop
The tool insert is not in the machining plane. The magnitude of angle α is greater than 1° . Angle α is the angle of rotation around the coordinate axis, which is perpendicular both to the axis of rotation of angle β and to the axis of rotation of angle γ (in the case of G18, the X axis).	14	Alarm output
	15	Program stop

Note

The 2nd bit (program stop) is only effective if the corresponding 1st bit (alarm output) is set.

Note

Errors with error numbers ≥ 100 always result in an alarm being output and program stop.

Initialization value for CUTMOD

The function that can be programmed with NC command CUTMOD (Page 198) is automatically initialized with the following value after a warm restart:

MD20127 \$MC_CUTMOD_INIT = <initialization value for CUTMOD>

Default value: 0

If MD20127 = -2, the value from the following machine data becomes the initialization value:

MD20126 \$MC_TOOL_CARRIER_RESET_VALUE (active tool carrier on reset)

Function-specific tool settings

The following machine data is a bit string with which various functions relating to tools can be controlled:

MD20360 \$MC_TOOL_PARAMETER_DEF_MASK

3.14 Modification of the offset data for rotatable tools

The following bits are relevant for function "Modification of the offset data for rotatable tools".

Bit	Meaning
17	Bit 17 is used to set whether, on modification of the offset data for turning and grinding tools, the cutting edge plane for calculating the offset values will be projected into the machining plane or rotated.
	= 0 The cutting edge plane is projected into the machining plane.
	= 1 The cutting edge plane is rotated into the machining plane.
	Note: Bit 17 is only evaluated if the function is active in combination with a kinematic orientation transformation.
18	Bit 18 is used to determine how the plane is defined for modification of offset data for turning and grinding tools.
	= 0 Priority is given to defining the plane with setting data SD42940 \$SC_TOOL_LENGTH_CONST over defining the plane with the G commands of G Group 6 (plane selection G17 - G19).
	= 1 The active plane (G17 - G19) is always used.
20	If tool parameters \$TC_DP10 (holder angle) and/or \$TC_DP24 (clearance angle) have the value "0", default values deviating from "0" can be used to calculate the modified cutting edge position and the modified cutting direction.
	= 0 The following default values are used: • Holder angle 112.5° for cutting edge positions 1 - 4 • Holder angle 67.5° for cutting edge positions 5 - 8 • Clearance angle 22.5° for cutting edge positions 1 - 4 • Clearance angle 67.5° for cutting edge positions 5 - 8
	= 1 An alarm is output.
	Note: Bit 20 is used to establish compatibility with older software versions.
21	Bit 21 is used to set whether the active total frame will be taken into account in the modification of the offset data when tool carrier with orientation capability is active.
	= 0 Rotation of the table component of the tool carrier is taken into account. Frames are ignored.
	= 1 Instead of the table component of the tool carrier, the active total frame is used. The total frame can also contain a table component of the tool carrier.

Difference between tool tip plane and machining plane for CUTMOD or CUTMODK

The maximum permissible angle through which the tool insert can be turned away from the machining plane (standard G18) when function CUTMOD or CUTMODK (Page 206) is called ($\neq$ maximum permissible deviation of γ from one of the two standard positions 0° or 180°) is set in the setting data:

SD42998 \$SC_CUTMOD_PLANE_TOL

Example:

3.14 Modification of the offset data for rotatable tools

SD42998 = 5.0 $\Rightarrow$ The tool insert must not be turned by more than 5° away from the machining plane.

Note

SD42998 = 0

If SD42998 \$SC_CUTMOD_PLANE_TOL is set to "0", variations of up to 89° are permissible!

Difference between tool tip plane and machining plane for ORISOLH

The maximum permissible angle through which the tool insert can be turned away from the machining plane when function ORISOLH (Page 206) is called is set in the setting data:

SD42999 \$SC_ORISOLH_INCLINE_TOL

3.14.5 Programming

3.14.5.1 Calculating orientations (ORISOLH)

The predefined ORISOLH function helps the user to set the rotary axis positions of a machine so that a turning tool can be brought into a defined, kinematic-independent position relative to the workpiece. Prerequisite is that a 6-axis transformation is active that has been parameterized with kinematic chains.

Two basic functions are available:

- Tool alignment
The β and γ angles are specified. The function calculates the angles of the three orientation axes required for this.
- Direct tool alignment
The angles of the second and third orientation axes are specified. The function calculates the associated β and γ angles as well as that of the missing first orientation axis.

Note

Order of the orientation axes

If you run through the kinematic chain that describes the structure of the machine, from the workpiece to the tool, then the following specifications apply for the order of the three orientation axes of a 6-axis transformation:

- The orientation axis that is closest to the **workpiece** is the **first** orientation axis.
- The orientation axis that is closest to the **tool** is the **third** orientation axis.

Generally, the first orientation axis is a spindle and the corresponding rotation is therefore implemented in these cases through a rotating frame.

3.14 Modification of the offset data for rotatable tools

Syntax

<RetVal> = ORISOLH(<Cntrl>,<W1>,<W2>)

Meaning

ORISOLH:	Function call			
<RetVal>:	Function return value			
	Data type:	INT		
	Range of values:	0, -2, -3, ..., -17		
	Values:	0	Function has ended without an error.	
		-2	No valid transformation (6-axis orientation transformation) is active.	
		-3	The first parameter (<Cntrl>) is negative.	
		-4	The unit position of the first parameter (<Cntrl>) is invalid. Only the values 0 and 1 are permissible.	
		-5	The tens position of the first parameter (<Cntrl>) is invalid. Only the values 0 to 3 are permissible.	
		-6	The hundreds position of the first parameter (<Cntrl>) is invalid. Only the values 0 and 1 are permissible.	
		-7	The thousands position of the first parameter (<Cntrl>) is invalid. Only the values 0 to 3 are permissible.	
		-8	Angle γ is too large for the "Direct tool alignment" function.	
		-9	At least one of the specified axis positions violates an axis limit for the "Direct tool alignment" function.	
		-10	No tool is active.	
		-11	The requested orientation cannot be set.	
		-12	The adaptation of the free axis angle for the Hirth joint is not possible for the first or only solution.	
		-13	The adaptation of the free axis angle for the Hirth joint is not possible for the second solution.	
		-14	The adaptation of the free axis angle for the Hirth joint is not possible for either of the two solutions.	
-15		The first orientation axis is parameterized as Hirth axis.		
-16		The second as well as the third rotary axis has been parameterized as Hirth axis. Only one of the two axes can be the Hirth axis.		
-17	At least one of the specified axis positions is not compatible with the associated Hirth joint for the "Swivel directly" function.			

3.14 Modification of the offset data for rotatable tools

<Cntrl>:	Controls the behavior of the function	
	Data type:	INT
	The <Cntrl> parameter is decimal coded (unit to thousands position):	
	Unit position:	The unit position controls the response to errors.
	xxx0	In the event of an error (return value < 0), alarm 14106 is output and program processing is aborted. Note: The alarm is also output irrespective of the value of the unit position when the <Cntrl> parameter is negative.
	xxx1	In the event of an error (return value < 0) no alarm is output. The user can react suitably in the program.
	Tens position:	Controls the behavior when an orientation axis with Hirth joint is present. Note: This parameter is only evaluated for the "Tool alignment" function (i.e. when the hundreds position has the value "0").
	xx0x	The axis position is rounded off to the nearest position.
	xx1x	The axis positions are rounded off so that the difference of the β angle to its programmed value is minimal.
	xx2x	The axis positions are rounded off so that the β angle is equal to the highest possible value which is less than the programmed value (β is rounded down).
	xx3x	The axis positions are rounded off so that the β angle is equal to the lowest possible value which is greater than the programmed value (β is rounded up).
	Hundreds position:	Specifies which function is to be executed or the significance of the two following parameters <W1> and <W2>.
	x0xx	"Tool alignment" function Parameters <W1> and <W2> have the following meaning: • <W1> = β • <W2> = γ The associated angles of the orientation axes are calculated.
	x1xx	"Direct tool alignment" function <W1> is the position specification for the second orientation axis, <W2> is the position specification for the third orientation axis of a 6-axis transformation. The position of the first orientation axis and the β and γ angles are defined which are compatible with the two position specifications. If no error occurs, two solutions are always output in the \$P_ORI_POS[<n>, <m>] system variables. The first index <n> (0 or 1) refers to the solution and the second index <m> (0 ... 2) to the orientation axis: • \$P_ORI_POS[0/1, 0]: Position of the first orientation axis • \$P_ORI_POS[0/1, 1]: Angle β • \$P_ORI_POS[0/1, 2]: Angle γ A check is made as to whether the position specifications <W1> and <W2> are compatible with any Hirth joints or active software limits. If this is not the case, a corresponding error number is returned (see <RetVal> parameter).

3.14 Modification of the offset data for rotatable tools

			If the angles <W1> and <W2> are selected arbitrarily, the cutting edge of the tool is generally not in the machining plane. The angle γ through which the cutting edge is rotated out of the machining plane, must not be greater than the limit value which is defined by the setting data SD42999 \$SC_ORI-SOLH_INCLINE_TOL.
	Thousands position:		Specifies which positions of the solutions may be modified when the hundreds position has the value "0", i.e. for the "Tool alignment" function.
		0xxx	The calculated axis positions should be as close as possible to the current machine axis positions.
		1xxx	The calculated axis positions for modulo axes should be as close as possible to the middle of the modulo range, for other axes as close as possible to 0. For non-modulo axes, this means that the axis positions are reduced to the range $-180^\circ \dots +180^\circ$.
		2xxx	The calculated axis positions should be reduced to the range $-180^\circ \dots +180^\circ$ irrespective of the axis type.
<W1>:	First angle	The meaning results from the hundreds position of the <Cntrl> paramter.	
	Data type:	REAL	
<W1>:	Second angle	The meaning results from the hundreds position of the <Cntrl> paramter.	
	Data type:	REAL	

Note

Parameters that have not been programmed have the default value "0".

Further information

The number of solutions found together with further status information when executing the ORISOLH function, can be read via the following system variables:

System variable	Meaning	
\$P_ORI_POS [<n>, <m>]	Returns the angles of the orientation axes that result from the orientation programming.	
	<n>:	Index of the solution
	Range of values:	0, 1
	<m>:	Index of the orientation axis
	Range of values:	0 ... 2 The order of the orientation axes (1 ... 3) refers to the definition of the axes in \$NT_ROT_AX_NAME.
When the ORISOLH function is called in the "Direct tool alignment" mode, the \$P_ORI_POS[0/1, 1] and P_ORI_POS[0/1, 2] variables contain the values of the two angles β and γ belonging to the two solutions. The first solution entered in \$P_ORI_POS[<n>, <m>], i.e. with the index <n> = 0, is always the solution that is selected by the control when the requested orientation is approached directly. The second index <m> refers to the orientation axis, i.e. on \$NT_ROT_AX_NAME. The axis positions entered in \$P_ORI_POS[<n>, <m>] take into account the offsets entered in \$NK_OFF and \$NK_OFF_FINE, i.e. these axis angles can be used in the following blocks to set the required orientation without any further modification. If a rotary axis is a Hirth axis, the solution positions are rounded off to the nearest position of rest of the Hirth joint. For Hirth jointed rotary axes, you can read the differences between the axis positions for the exact solutions and those of the solutions adapted to the Hirth incrementing in the \$P_ORI_DIFF system variable.		
\$P_ORI_DIFF [<n>, <m>]	Returns the difference between the exact positions of the orientation axes and those provided in \$P_ORI_POS that result from the orientation programming.	
	<n>:	Index of the solution
	Range of values:	0, 1
	<m>:	Index of the orientation axis
	Range of values:	0 ... 2 The order of the orientation axes (1 ... 3) refers to the definition of the axes in \$NT_ROT_AX_NAME.
The content can only be not equal to zero when the positions are incremented (Hirth joint), i.e. when the system data \$NT_HIRTH_INCR of the relevant axis is not equal to zero and when this axis is a manual rotary axis.		

3.14 Modification of the offset data for rotatable tools

System variable	Meaning		
\$P_ORI_SOL	If for an orientation transformation with more than one orientation axis, the axis angles are calculated that should result in a specified orientation, there is generally more than one solution. The \$P_ORI_SOL system variables contain the number of valid solutions together with additional status information. The content of \$P_ORI_SOL is coded as follows:		
	Values < 0	General error states	
		-1	No solutions have been calculated yet for the active transformation (missing call of ORISOLH).
		-2	A transformation is not active, or the active transformation is not an orientation transformation (6-axis transformation) that can provide positions for a specified orientation programming.
		-4	The desired orientation cannot be set with the present kinematics.
		-5	No solution was found when the ORISOLH function was called in the "Direct tool alignment" mode.
		-6	Angle γ is too large when the ORISOLH function was called in the "Direct tool alignment" mode.
		-7	An angle was specified when the ORISOLH function was called in the "Direct tool alignment" mode that cannot be set because of the Hirth joint.
		-8	The first orientation axis (frame axis) must not be parameterized as Hirth axis.
		-9	The second as well as the third rotary axis has been parameterized as Hirth axis. Only one of the two axes can be the Hirth axis.
		-10	No adaptation of the solution(s) to the Hirth joint has been found.
	Values > 0	Number of mathematically possible solutions without consideration of axis limits and any error conditions.	
	Unit position	<table><tr><td>0</td><td>There is no solution, i.e. the requested orientation cannot be set. There can be three different causes for this case: - In principle, the requested orientation cannot be achieved because of the machine kinematics (orientation axes not arranged at right angles) even with an arbitrary traversing range of the orientation axes. In this case, the tens and hundreds positions of \$P_ORI_SOL are both zero, the \$P_ORI_STAT status variables assigned to the orientation axis have the value "-4". - The calculated solutions cannot be achieved because they would violate the axis limits. The positions of the orientation axes that would result without the axis limits, can be read in \$P_ORI_POS. - Axis positions were specified when the ORISOLH function was called in the "Direct tool alignment" mode which would result in either the orientation vector or the orientation normal vector of the tool being aligned parallel to the first orientation axis, whose position is to be calculated. The position of this axis is not defined in these cases. </td></tr></table>	0
0	There is no solution, i.e. the requested orientation cannot be set. There can be three different causes for this case: - In principle, the requested orientation cannot be achieved because of the machine kinematics (orientation axes not arranged at right angles) even with an arbitrary traversing range of the orientation axes. In this case, the tens and hundreds positions of \$P_ORI_SOL are both zero, the \$P_ORI_STAT status variables assigned to the orientation axis have the value "-4". - The calculated solutions cannot be achieved because they would violate the axis limits. The positions of the orientation axes that would result without the axis limits, can be read in \$P_ORI_POS. - Axis positions were specified when the ORISOLH function was called in the "Direct tool alignment" mode which would result in either the orientation vector or the orientation normal vector of the tool being aligned parallel to the first orientation axis, whose position is to be calculated. The position of this axis is not defined in these cases.		

3.14 Modification of the offset data for rotatable tools

System variable	Meaning	
		1 There is a solution. There can be three different causes for this case: - Based on the specified orientation and the machine kinematics, there is only one solution (from the mathematical point of view, two coinciding solutions) even without consideration of the axis limits. This case occurs at the edge of the orientation range for kinematics that are not at right angles. \$P_ORI_POS contains both (identical) solutions. - There is only one solution because a second solution is invalid due to the violated axis limits. The valid solution is always the first solution in \$P_ORI_POS. The second solution which would result when the axis limits are not taken into account, can also be read in \$P_ORI_POS. - This is the normal case when the ORISOLH function is called in the "Direct tool alignment" mode. For the specified axis positions of two orientation axes, there is generally only one valid position for the missing orientation axis to be calculated.
		2 There are two solutions.
		8 There are an infinite number of solutions, i.e. the position of an orientation axis (the polar axis) is arbitrary. However, from the two possible positions of the other axes, one is excluded because of the violated axis limits.
		9 There are an infinite number of solutions, i.e. the position of an orientation axis (the polar axis) is indefinite. The indefinite axis can be determined from the hundreds position or from the \$P_ORI_STAT system variable.
	Values > 0 Tens position	Bit-coded display for violated axis limits. The precise cause of the error can be determined from the \$P_ORI_STAT system variable.
		Bit 0 (value 10): For at least one solution, at least one axis limit of the first orientation axis is violated.
		Bit 1 (value 20): For at least one solution, at least one axis limit of the second orientation axis is violated.
		Bit 2 (value 40): For at least one solution, at least one axis limit of the third orientation axis is violated.
	Values > 0 Hundreds position	Bit-coded display for non-defined axis positions (can only occur when there is an infinite number of solutions, i.e. when the unit position is equal to "9").
		Bit 0 (value 100): The position of the first orientation axis is not defined.
		Bit 1 (value 200): The position of the second orientation axis is not defined.
		Bit 2 (value 400): The position of the third orientation axis is not defined.
	The designations first, second and third orientation axis refer to the definition of the axes in \$NT_ROT_AX_NAME.	

3.14 Modification of the offset data for rotatable tools

System variable	Meaning	
\$P_ORI_STAT [<n>]	Returns the status for each of the maximum three orientation axes after ORISOLH has been called.	
	<n>:	Index of the orientation axis (corresponds to the index of the relevant orientation axis in \$NT_ROT_AX_NAME)
		Range of values: 0 ... 2
		The order of the orientation axes (1 ... 3) refers to the definition of the axes in \$NT_ROT_AX_NAME.
	The content of \$P_ORI_STAT is coded as follows:	
	Values < 0	General error states
		-1 The status is not defined (missing call of ORISOLH).
		-2 A transformation is not active, or the active transformation is not an orientation transformation (6-axis transformation) that can provide positions for a specified orientation programming.
		-3 The axis is not included in the active transformation.
		-4 The position of the axis cannot be calculated because the requested orientation cannot be achieved with the present kinematics even with an arbitrary assumed traversing range of the axis.
		-5 Axis positions were specified when the ORISOLH function was called in the "Direct tool alignment" mode which would result in either the orientation vector or the orientation normal vector of the tool being aligned parallel to the first orientation axis, whose position is to be calculated. The position of this axis is not defined in these cases.
		-6 Angle γ is too large when the ORISOLH function was called in the "Direct tool alignment" mode.
		-7 An angle was specified when the ORISOLH function was called in the "Direct tool alignment" mode that cannot be set because of the Hirth joint.
		-8 The first orientation axis (frame axis) must not be parameterized as Hirth axis.
		-9 The second as well as the third rotary axis has been parameterized as Hirth axis. Only one of the two axes can be the Hirth axis.
		-10 No adaptation of the solution(s) to the Hirth joint has been found.
	Values > 0	Bit-coded display for violated axis limits of the first solution.
	Unit position	Bit 0 (value 1): The first solution violates the lower axis limit.
		Bit 1 (value 2): The first solution violates the upper axis limit.
	Values > 0	Bit-coded display for violated axis limits of the second solution.
	Tens position	Bit 0 (value 10): The second solution violates the lower axis limit.
		Bit 1 (value 20): The second solution violates the upper axis limit.
	Values > 0	Display of a non-defined axis position.
	Hundreds position	Bit 0 (value 100): The position of the orientation axis is not defined, i.e. the requested orientation is achieved with each arbitrary setting of the rotary axis (polar po-

3.14 Modification of the offset data for rotatable tools

System variable	Meaning
	sition). This information is also contained in the \$P_ORI_SOL system variable. Of the error numbers that indicate a violation of the axis limits, several can occur simultaneously. When an axis limit is violated, an attempt is made to reach a position within the permissible axis limits by adding or subtracting multiples of 360°. If this is not possible, it is not clearly defined whether the lower or the upper axis limit has been violated. If there is no solution for the requested orientation (\$P_ORI_SOL = 0), the status of the orientation axes in the transformation is "0".

Note**\$NT_ROT_AX_NAME**

This system variable refers to a maximum of three axes used for setting the orientation. It contains the names of the chain elements (\$NK_NAME) that define the machine axes (rotary axes) that must perform the orientation movements resulting from a kinematic transformation. The order in which the maximum three rotary axes are contained in this system variable is irrelevant for the machine kinematics because this is derived from the structure of the kinematic chains. However, as it defines the order in which other variables access the rotary axes, the order of the orientation axes in \$NT_ROT_AX_NAME must match the kinematic description.

Note**Status information**

The status information that shows, for example, that an orientation cannot be achieved or can only be achieved when relevant axis limits are violated, does not trigger an NC alarm. It is the responsibility of the user to react suitably to the specified conditions.

3.14.5.2 Activating the modification of the offset data for rotatable tools (CUTMOD, CUTMODK)

The modification of the offset data for rotatable tools is activated in the NC program via the CUTMOD (in combination with orientable tool carriers) or CUTMODK language command (for orientation transformations that were defined by means of kinematic chains).

Note

As the orientable tool carriers and orientation transformations that were defined by means of kinematic chains cannot be active at the same time, there are no conflicts between the two variants.

Syntax

CUTMOD = <Value>

or

3.14 Modification of the offset data for rotatable tools

CUTMODK = <Command>

Meaning

CUTMOD:	Function call in combination with orientable tool carriers		
<Value>:	Assigned value		
	Data type:	INT	
	Value:	0	The function is deactivated. The values supplied from system variables \$P_AD... are the same as the corresponding tool parameters.
		> 0	The function is activated if an orientable tool carrier with the specified number is active, i.e. the activation is linked to a specific orientable tool carrier. The values supplied from system variables \$P_AD... may be modified with respect to the corresponding tool parameters depending on the active rotation. The deactivation of the designated orientable tool carrier temporarily deactivates the function; the activation of another orientable tool carrier permanently deactivates it. This is the reason why in the first case, the function is re-activated when again selecting the same orientable tool carrier; in the second case, a new selection is required - even if at a subsequent time, the orientable tool carrier is re-activated with the specified number. The function is not influenced by a reset.
		-1	The function is always activated if an orientable tool carrier is active. When changing the tool carrier or when de-selecting it and a subsequent new selection, CUTMOD does not have to be set again.
		-2	The function is always activated if an orientable tool carrier is active whose number is the same as the currently active orientable tool carrier. If an orientable tool carrier is not active, then this has the same significance as CUTMOD=0. If an orientable tool carrier is active, then this has the same significance as when directly specifying the actual tool carrier number.
		-3	The function is only activated if no orientable tool carrier is active. The values supplied from system variables \$P_ADT... may be modified with respect to the corresponding tool parameters depending on the active rotation. The activation of the designated orientable tool carrier deactivates the function. The function is not influenced by a reset.
		< -3	Values less than -3 are ignored. I.e. this case is treated as if CUTMOD was not programmed. Note: This value range should not be used as it is reserved for possible subsequent expansions.
CUTMODK:	Function call in combination with orientation transformations that have been defined by means of kinematic chains		

3.14 Modification of the offset data for rotatable tools

<Command>:	Assigned Command		
	Data type:	STRING	
	Value:	"NEW"	The states of an active transformation defined with kinematic chains relevant for the "Modification of the offset data", the name of the transformation and the current contour frame are saved. Note: This command is only permissible when a suitable transformation (TRAORI_DYN, TRAORI_STAT or TRAANG_K) is active.
		"OFF"	Switches the active "Modification of the offset data" off. The data previously stored with "NEW" is retained. Note: This command is also permissible when CUTMODK is not active. It then remains without effect. Any data set present for the "Modification of the offset data" is retained.
		"ON"	With this command, the "Modification of the offset data" is re-activated with a data set previously stored with the "NEW" command. If a transformation with the name of the stored data set is active when this command is executed, the "Modification of the offset data" takes effect immediately. Otherwise, the activation is delayed until an active transformation is activated.
		"CLEAR"	As with the "OFF" command, switches the "Modification of the offset data" off and also deletes the stored data set. Note: This command is also permissible when CUTMODK is not active.

Note**SD42984 \$SC_CUTDIRMOD**

The CUTMOD or CUTMODK command replaces the function that can be activated using the setting data SD42984 \$SC_CUTDIRMOD. However, this function remains available unchanged. However, as it doesn't make sense to use both functions in parallel, it can only be activated if CUTMOD is equal to zero and CUTMODK is the zero string.

More information**Reading modified offset data**

The modified offset data is provided in the following system variables and OPI variables:

Meaning	System variable	OPI variable
Cutting edge position	\$P_AD[2]	cuttEdgeParam2
Holder angle	\$P_AD[10]	cuttEdgeParam10
Cut direction	\$P_AD[11]	cuttEdgeParam11
Clearance angle	\$P_AD[24]	cuttEdgeParam24

The data is always modified with respect to the corresponding tool parameters (\$TC_DP2[...], ...) etc.) when the "Modification of the offset data for rotatable tools" function

3.14 Modification of the offset data for rotatable tools

was activated with the CUTMOD or CUTMODK command and the tool was rotated by an orientable tool carrier or a suitable orientation transformation.

Further function-relevant system variables

System variable	Meaning
\$P_CUTMOD_ANG / \$AC_CUTMOD_ANG	Returns the angle through which a tool was rotated in the active machining plane and the modified cutting edge data available for the CUTMOD and CUTMODK functions.
\$P_CUTMOD / \$AC_CUTMOD	Reads the currently valid value that was last programmed with the CUTMOD command (number of the tool carrier for which the modification of the offset data should be activated). If the last programmed value was <code>CUTMOD = -2</code> (activation with the currently active orientable tool carrier), then the value "-2" is not returned in the system variable, but rather the number of the orientable tool carrier active at the time of programming.
\$P_CUTMODK / \$AC_CUTMODK	Reads the name of the transformation under which the currently valid data set for the "Modification of the offset data" was created.
\$P_CUT_INV / \$AC_CUT_INV	Supplies the value TRUE if the tool is rotated so that the spindle direction of rotation must be inverted. To do this, the following four conditions must be fulfilled in the block to which the read operations refer: 1. If a turning or grinding tool is active (tool types 400 to 599 and / or SD42950 \$SC_TOOL_LENGTH_TYPE = 2). 2. The modification of the offset data was activated with the CUTMOD or CUTMODK command. 3. An orientable tool carrier or an orientation transformation defined with kinematic chains is active, which was selected with the CUTMOD or CUTMODK command. 4. The tool is rotated by the orientable tool carrier or the kinematic orientation transformation so that the resulting normal of the tool cutting edge is rotated with respect to the initial position by more than 90° (typically 180°). If at least one of the specified four conditions is not fulfilled, the variable returns the value FALSE. For tools whose cutting edge position is not defined, the value of the variable is always FALSE.

3.14 Modification of the offset data for rotatable tools

System variable	Meaning
\$P_CUTMOD_ERR	Error state after the last call of the CUTMOD function The CUTMOD function can also be called implicitly for a tool change. At a reset, the variable is reset to zero. It is reset at every tool change and, if required, rewritten. The variable is bit-coded. The bits have the following meanings:
	Bit 0: No valid cut direction is defined for the active tool.
	Bit 1: The cutting edge angle (clearance angle and holder angle) of the active tool are both zero.
	Bit 2: The clearance angle of the active tool has an impermissible value ($< 0^\circ$ or $> 180^\circ$).
	Bit 3: The holder angle of the active tool has an impermissible value ($< 0^\circ$ or $> 90^\circ$).
	Bit 4: The plate angle of the active tool has an impermissible value ($< 0^\circ$ or $> 90^\circ$).
	Bit 5: The cutting edge position - holder angle combination of the active tool is not permitted (the holder angle must be $\leq 90^\circ$ for cutting edge position 1 to 4; for cutting edge positions 5 to 8 it must be $\geq 90^\circ$).
	Bit 6: Illegal rotation of the active tool. The tool was rotated out of the active machining plane by $\pm 90^\circ$ (with a tolerance of about 1°). The cutting edge position is therefore no longer defined in the machining plane.
	Bit 7: The cutting plate is not in the machining plane and the angle between the cutting plate and the machining plane exceeds the upper limit specified with the setting data SD42998 \$SC_CUTMOD_PLANE_TOL.
	Bit 8: The cutting plate is not in the machining plane. Angle α is greater than 1° . Angle α is the angle of rotation around the coordinate axis which is perpendicular to the axis of rotation of angle β as well as to the axis of rotation of angle γ (the X axis for G18).
	Bit 9: Orientable tool carrier is active when CUTMOD = -3.

\$P_...: Preprocessing variables

\$AC_...: Main run variables

All main run variables can be read in synchronized actions. A read access operation from the preprocessing generates a preprocessing stop.

Plane change

To determine the modified cutting edge position, cutting direction and holder or clearance angle, the evaluation of the cutting edge in the active plane (G17 - G19) is decisive.

However, if setting data SD42940 \$SC_TOOL_LENGTH_CONST (change of the tool length component when selecting the plane) has a valid non-zero value (plus or minus 17, 18 or 19), its contents define the plane in which the relevant quantities are evaluated.

This priority rule of the setting data over the G code can be deactivated by setting bit 18 of the machine data \$MC_TOOL_PARAMETER_DEF_MASK. This means that when this bit is set, the plane defined with the G command of group 6 is still valid.

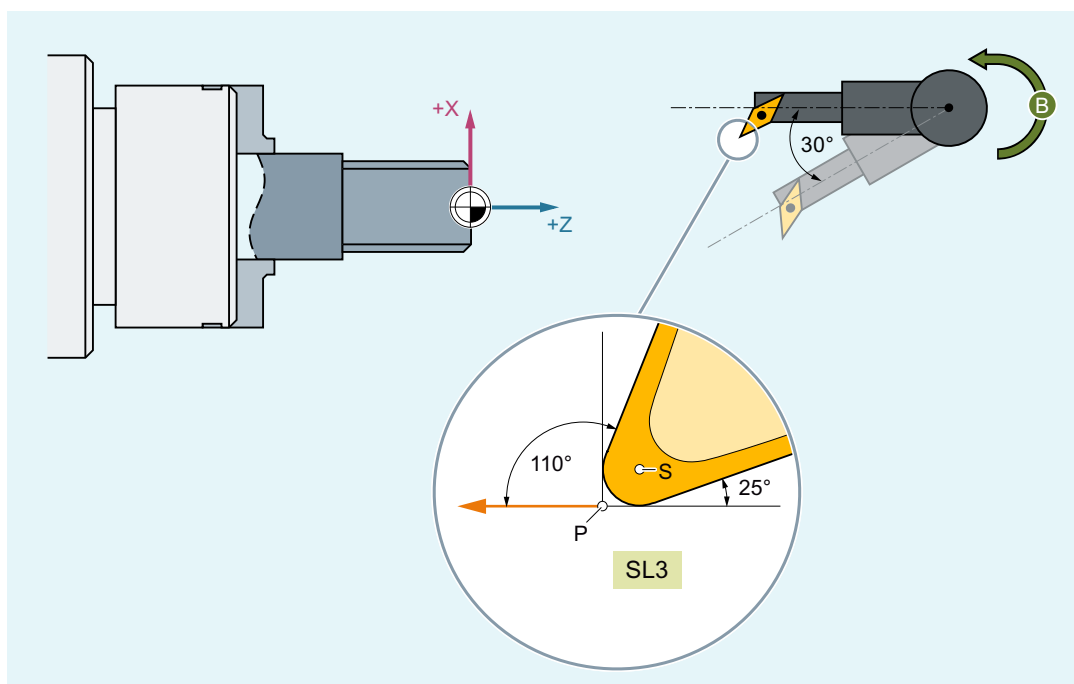
Effectiveness of the modified cutting data

The modified cutting edge position and the modified cutting edge reference point are immediately effective when programming, even for a tool that is already active. A tool does not have to be re-selected for this purpose.

3.14.6 Example

Example

For a tool with cutting edge position 3 and an orientable tool carrier that can rotate the tool around the B axis, the cutting edge position shall be modified after a tool rotation with the aid of the CUTMOD command.



- S: Cutting edge center point
- P: Cutting edge reference point
- SL: Cutting edge position

Program code	Comment
N10 \$TC_DP1[1,1]=500	
N20 \$TC_DP2[1,1]=3	;Cutting edge position
N30 \$TC_DP3[1,1]=12	
N40 \$TC_DP4[1,1]=1	
N50 \$TC_DP6[1,1]=6	
N60 \$TC_DP10[1,1]=110	; Holder angle
N70 \$TC_DP11[1,1]=3	; Cut direction
N80 \$TC_DP24[1,1]=25	; Clearance angle

3.14 Modification of the offset data for rotatable tools

Program code	Comment		
N90 \$TC_CARR7[2]=0 \$TC_CARR8[2]=1 \$TC_CARR9[2]=0	; B axis		
N100 \$TC_CARR10[2]=0 \$TC_CARR11[2]=0	; C axis		
\$TC_CARR12[2]=1			
N110 \$TC_CARR13[2]=0			
N120 \$TC_CARR14[2]=0			
N130 \$TC_CARR21[2]=X			
N140 \$TC_CARR22[2]=X			
N150 \$TC_CARR23[2]="M"			
N160 TCOABS CUTMOD=0			
N170 G18 T1 D1 TCARR=2	; X	Y	Z
N180 X0 Y0 Z0 F10000	; 12.000	0.000	1.000
N190 \$TC_CARR13[2]=30			
N200 TCARR=2			
N210 X0 Y0 Z0	; 10.892	0.000	-5.134
N220 G42 Z-10	; 8.696	0.000	-17.330
N230 Z-20	; 8.696	0.000	-21.330
N240 X10	; 12.696	0.000	-21.330
N250 G40 X20 Z0	; 30.892	0.000	-5.134
N260 CUTMOD=2 X0 Y0 Z0	; 8.696	0.000	-7.330
N270 G42 Z-10	; 8.696	0.000	-17.330
N280 Z-20	; 8.696	0.000	-21.330
N290 X10	; 12.696	0.000	-21.330
N300 G40 X20 Z0	; 28.696	0.000	-7.330
N310 M30			

The numerical values in the comments specify the end of block positions in the machine coordinates (MCS) in the sequence X → Y → Z.

Explanations

In block N180, initially the tool is selected for CUTMOD=0 and non-rotated tool holders that can be orientated. As all offset vectors of the tool holder that can be orientated are 0, the position that corresponds to the tool lengths specified in \$TC_DP3[1,1] and \$TC_DP4[1,1] is approached.

The tool holder that can be orientated with a rotation of 30° around the B axis is activated in block N200. As the cutting edge position is not modified due to CUTMOD=0, the old cutting edge reference point is decisive just as before. This is the reason why in block N210 the position is approached, which keeps the old tool nose reference point at the zero (i.e. the vector (1, 12) is rotated through 30° in the Z/X plane).

In block N260, contrary to block N200, CUTMOD=2 is effective. As a result of the tool holder rotation that can be orientated, the modified cutting edge position becomes 8. Deviating axis positions also result from this.

The tool radius compensation (TRC) is activated in blocks N220 and/or N270. The different cutting edge position in both program sections has no effect on the end positions of the blocks in which the TRC is active; the corresponding positions are therefore identical. The different cutting edge positions only become effective again in the deselect blocks N260 and/or N300.

3.15 Incrementally programmed compensation values

3.15.1 G91 extension

Requirements

Incremental dimension programming with G91 is defined such that the offset value is traversed additively to the incrementally programmed value when a tool offset is selected.

Applications

For applications such as scratching, it is necessary only to traverse the path programmed in the incremental dimension. The activated tool offset is not traversed.

Sequence

Selection of a tool offset with incremental dimension programming

- Scratch workpiece with tool tip.
- Adopt the actual position in the basic frame (set actual value) after reducing it by the tool offset.
- Traverse incrementally from the zero position.

Activation

It is possible to set whether a changed tool length is traversed with FRAME and incremental dimension programming of an axis, or whether only the programmed path is traversed with the setting data:

SD42442 \$SC_TOOL_OFFSET_INCR_PROG (tool length compensations)

Work offset / frames G91

It is possible to set whether a work offset is traversed as standard with value = 1 with FRAME and incremental dimension programming of an axis, or whether only the programmed path is traversed with value = 0 with the setting data:

SD42440 \$SC_FRAME_OFFSET_INCR_PROG (work offset in frames)

3.15 Incrementally programmed compensation values

For further information, see Function Manual "Basic Functions", Chapter "Axes, coordinate systems, frames".

Constraint

If the behavior is set such that the offset remains active after the end of the program and `RESET MD20110 $MC_RESET_MODE_MASK, bit6=1` (definition of basic control settings after reset/PP end) and if an incremental path is programmed in the first part program, the offset is always traversed in addition to the programmed path.

Note

With this configuration, part programs should always begin with absolute programming.

3.15.2 Traversing in the direction of tool orientation (MOVT)

Typical application

On machines with toolholders with orientation capability, traversing should take place in the tool direction (typically, when drilling) without activating a frame (e.g., using `TOFRAME` or `TOROT`), on which one of the axes points in the direction of the tool.

This is also true of machines on which a frame defining the oblique plane is active during oblique machining operations, but the tool cannot be set exactly perpendicular because an indexed toolholder (Hirth tooth system) is restricting the setting of the tool orientation.

In these cases it is then necessary - contrary to the motion actually requested perpendicular to the plane - to drill in the tool direction, as the drill would otherwise not be guided in the direction of its longitudinal axis, which, among other things, would lead to breaking of the drill.

MOVT

The end point of such a motion is programmed with `MOVT= . . .`. The programmed value is effective incrementally in the tool direction as standard. The positive direction is defined from the tool tip to the toolholder. The content of `MOVT` is thus generally negative for the infeed motion (when drilling), and positive for the retraction motion. This corresponds to the situation with normal paraxial machining, e.g., with `G91Z . . .`.

If the motion is programmed in the form `MOVT=AC ( . . . )`, `MOVT` functions absolutely. In this case a plane is defined, which runs through the current zero point, and whose surface normal vector is parallel to the tool orientation. `MOVT` then gives the position relative to this plane:

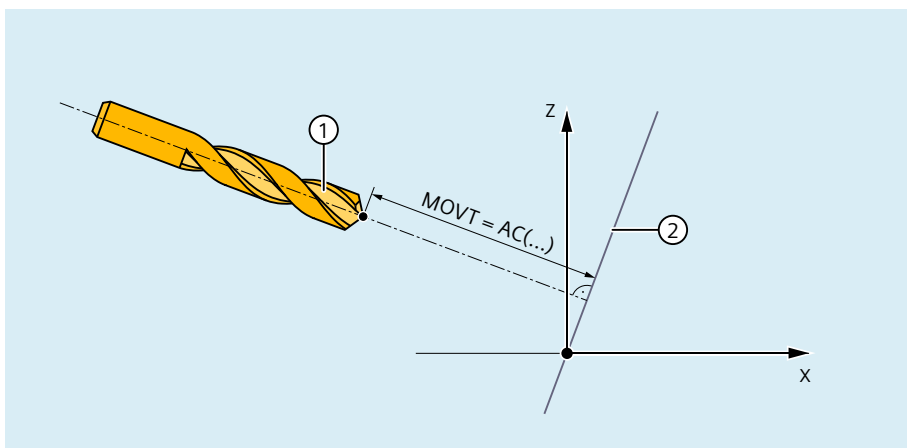


Figure 3-46 Definition of the position for absolute programming of a motion in tool direction

The reference to this auxiliary plane serves only to calculate the end position. Active frames are not affected by this internal calculation.

Instead of `MOVT= . . .` it is also possible to write `MOVT=IC ( . . . )` if it is to be plainly visible that `MOVT` is to function incrementally. There is no functional difference between the two forms.

Supplementary conditions

The following supplementary conditions apply to programming with `MOVT`:

- It is independent of the existence of a toolholder with orientation capability. The direction of the motion is dependent on the active plane. It runs in the direction of the vertical axes, i.e., with G17 in Z direction, with G18 in Y direction and with G19 in X direction. This applies both where no toolholder with orientation capability is active and for the case of a toolholder with orientation capability without rotary tool or with a rotary tool in its basic setting.
- `MOVT` acts similarly for active orientation transformation (345axis transformation).
- If in a block with `MOVT` the tool orientation is changed simultaneously (e.g., active 5axis transformation by means of simultaneous interpolation of the rotary axes), the orientation at the start of the block is decisive for the direction of movement of `MOVT`. The path of the tool tip (TCP - Tool Center Point) is not affected by the change in orientation.
- Linear or spline interpolation (G0, G1, ASPLINE, BSPLINE, CSPLINE) must be active. Otherwise, an alarm is produced. If a spline interpolation is active, the resultant path is generally not a straight line, since the end point determined by `MOVT` is treated as if it had been programmed explicitly with X, Y, Z.
- A block with `MOVT` must not contain any programming of geometry axes (alarm 14157).

3.16 Assignment of tool length components to geometry axes

3.16.1 Assignment according to tool type and working plane.

The values of the tool parameters length 1 ... 3 are stored in the system variables \$TC_DP3 ... \$TC_DP5 (see Chapter "Tool parameters (Page 37)"). The assignment to the geometry axes and therefore the resultant cutting direction of the tool length components depends on the tool type (\$TC_DP1) and the active machining plane (G17/G18/G19).

Table 3-1 Turning/grinding tools (\$TC_DP1 = 400 ... 599)

Working plane	Length 1	Length 2	Length 3
G17 (X/Y)	Y	X	Z
G18 (Z/X)	X	Z	Y
G19 (Y/Z)	Z	Y	X

Table 3-2 Milling tools / special tools (\$TC_DP1 <> 400 ... 599)

Working plane	Length 1	Length 2	Length 3
G17 (X/Y)	Z	Y	X
G18 (Z/X)	Y	X	Z
G19 (Y/Z)	X	Z	Y

3.16.2 Assignment when changing plane

The assignment of tool length components (length, wear, and tool base dimension) to the geometry axes does **not** change when the machining plane is changed if the following setting data is set to not equal to zero:

SD42940 \$SC_TOOL_LENGTH_CONST <> 0

The assignment of the tool length components to the geometry axes is then derived from the ones and tens position of the setting data as shown in the following tables

Table 3-3 Turning / grinding tools (\$TC_DP1 = 400 ... 599)

SD42940	Assignment of tool length components to geometry axes		
	Length L1	Length L2	Length L3
= x17	Y	X	Z
= x18	X	Z	Y
= x19	Z	Y	X
= -x17	X	Y	Z
= -x18	Z	X	Y
= -x19	Y	Z	X

3.16 Assignment of tool length components to geometry axes

Table 3-4 Milling / special tools (\$TC_DP1 <> 400 ... 599)

SD42940	Assignment of tool length components to geometry axes		
	Length L1	Length L2	Length L3
= x17	Z	Y	X
= x18	Y	X	Z
= x19	X	Z	Y
= -x17	Z	X	Y
= -x18	Y	Z	X
= -x19	X	Y	Z

Each value not equal to 0 and not equal to one of the six listed values is evaluated as value "17" (for milling/special tools) or "18" (for turning/grinding tools)..

Note**Assignment of tool orientation components**

The assignment of the tool orientation components is not influenced by SD42940 \$SC_TOOL_LENGTH_CONST.

If necessary, the following setting data must be adapted:

- SD42954 \$SC_TOOL_ORI_CONST_M
- SD42956 \$SC_TOOL_ORI_CONST_T

See Chapter "Tool orientation for plane change (Page 218)".

3.16.3 Assignment independent of tool type

With the following setting data, the assignment of the tool length components (length, wear and tool base dimension) to the geometry axes that is independent of the actual tool type (\$TC_DP1) can be defined:

SD42950 \$SC_TOOL_LENGTH_TYPE = <value>

<value>	Assignment of the tool length components
0 (default value)	In accordance with the specification in the system variable \$TC_DP1[...]. A distinction is made between the following tool types: • \$TC_DP1[...] == 400 ... 599 ⇒ turning/grinding tools • \$TC_DP1[...] <> 400 ... 599 ⇒ milling/special tools
1	Always as for milling/special tools.
2	Always as for turning/grinding tools.

3.17 Paraxial tool orientation

<value>	Assignment of the tool length components	
3	Activates setting data SD42942 \$SC_TOOL_LENGTH_CONST_T. With this setting, it is possible to define the assignment of the tool length components that is effective on a machining plane change (Page 216) separately for milling/special tools and turning/grinding tools:	
	SD42940 \$SC_TOOL_LENGTH_CONST	Assignment for milling/special tools
	SD42942 \$SC_TOOL_LENGTH_CONST_T	Assignment for turning/grinding tools
	Example: SD42940 \$SC_TOOL_LENGTH_CONST = 17 SD42942 \$SC_TOOL_LENGTH_CONST_T = 18 Tool length compensations for milling/special tools take effect in plane G17; length compensations for turning/grinding tools take effect in plane G18.	
> 3	As for SD42950 = 0.	

3.17 Paraxial tool orientation

3.17.1 Basic tool orientation

The basic tool orientation results from the active machining plane:

Working plane	Tool orientation (basic setting)
G17 (X/Y)	Parallel to Z
G18 (Z/X)	Parallel to Y
G19 (Y/Z)	Parallel to X

3.17.2 Tool orientation for plane change

A setting in the setting data determines how the tool orientation changes when the machining plane is changed.

- SD42954 \$SC_TOOL_ORI_CONST_M (for milling/special tools)
- SD42956 \$SC_TOOL_ORI_CONST_T (for turning/grinding tools)

Standard behavior

With the default setting (SD42954 and SD42956 = 0), the tool orientation changes with a plane change as follows:

Plane change	Change in tool orientation
G17 → G18	1. Rotation by -90° about the Z coordinate 2. Rotation by -90° about the X coordinate
G18 → G19	
G19 → G17	

Plane change	Change in tool orientation
G18 → G17	1. Rotation by 90° about the X coordinate 2. Rotation by 90° about the Z coordinate
G19 → G18	
G17 → G19	

Constant tool orientation for a plane change

If SD42954 or SD42956 are not equal to zero, a right-handed orthogonal tool coordinate system is defined, which remains unchanged if the machining plane is changed (G17-G19). The orientation coordinate system is determined by the orientation vector and an orientation normal vector perpendicular to it. The basic orientation is defined by the ones and tens position of the setting data:

Table 3-5 Milling / special tools (\$TC_DP1 <> 400 ... 599)

SD42954	Coordinate system of the tool orientation	
	Orientation vector	Orientation normal vector
= xx17	(0, 0, 1)	(0, 1, 0)
= xx18	(0, 1, 0)	(1, 0, 0)
= xx19	(1, 0, 0)	(0, 0, 1)

Table 3-6 Turning / grinding tools (\$TC_DP1 = 400 ... 599)

SD42956	Coordinate system of the tool orientation	
	Orientation vector	Orientation normal vector
= xx17	(0, 0, 1)	(0, 1, 0)
= xx18	(0, 1, 0)	(1, 0, 0)
= xx19	(1, 0, 0)	(0, 0, 1)

Each value not equal to 0 and not equal to one of the three listed values is evaluated as value "17" (for milling/special tools) or "18" (for turning/grinding tools).

Further information

A detailed description of SD42954 and SD42956 is given in the List Manual System Variables.

Note

In the case of tools whose orientation is defined by cutting edge data (\$TC_DPV...), SD42954 or SD42956 is usually ignored (see Chapter "Parameterizable basic tool orientation (Page 220)").

3.18 Parameterizable basic tool orientation

3.18.1 Function

Note**Tool T and cutting edge D**

In the following, the syntax [...] represents [<t>, <d>] in relation to the system variables of the basic tool orientation. In this regard, <t> designates the number of the tool T=<t> and <d> the number of the tool cutting edge D=<d>.

The "Parameterizable basic tool orientation" function allows each tool cutting edge to be assigned an individual initial orientation with the following system variables:

- Selection of a predefined orientation vector
 - \$TC_DPV[...]: Orientation vector (values 1 ... 6)
- Definition of an orientation vector.
 - \$TC_DPV3[...]: L1 component of the orientation vector
 - \$TC_DPV4[...]: L2 component of the orientation vector
 - \$TC_DPV5[...]: L3 component of the orientation vector
- Definition of a normal vector for the orientation vector
 - \$TC_DPVN3[...]: L1 component of the normal vector
 - \$TC_DPVN4[...]: L2 component of the normal vector
 - \$TC_DPVN5[...]: L3 component of the normal vector

If the system variable \$TC_DPV[...] is equal to zero, the three system variables \$TC_DPV3 - 5[...] define the direction vector of the basic tool orientation. Additionally, the three system variables \$TC_DPVN3 - 5[...] can define the orientation of the tool in the plane perpendicular to the orientation vector. The magnitudes of the orientation vectors are immaterial.

3.18.2 Commissioning

3.18.2.1 Activation

The "Parameterizable basic tool orientation" is activated via the following machine data:

MD18114 \$MN_MM_ENABLE_TOOL_ORIENTATION = <value>

<value>	Meaning
0	The function "Parameterizable basic tool orientation" is not active.
1	Activation of system variables: Function selection: \$TC_DPV[...] With the system variable \$TC_DPV[...] = 1, 2, ... 6, one of six predefined basic tool orientations can be assigned for each tool cutting edge D=<d> of a tool T=<t>.
2	Activation of system variables: - Function selection: \$TC_DPV[...] - Orientation vector: \$TC_DPV3[...], \$TC_DPV4[...], and \$TC_DPV5[...] With the system variables \$TC_DPV[...] = 0 and \$TC_DPV3[...], \$TC_DPV4[...], and \$TC_DPV5[...], any orientation vector can be assigned for each tool cutting edge D=<d> of a tool T=<t>.
3	Activation of system variables: - Function selection: \$TC_DPV[...] - Orientation vector: \$TC_DPV3[...], \$TC_DPV4[...], and \$TC_DPV5[...] - Normal vector for the orientation vector: \$TC_DPVN3[...], \$TC_DPVN4[...], and \$TC_DPVN5[...] With the setting \$TC_DPV[...] = 0, any orientation vector can be assigned for each tool cutting edge D=<d> of a tool T=<t> with the system variables \$TC_DPV3[...], \$TC_DPV4[...], and \$TC_DPV5[...]. Additionally, a normal vector can be defined for the orientation vector with the system variables \$TC_DPVN3[...], \$TC_DPVN4[...], and \$TC_DPVN5[...].

3.18.2.2 Parameterization

Assignment of the system variables \$TC_DPVx[...]

A differentiation is made between the following tool types in accordance with the specification in system variable \$TC_DP1[...]:

- \$TC_DP1[...] == 400 ... 599 ⇒ turning/grinding tools
- \$TC_DP1[...] <> 400 ... 599 ⇒ milling/special tools

The basic tool orientation programmed in \$TC_DPVx[...] is assigned to the coordinate axes depending on the tool type.

The tool type may be changed over by the setting in setting data SD42950 \$SC_TOOL_LENGTH_TYPE (as described in Chapter "Assignment independent of tool type (Page 217)" for tool length components).

If no other orientation is defined in setting data SD42954 \$SC_TOOL_ORI_CONST_M and SD42956 \$SC_TOOL_ORI_CONST_T (Page 218), the standard case (G17 for milling/special tools, G18 for turning/grinding tools) applies for assignment of the system variables \$TC_DPVx[...]:

Tool type	TC_DPV3[...]	TC_DPV4[...]	TC_DPV5[...]
Turning/grinding tools	X	Z	Y
Milling/special tools	Z	Y	X

Note

SD42954 / SD42956

The assignment of the system variables \$TC_DPVx[...] can only be changed with SD42954 \$SC_TOOL_ORI_CONST_M and SD42956 \$SC_TOOL_ORI_CONST_T if the 1000s position is equal to "1"

Irrespective of the value of setting data SD42950 \$SC_TOOL_LENGTH_TYPE, SD42954 is only effective for a tool for which parameter \$TC_DP1[...] defines a milling tool. Analogously, SD42956 is only effective for a tool for which parameter \$TC_DP1[...] defines a turning/grinding tool.

3.18.3 Programming

The system variable \$TC_DPV[...] or the system variables \$TC_DPV3 - 5[...] and \$TC_DPVN3 - 5[...] can be used to assign each tool cutting edge its own basic orientation.

Setting options

Basically, the following setting options are available:

- **\$TC_DPV[...] == 0 AND \$TC_DPV3 - 5[...] == 0**

The vector for the basic tool orientation results from the active machining plane:

- G17: Z coordinate
- G18: Y coordinate
- G19: X coordinate

See also "Paraxial tool orientation (Page 218)".

- **\$TC_DPV[...] == 0 AND \$TC_DPV3 - 5[...] <> 0**

The vector for the basic tool orientation is prescribed by \$TC_DPV3 - 5[...] and \$TC_DPVN3 - 5[...]:

- \$TC_DPV3[...] = <L1 component of the orientation vector>
- \$TC_DPV4[...] = <L2 component of the orientation vector>
- \$TC_DPV5[...] = <L3 component of the orientation vector>
- \$TC_DPVN3[...] = <L1 component of the normal vector>
- \$TC_DPVN4[...] = <L2 component of the normal vector>
- \$TC_DPVN5[...] = <L3 component of the normal vector>

Example:

Basic tool orientation points in the direction of the bisectors in the L1-L3 plane, i.e. for a milling tool and active plane G17, in the direction of the bisectors in the ZX plane.

```
$TC_DPV[1,1] = 0
$TC_DPV3[1,1] = 1.0
$TC_DPV4[1,1] = 0.0
$TC_DPV5[1,1] = 1.0
```

Note

The system variables \$TC_DPVN3-5[...] are not relevant for rotationally symmetrical tools such as milling tools.

- **\$TC_DPV[...] == 1, 2, ... 6**

The vector for the basic tool orientation is prescribed by \$TC_DPV[...].

The following tables show which basic tool orientations are predefined and can be selected via \$TC_DPV[...].

Selection of a predefined orientation vector

\$TC_DPV[...] = <value>

Table 3-7 Turning / grinding tools (\$TC_DP1 = 400 ... 599)

<value>	\$TC_DPV3[...] / \$TC_DPVN3[...]	\$TC_DPV5[...] / \$TC_DPVN5[...]	\$TC_DPV4[...] / \$TC_DPVN4[...]
1	0 / 0	1 / 0	0 / 1
2	1 / 0	0 / 1	0 / 0

3.18 Parameterizable basic tool orientation

<value>	\$TC_DPV3[...] / \$TC_DPVN3[...]	\$TC_DPV5[...] / \$TC_DPVN5[...]	\$TC_DPV4[...] / \$TC_DPVN4[...]
3	0 / 1	0 / 0	1 / 0
4	0 / 0	-1 / 0	0 / -1
5	-1 / 0	0 / -1	0 / 0
6	0 / -1	0 / 0	-1 / 0

Table 3-8 Milling / special tools (\$TC_DP1 <> 400 ... 599)

<value>	\$TC_DPV5[...] / \$TC_DPVN5[...]	\$TC_DPV4[...] / \$TC_DPVN4[...]	\$TC_DPV3[...] / \$TC_DPVN3[...]
1	0 / 1	0 / 0	1 / 0
2	0 / 0	1 / 0	0 / 1
3	1 / 0	0 / 1	0 / 0
4	0 / -1	0 / 0	-1 / 0
5	0 / 0	-1 / 0	0 / -1
6	-1 / 0	0 / -1	0 / 0

Examples

Turning/grinding tools:

\$TC_DPV[...] = 2 Corresponds to

- \$TC_DPV3[...] = 1
- \$TC_DPV4[...] = 0
- \$TC_DPV5[...] = 0
- \$TC_DPVN3[...] = 0
- \$TC_DPVN4[...] = 0
- \$TC_DPVN5[...] = 1

Milling/special tools:

\$TC_DPV[...] = 3 Corresponds to

- \$TC_DPV3[...] = 0
- \$TC_DPV4[...] = 0
- \$TC_DPV5[...] = 1
- \$TC_DPVN3[...] = 0
- \$TC_DPVN4[...] = 1
- \$TC_DPVN5[...] = 0

3.18.4 Examples

A milling tool is defined with length L1=10 whose basic tool orientation is in the bisector of the X-Z plane.

Program code	Comment
\$SC_TOOL_LENGTH_TYPE=2	; assignment to the coordinate axes as for turning/ grinding tools.
\$SC_TOOL_ORI_CONST_M=1019	; Working plane: G19
N10 \$TC_DP1[1,1]=120	; Tool type: Milling tool
N20 \$TC_DP3[1,1]=10	; Length compensation vector: L1=10
N30 \$TC_DPV[1,1]=0	; Basic tool orientation via \$TC_DPV3-5
N40 \$TC_DPV3[1,1]=1	; Orientation vector in the Z direction
N50 \$TC_DPV4[1,1]=0	; Orientation vector in the Y direction
N60 \$TC_DPV5[1,1]=1	; Orientation vector in the X direction
N70 TRAFoon(...)	; Activating a transformation.
N80 G17 F1000 X0 Y0 Z0 T1 D1	; Activate tool and approach zero position. ; Due to tool length L1=10 => ; Machine positions: X=0, Y=0, Z=10 ; Plane selection G17 in block N80 is only rele- vant for length compensations.
N90 MOVt=10	; Incr. Traversing movement in the tool direction.
N100 M30	; Resulting axis positions: X=7.071 Y=0 Z=17.071

3.19 Special handling of tool offsets

3.19.1 Relevant setting data

SD42900- 42960

Setting data SD42900 - SD42940 can be used to make the following settings with reference to tool offset:

- Sign of the tool length
- Sign of the wear
- Behavior of the wear components when mirroring geometry axes
- Behavior of the wear components when changing the machining plane via setting data

3.19 Special handling of tool offsets

- Allocation of the tool length components independent of actual tool type
- Transformation of wear components into a suitable coordinate system for controlling the effective tool length

Note

In the following description, the wear includes the total values of the following components:

- Wear values: \$TC_DP12 bis \$TC_DP20
 - Sum offset, consisting of:
 - Wear values: \$SCPX3 to \$SCPX11
 - Setup values: \$ECPX3 to \$ECPX11
-

You will find detailed information about sum and tool offsets in:

Further information

Function Manual Tool Management

Programming Manual. Fundamentals; Tool Offsets

Required setting data

- SD42900 \$SC_MIRROR_TOOL_LENGTH (mirroring of tool length components and components of the tool base dimension) (Page 226)
- SD42910 \$SC_MIRROR_TOOL_WEAR (mirroring of wear values of tool length components) (Page 226)
- SD42920 \$SC_WEAR_SIGN_CUTPOS (sign evaluation of the wear components) (Page 227)
- SD42930 \$SC_WEAR_SIGN (inverts the sign of the wear dimensions) (Page 227)
- SD42935 \$SC_WEAR_TRANSFORM (transformation of wear values) (Page 245)
- SD42940 \$SC_TOOL_LENGTH_CONST (allocation of the tool length components to the geometry axes) (Page 216)
- SD42950 \$SC_TOOL_LENGTH_TYPE (allocation of the tool length components independent of tool type) (Page 217)
- SD42960 \$SC_TOOL_TEMP_COMP (tool length offsets) (Page 229)

3.19.2 Mirroring tool lengths

Activation

Tool length mirroring is activated via the setting data:

SD42900 \$SC_MIRROR_TOOL_LENGTH <> 0 (TRUE) (Sign change tool length when mirroring)

Function

The following components are mirrored by inverting the sign:

- Tool lengths: \$TC_DP3, \$TC_DP4, \$TC_DP5
- Tool base dimensions: \$TC_DP21, \$TC_DP22, \$TC_DP23

Mirroring is performed for all tool base dimensions whose associated axes are mirrored. Wear values are **not** mirrored.

Mirror wear values

The following setting data should be set in order to mirror the wear values:

SD42910 \$SC_MIRROR_TOOL_WEAR <> 0 (Sign change tool wear when mirroring)

Inverting the sign mirrors the wear values of the tool length components whose associated axes are mirrored.

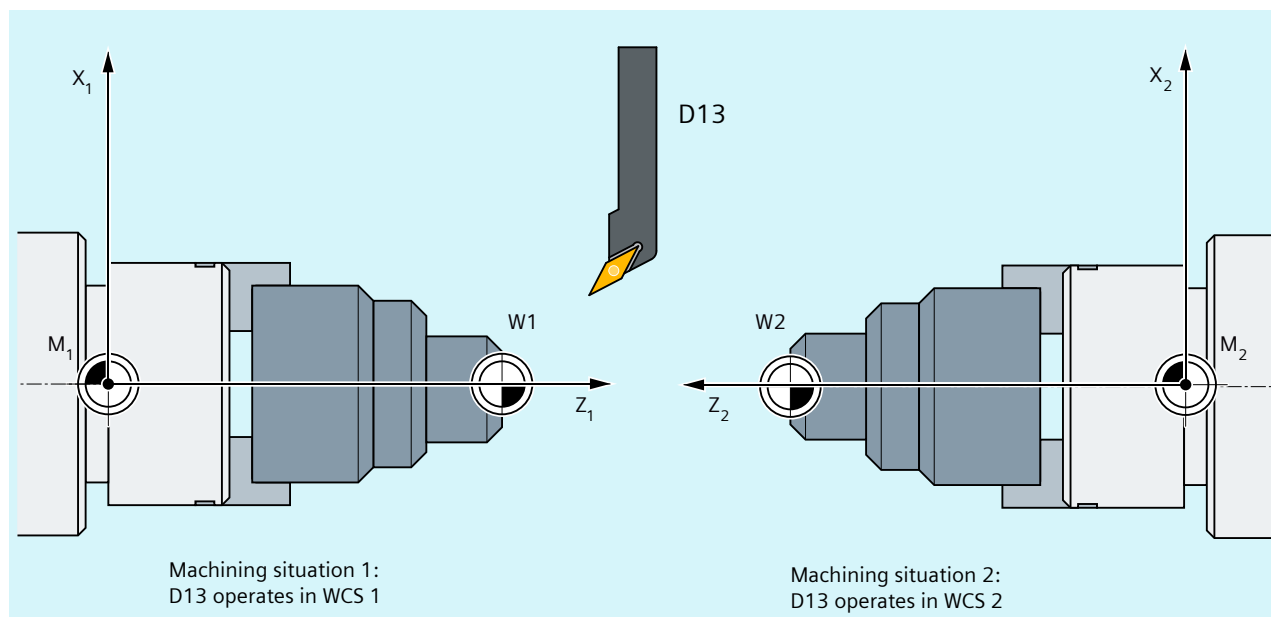


Figure 3-47 Application example: Double-spindle turning machine

3.19.3 Mirroring wear lengths

Activation

Wear length mirroring is activated by:

SD42920 \$SC_WEAR_SIGN_CUTPOS <> 0 (TRUE) (Sign of wear for tools with cutting edge systems)

Function

Length of cutting edge	Length 1	Length 2
1	---	---
2	---	Inverted
3	Inverted	Inverted
4	Inverted	---
5	---	---
6	---	---
7	---	Inverted
8	Inverted	---
9	---	---

In the case of tool types without a relevant cutting edge position, the wear length is not mirrored.

Note

The mirroring (sign inversion) in one or more components can cancel itself through a simultaneous activation of the functions:

Tool length-mirroring (SD42900 $\leftrightarrow$ 0)

And:

Tool length-mirroring (SD42920 $\leftrightarrow$ 0)

SD42930 \$SC_WEAR_SIGN

Setting data not equal to zero:

Inverts the sign of all wear dimensions. This affects both the tool length and other variables such as tool radius, rounding radius, etc.

Entering a positive wear dimension makes the tool "shorter" and "thinner".

Activation of modified setting data

When the setting data described above are modified, the tool components are not reevaluated until the next time a tool edge is selected. If a tool is already active and the data of this tool are to be reevaluated, the tool must be selected again.

Example:

```

N10 $SC_WEAR_SIGN = 0           ; No sign inversion of the wear values
N20 $TC_DP1[1,1] = 120          ; End mill
N30 $TC_DP6[1,1] = 100          ; Tool radius 100 mm
N40 $TC_DP15[1,1] = 1           ; Wear dimension of tool radius 1 mm, resulting
                                ; tool radius 101 mm
N100 T1 D1 G41 X150 Y20

```

```

....
N150 G40 X300N10
....
N200 $SC_WEAR_SIGN = 1          ; Sign inversion for all wear values; the new ra-
                                ; dius of 99 mm is activated on a new selection
                                ; (D1). Without D1, the radius would continue to
                                ; be 101 mm.
N300 D1 G41 X350 Y-20
N310 ....

```

The same applies in the event that the resulting tool length is modified due to a change in the mirroring status of an axis. The tool must be selected again after the mirror command, in order to activate the modified tool-length components.

3.19.4 Tool lengths in the WCS, allowing for the orientation

Change tool or working plane

The values displayed for the tool correspond to the expansion in the WCS. If a toolholder with an inclined clamping position is to be used, you should make sure that the transformation used supports the toolholder. If this is not the case, incorrect tool dimensions will be displayed. When changing the working plane from G17 to G18 or G19, you should ensure that the transformation can also be used for these working planes. If the transformation is only available for G17 machining, the dimensions continue to be displayed for a tool in the Z direction after the plane change.

When transformation is deactivated, the basic tool is displayed in the x, y or z direction, according to the working plane. Allowance is made for a programmed toolholder. These tool dimensions are not altered when traversing without a transformation.

3.19.5 Tool length offsets in tool direction

Temperature compensation in real time

On 5-axis machines with a moving tool, temperature fluctuations can occur in the machining heads. These can result directly in expansion fluctuations which are transmitted to the tool spindle in the form of linear expansion. A typical case on 5-axis heads, for example, is thermal expansion in the direction of the longitudinal spindle axis.

It is possible to compensate this thermal expansion even when the tool is orientated by assigning the temperature compensation values to the tool rather than to the machine axes. In this way, linear expansion fluctuations can be compensated even when the tool orientation changes.

Using the orientation transformation whose direction is determined by the current tool orientation, it is possible to overlay motions in real time and rotate them simultaneously. At

the same time, the compensation values are adjusted continuously in the tool coordinate system.

The temperature compensation only becomes effective if the axis to be compensated is really referenced.

Activation

The temperature compensation in the tool direction is activated by setting the following machine data to a value **not equal to zero**.

MD20390 \$MC_TOOL_TEMP_COMP_ON (activation of temperature compensation for tool length)

In addition, bit 2 must be set for each affected channel axis in the machine data:

MD32750 \$MA_TEMP_COMP_TYPE [<axis index>] (temperature compensation type)

This can be more than three axes in cases where more than three channel axes in succession can be temporarily assigned to geometry axes as a result of geometry axis replacement of transformation switchover. If this bit is not set for a particular channel axis, the compensation value cannot be applied in the axis. This does not have any effect on other axes. In this case, an alarm is not output.

Applicability

The temperature compensation in the tool direction is only effective with generic 5-axis transformations with:

- Transformation type 24
Two axes rotate the tool
- Transformation type 56
One axis rotates the tool, the other axis rotates the workpiece without temperature compensation

In generic 5-axis transformation with:

- Transformation type 40
The tool orientation is constant with a rotary workpiece, which means that the motion of the rotary axes on the machine does not affect the temperature compensation direction.

Temperature compensation in the tool direction also works in conjunction with orientation transformations (not generic 5-axis transformations) with:

- Transformation type 64 to 69
Rotating linear axis

Note

Temperature compensation can be activated with all other types of transformation. It is not affected by a change in tool orientation. The axes move as if no orientation transformation with temperature compensation were active.

Limit values

The compensation values are restricted to the maximum values by the machine data:

MD20392 \$MC_TOOL_TEMP_COMP_LIMIT[0...2] (maximum temperature compensation for tool length)

The limit value default setting is 1 mm. If a temperature compensation value higher than this limit is specified, it will be limited without an alarm.

SD42960

The three temperature compensation values together form a compensation vector and are contained in setting data:

SD42960 \$SC_TOOL_TEMP_COMP[0...2] (temperature compensation with reference to tools)

The setting data is user-defined, e.g. using synchronized actions or from the PLC. The compensation values can, therefore, also be used for other compensation purposes.

In the initial state or when orientation transformation is deactivated, all three compensation values apply in the direction of the three geometry axes (in the typical order X, Y, Z).

The assignment of components to geometry axes is independent of the tool type (turning, milling or grinding tools) and the selected machining plane G17 to G19. Changes to the setting data values take effect immediately.

Toolholder with orientation capability

If a toolholder with orientation capability is active, the temperature compensation vector is rotated simultaneously to any change in orientation. This applies independently of any active orientation transformation.

If a toolholder with orientation capability is active in conjunction with a generic 5-axis transformation or a transformation with rotating linear axis, the temperature compensation vector is subjected to both rotations.

Note

While transformations with rotating linear axes take changes in the tool vector (length) into account, they **ignore** its change in orientation, which can be effected by a toolholder with orientation capability.

Temperature compensation values immediately follow any applied change in orientation. This applies in particular when an orientation transformation is activated or deactivated.

The same is true when the assignment between geometry axes and channel axes is changed. The temperature compensation value for an axis is reduced to zero (interpolatory), for example, when it ceases to be a geometry axis after a transformation change. Conversely, any temperature compensation value for an axis which changes over to geometry axis status is applied immediately.

Examples

Temperature compensation in tool direction

Example of a 5-axis machine with rotating tool on which the tool can be rotated around the C and B axes.

In its initial state, the tool is parallel to the Z axis. If the B axis is rotated through 90 degrees, the tool points in the X direction.

Therefore, a temperature compensation value in the following setting data is also effective in the direction of the machine X axis if transformation is active:

SD42960 \$SC_TOOL_TEMP_COMP[2] (temperature compensation with reference to tools)

If the transformation is deactivated with the tool in this direction, the tool orientation is, by definition, parallel again to the Z axis and thus different to its actual orientation. The temperature offset in the X axis direction is therefore reduced to zero and reapplied simultaneously in the Z direction.

Example of a 5-axis machine with rotating tool (transformation type 24). The relevant machine data is listed below:

- The first rotary axis rotates around Z, C-axis
- The second rotary axis rotates around Y, B-axis

The essential machine data is shown in the table below:

Machine data	Value	Remark
MD20390 \$MC_TOOL_TEMP_COMP_ON	= TRUE	Temperature compensation active
MD32750 \$MA_TEMP_COMP_TYPE[AX1]	= 4	Compensation in tool direction
MD32750 \$MA_TEMP_COMP_TYPE[AX2]	= 4	Compensation in tool direction
MD32750 \$MA_TEMP_COMP_TYPE[AX3]	= 4	Compensation in tool direction
		Assignment of transformation type 24
MD24100 \$MC_TRAFO_TYPE_1	= 24	Transformer type 24 in first channel
MD24110 \$MC_TRAFO_AXES_IN_1[0]	= 1	First axis of the transformation
MD24110 \$MC_TRAFO_AXES_IN_1[1]	= 2	Second axis of the transformation
MD24110 \$MC_TRAFO_AXES_IN_1[2]	= 3	Third axis of the transformation
MD24110 \$MC_TRAFO_AXES_IN_1[3]	= 5	Fifth axis of the transformation
MD24110 \$MC_TRAFO_AXES_IN_1[4]	= 4	Fourth axis of the transformation
MD24120 \$MC_TRAFO_GEOAX_ASSIGN_TAB_1[0]	= 1	Geometry axis for channel axis 1
MD24120 \$MC_TRAFO_GEOAX_ASSIGN_TAB_1[1]	= 2	Geometry axis for channel axis 2
MD24120 \$MC_TRAFO_GEOAX_ASSIGN_TAB_1[2]	= 3	Geometry axis for channel axis 3
MD24570 \$MC_TRAFO5_AXIS1_1[0]	= 0.0	
MD24570 \$MC_TRAFO5_AXIS1_1[1]	= 0.0	Direction
MD24570 \$MC_TRAFO5_AXIS1_1[2]	= 1.0	First rotary axis is parallel to Z
MD24572 \$MC_TRAFO5_AXIS1_2[0]	= 0.0	Direction

Machine data	Value	Remark
MD24572 \$MC_TRAFO5_AXIS1_2[1]	= 1.0	Second rotary axis is parallel to Y
MD24572 \$MC_TRAFO5_AXIS1_2[2]	= 0.0	
MD25574 \$MC_TRAFO5_BASE_ORIENT_1[0]	= 0.0	
MD25574 \$MC_TRAFO5_BASE_ORIENT_1[1]	= 0.0	Basic tool orientation
MD25574 \$MC_TRAFO5_BASE_ORIENT_1[2]	= 1.0	In Z direction

Temperature compensation values in the NC program

The compensation values assigned to axes X and Z are not zero and are applied for temperature compensation with respect to tool length. The machine axis positions reached in each case are specified as comments in the program lines.

Program code	Comment
\$SC_TOOL_TEMP_COMP[0] = -0.3	; Compensation value in X
\$SC_TOOL_TEMP_COMP[1] = 0.0	
\$SC_TOOL_TEMP_COMP[2] = -1.0	; Compensation value in Z
	; Position setpoints of the machine axes
N10 G74 X0 Y0 Z0 A0 B0	; X Y Z
N20 X20 Y20 Z20 F10000	; 20.30 20.00 21.00
N30 TRAORI()	; 20.30 20.00 21.00
N40 X10 Y10 Z10 B90	; 11.00 10.00 9.70
N50 TRAFOOF	; 10.30 10.00 11.00
N60 X0 Y0 Z0 B0 C0	; 0.30 0.00 1.00
N70 M30	

With the exception of block N40, temperature compensation always acts in the original directions, as the tool is pointing in the basic orientation direction. This applies particularly in block N50. The tool is actually still pointing in the direction of the X axis because the B axis is still at 90 degrees. However, because the transformation is already deactivated, the applied orientation is parallel to the Z axis again.

Machine data	Value	Remark
MD20390 \$MC_TOOL_TEMP_COMP_ON	= TRUE	Temperature compensation active
MD32750 \$MA_TEMP_COMP_TYPE[AX1]	= 4	Compensation in tool direction
MD32750 \$MA_TEMP_COMP_TYPE[AX2]	= 4	Compensation in tool direction
MD32750 \$MA_TEMP_COMP_TYPE[AX3]	= 4	Compensation in tool direction

Further information

- For more details on "Temperature compensation" see:
Function Manual Monitoring and Compensations; Compensations
- For information on "Generic 5-axis transformations" see:
Function Manual Transformations; Multi-axis transformations

3.19.6 Special characteristics of orientable tool carriers

Setting data SD42900 - SD42950

Setting data SD42900 - SD42950 have no effect on the components of an active tool carrier with orientation capability. The calculation with a tool carrier with orientation capability always allows for a tool with its total resulting length (tool length + wear + tool base dimension). The calculation of the resulting total length allows for all modifications caused by the setting data.

Note

When tool carriers with orientation capability are used, it is common to define all tools for a non-mirrored basic system, even those, which are only used for mirrored machining. When machining with mirrored axes, the tool carrier is then rotated such that the actual position of the tool is described correctly. All tool-length components then automatically act in the correct direction, dispensing with the need for control of individual component evaluation via setting data, depending on the mirroring status of individual axes.

The use of tool carriers with orientation capability is also practical if the physical characteristics of the machine type prevents tools, which are permanently installed with different orientations, from being rotated. Tool dimensioning can then be performed uniformly in a basic orientation, where the dimensions relevant for machining are calculated according to the rotations of a virtual tool carrier.

3.20 Sum and setup offsets

3.20.1 General information

Sum offsets

Sum offsets can be treated as **programmable process compensations** during machining and are composed of all the error sizes (including the wear), which cause the workpiece to deviate from the specified dimensions.

Sum offsets are a generalized type of wear. They are part of the cutting edge data. The parameters of the sum offset refer to the geometrical data of a cutting edge.

The compensation data of a sum offset is addressed by a **DL number (DL: Location-dependent; offsets regarding the location of use)**.

In contrast, the wear values of a D number describe the physical wear of the cutting edge, i.e. in special situations, the sum offset can match the wear of the cutting edge.

Summed offsets are intended for general use, i.e. with active or inactive tool management.

Machine data is used to classify the sum offsets into:

- Sum offset fine
- Sum offset coarse (setup offset)

Setup offset

The setup offset is the compensation to be entered by the setup engineer before machining. These values are stored separately in the NC. The operator subsequently only has access to the "sum offset fine" via HMI.

The "sum offset fine" and "sum offset coarse" are added internally in the NC. This value is referred to below as the sum offset.

Note

The function is enabled via the machine data setting:

MD18080 \$MN_MM_TOOL_MANAGEMENT_MASK, Bit 8=1 (step-by-step memory reservation for tool management).

If kinematic transformations (e.g. 5-axis transformations) are active, the tool length is calculated first after allowing for the various wear components. The total tool length is then used in the transformation. Unlike the case of a toolholder with orientation capability, the wear values are thus always included in the transformation irrespective of the G command of group 56.

3.20.2 Description of function

Sum offsets

Several sum offsets (DL numbers) can be defined per D number. In this way, for example, **workpiece location-dependent** offset values can be determined and assigned to a cutting edge. Sum offsets have the same effect as wear, i.e. they are added to the offset values of the D number. The data is permanently assigned to a D number.

Settings

You can define the following settings via machine data:

- Activate sum offset
- Define maximum number of DL data sets to be created in NC
- Define maximum number of DL numbers to be assigned to a D number
- Define whether the sum offsets (fine/coarse) are to be saved during data backup
- Define the sum offset to be activated, if:
 - A new cutting edge compensation is activated
 - An operator panel RESET is performed
 - An operator panel START is performed
 - The end of the program has been reached

The name is oriented to the logic of the corresponding machine data for tools and cutting edges.

3.20 Sum and setup offsets

The "setup offset" and "sum offset fine" can be read and written via system variables and corresponding OPI services.

Note

When tool management is active, a machine data item can be used to define whether the sum offset of a tool activated during a programmed tool change remains unchanged or is set to zero.

Summary of compensation parameters \$TC_DPx

The following general system variables were previously defined for describing a cutting edge:

\$TC_DP1	Tool type
\$TC_DP2	Cutting edge position

Parameters for geometry and wear

Tool geometry compensations are assigned to system variables \$TC_DP3 to \$TC_DP11. System variables \$TC_DP12 to \$TC_DP20 allow you to name a wear for each of these parameters:

Geometry	Wear	Compensations
Length compensations		
\$TC_DP3	\$TC_DP12	Length 1
\$TC_DP4	\$TC_DP13	Length 2
\$TC_DP5	\$TC_DP14	Length 3
Radius compensation		
\$TC_DP6	\$TC_DP15	Radius 1
\$TC_DP7	\$TC_DP16	Radius 2
Other compensations		
\$TC_DP8	\$TC_DP17	Length 4
\$TC_DP9	\$TC_DP18	Length 5
\$TC_DP10	\$TC_DP19	Angle 1
\$TC_DP11	\$TC_DP20	Angle 2

Tool base dimension or adapter dimension

\$TC_DP21	Adapter length 1
\$TC_DP22	Adapter length 2
\$TC_DP23	Adapter length 3

Technology

System variable	Clearance angle
\$TC_DP24	The clearance angle is stored here for ManualTurn; tool type 5xx. Same significance as in standard cycles for turning tools. The tip angle of the drill is stored here for ShopMill; tool type 2xx. Used in standard cycles for turning tools; tool type 5xx. This is the angle at the secondary cutting edge for these tools.
\$TC_DP25	The value for the cutting rate is stored here for ManualTurn. A bit-coded value for various states of tool types 1xx and 2xx is stored here for ShopMill.

Parameters of the sum and setup offsets (\$TC_SCPxy, \$TC_ECPxy)

The numbering of the parameters is oriented to the numbering of system variables \$TC_DP3 to \$TC_DP11.

The effect of the parameters is similar to the wear (additive to the tool geometry). Up to six sum/setup parameters can be defined per cutting edge parameter.

Tool geometry parameters (to which the compensation is added)	Sum / setup parameters	Tool wear parameter
Length compensations		
\$TC_DP3	Length 1 \$TC_SCP13, \$TC_SCP23, \$TC_SCP33, \$TC_SCP43, \$TC_SCP53, \$TC_SCP63 \$TC_ECP13, \$TC_ECP23, \$TC_ECP33, \$TC_ECP43, \$TC_ECP53, \$TC_ECP63 The numbers in bold, 1, 2, ... 6, designate the parameters of a maximum of six (location-dependent or similar) compensations that can be programmed with DL =1 to 6 for the parameter specified in column one.	\$TC_DP12
\$TC_DP4	Length 2 \$TC_SCP14, \$TC_SCP24, \$TC_SCP34, \$TC_SCP44, \$TC_SCP54, \$TC_SCP64 \$TC_ECP14, \$TC_ECP24, \$TC_ECP34, \$TC_ECP44, \$TC_ECP54, \$TC_ECP64	\$TC_DP13
\$TC_DP5	Length 3 etc. ...	\$TC_DP14
Radius compensation		
\$TC_DP6	Radius	\$TC_DP15
\$TC_DP7	Corner radius	\$TC_DP16
Other compensations		
\$TC_DP8	Length 4	\$TC_DP17

3.20 Sum and setup offsets

Tool geometry parameters (to which the compensation is added)	Sum / setup parameters	Tool wear parameter
\$TC_DP9	Length 5	\$TC_DP18
\$TC_DP10	Angle 1 ...etc.	\$TC_DP19
\$TC_DP11	Angle 2 \$TC_SCP21, \$TC_SCP31, \$TC_SCP41, \$TC_SCP51, \$TC_SCP61, \$TC_SCP71 \$TC_ECP21, \$TC_ECP31, \$TC_ECP41, \$TC_ECP51, \$TC_ECP61, \$TC_ECP71 The numbers in bold, 2, 3, ... 7, designate the parameters of a maximum of six (location-dependent or similar) compensations that can be programmed with DL = 1 to 6 for the parameter specified in column one.	\$TC_DP20

Supplementary conditions

The maximum number of DL data sets of a cutting edge and the total number of sum offsets in the NC are defined via machine data. The default value is zero, i.e. no sum offsets can be programmed.

Activate the "monitoring function" to monitor a tool for wear or for "sum offset".

The additional sum/setup data sets use additional buffered memory. 8 bytes are required per parameter.

A sum offset data set requires: 8 bytes * 9 parameters = 72 bytes

A setup data set requires an equal amount of memory. A certain number of bytes is also required for internal management data.

3.20.3 Activation

Function

The function must be activated via the machine data:

MD18108 \$MN_MM_NUM_SUMCORR (sum offsets in TO area).

System variables \$TC_ECPx and \$TC_SCPx and setup and sum offsets ("fine") defined via the OPI interface can be activated in the part program.

This is done by programming the language command `DL="number"`.

When a new D number is activated, either a new DL number is programmed, or the DL number defined via the following machine data becomes active:

MD20272 \$MC_SUMCORR_DEFAULT (basic setting of the sum offset without a program)

DL programming

The sum offset is always programmed relative to the active D number with the command:

DL = "n"

The sum offset "n" is added to the wear of the active D number.

Note

If you use "setup offset" **and** "sum offset fine", both compensations are combined and added to the tool wear.

The sum offset is deselected with the command:

DL = 0

Note

DL0 is not permitted. If compensation is deselected (D0 and T0), the sum offset also becomes ineffective.

Programming a sum offset that does not exist triggers an alarm, similar to programming a D compensation that does not exist.

Thus, only the defined wear remains part of the compensation (defined in system variables \$TC_DP12 to \$TC_DP20).

Programming a sum offset when a D compensation is active (also applies to deselection) has the same effect on the path as programming a D command. An active radius compensation will, therefore, lose its reference to adjacent blocks.

Configuration

MD18112 \$MN_MM_KIND_OF_SUMCORR, bit 4=0: (Properties of sum offset in the TO area) default setting:

Only **one** set of sum offsets exists per DL number.

We refer in general to the sum offset.

This describes the data represented by \$TC_SCPx.

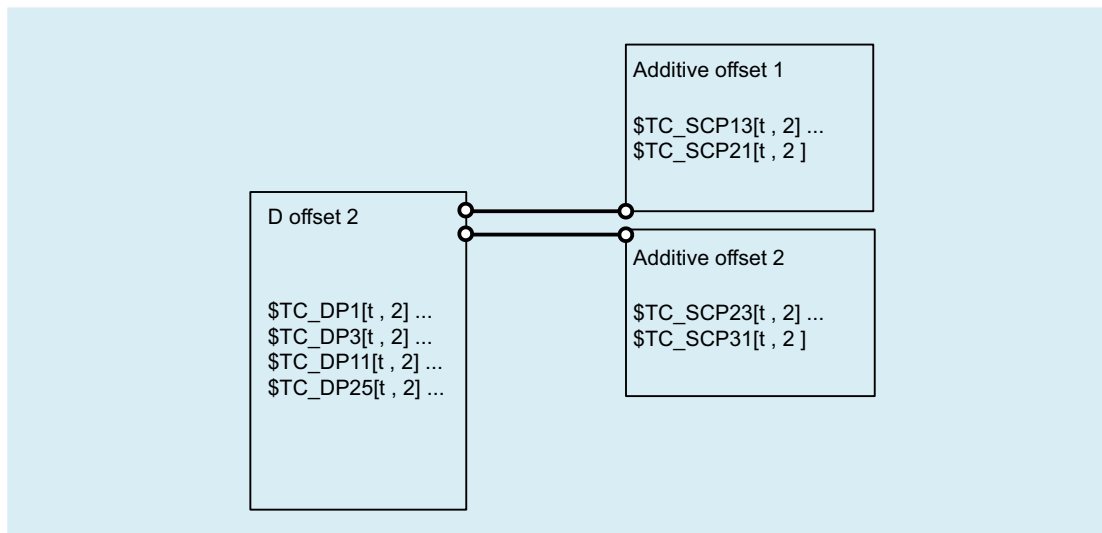


Figure 3-48 MD18112 \$MN_MM_KIND_OF_SUMCORR, bit 4 = 0

The tool with T = t is active. With the data in the figure, the following is programmed:

D2	;	Cutting edge compensations, i.e. \$TC_DP3 to \$TC_DP11 + wear (\$TC_DP12 to \$TC_DP20) + adapter dimension
...		
DL=1	;	Sum offset 1 is added to the previous D2 compensations, i.e. \$TC_SCP13 to \$TC_SCP21.
...		
DL=2	;	Sum offset 2 is added to the D2 compensation instead of sum offset 1, i.e. \$TC_SCP23 to \$TC_SCP31.
...		
DL=0	;	Deselection of sum offset; only the data of D2 remains active.

MD18112 \$MN_MM_KIND_OF_SUMCORR, bit 4=1: Setup offsets are available

The sum offset is now composed of the "sum offset fine" (represented by \$TC_SCPx) and the setup offset (represented by \$TC_ECPx). Two data sets therefore exist for one DL number. The sum offset is calculated by adding the corresponding components (\$TC_ECPx + \$TC_SCPx).

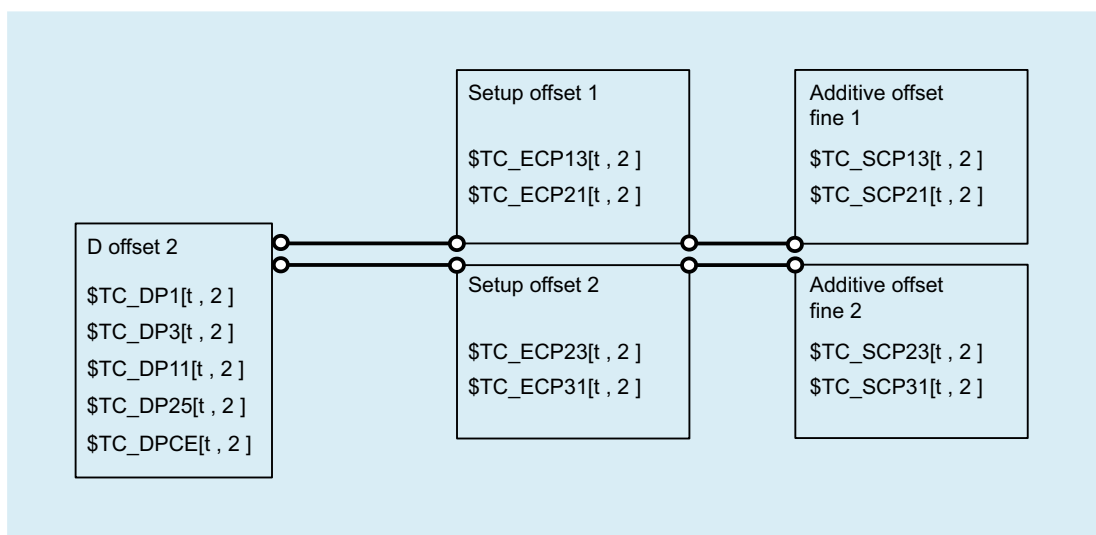


Figure 3-49 MD18112 \$MN_MM_KIND_OF_SUMCORR, bit 4 = 1 "setup offsets" + "sum offsets fine"

The tool with T = t is active. With the data in the figure, the following is programmed:

D2	;	Cutting edge compensations, i.e. \$TC_DP3 to \$TC_DP11 + wear (\$TC_DP12 to \$TC_DP20) + adapter dimension
...		
DL=1	;	Sum offset 1 is added to the previous D2 compensations, i.e. \$TC_ECP13 + \$TC_SCP13 to \$TC_ECP21 + \$TC_SCP21.
...		
DL=2	;	Sum offset 2 is added to the D2 compensation instead of sum offset 1; i.e. \$TC_ECP23 + \$TC_SCP23,...\$TC_ECP31 + \$TC_SCP31
...		
DL=0	;	Deselection of sum offset. Only the data of D2 remains active.

Reading/writing in the part program

The individual sets of sum offset parameters are differentiated according to the number ranges of system variable \$TC_SCP.

The significance of the individual variables is similar to geometry variables \$TC_DP3 to \$TC_DP11. Only length 1, length 2 and length 3 are enabled for the basic functionality (variables \$TC_SCP13 to \$TC_SCP15 for the first sum offset of the cutting edge).

R5 = \$TC_SCP13[t, d]	;	Sets the value of the R parameter to the value of the first component of sum offset 1 for cutting edge (d) on tool (t).
R6 = \$TC_SCP21[t, d]	;	Sets the value of the R parameter to the value of the last component of sum offset 1 for cutting edge (d) on tool (t).
R50 = \$TC_SCP23[t, d]	;	Sets the value of the R parameter to the value of the first component of sum offset 2 for cutting edge (d) on tool (t).
\$TC_SCP43[t, d] = 1.234	;	Sets the value of the first component of sum offset 4 for cutting edge (d) on tool (t) to the value 1.234.

3.20 Sum and setup offsets

The above statements also apply to the setup offsets (if the NCK is configured with this option), i.e.

$R5 = \$TC_ECP13[t, d]$	;	Sets the value of the R parameter to the value of the first component of setup offset 1 for cutting edge (d) on tool (t).
$R6 = \$TC_ECP21[t, d]$	;	Sets the value of the R parameter to the value of the last component of setup offset 1 for cutting edge (d) on tool (t).
etc.		

When working with setup offsets, "sum offsets fine" are written with the $\$TC_SCP_x$ system variables.

Creating a new sum offset

If the compensation data set (x) does not yet exist, it is created on the first write operation to one of its parameters (y).

$\$TC_SCP_{xy}[t, d] = r.r$	;	The value "r.r" is assigned to parameter y of sum offset x. The other parameters of x have the value zero.
------------------------------	---	--

When working with setup offsets, "sum offsets fine" are written with the $\$TC_SCP_x$ system variables.

Note

When working with setup offsets, the data set for the setup offset is created when a data set is created for "sum offset fine", if a data set did not already exist for [t, d].

Creating a new setup offset

If the compensation data set (x) does not yet exist, it is created on the first write operation to one of its parameters (y).

$\$TC_ECP_{xy}[t, d] = r.r$	;	The value "r.r" is assigned to parameter y of setup offset x. The other parameters of x have the value zero.
------------------------------	---	--

Note

When working with setup offsets, the data set for the "sum offset fine" is created when a data set is created for setup offsets, if a data set did not already exist for [t, d].

DELDL - Delete sum offset

Sum offsets are generally only relevant when machining with a cutting edge at a certain time at a certain location of the workpiece. You can use the NC language command `DELDL` to delete sum offsets from cutting edges (in order to release memory).

status = DELDL(t, d)	;	Deletes all sum offsets of cutting edge d on tool t.
	;	t, d are optional parameters.

If d is not specified, all sum offsets of all cutting edges on tool t are deleted.

If d and t are not specified, all sum offsets for the cutting edges on all tools of the TO unit are deleted (for the channel, in which the command is programmed).

When working with setup offsets, the `DELDL` command deletes both the setup offset and the "sum offsets fine" of the specified cutting edge(s).

Note

The memory used for the data sets is released after deletion.

The deleted sum offsets can subsequently no longer be activated or programmed.

Sum offsets and setup offsets on active tools cannot be deleted (similar to the deletion of D compensations or tool data).

The "status" return value indicates the result of the deletion command:

0:	Deletion was successful
-1:	Deletion was not (one cutting edge) or not completely (several cutting edges) successful

Data backup

The data is saved during a general tool data backup (as a component of the D number data sets).

It is advisable to save the sum offsets, in order to allow the current status to be restored in the event of an acute problem. Machine data settings can be made to exclude sum offsets from a data backup (settings can be made separately for "setup offsets" and "sum offsets fine").

Note

Sum offsets behave in the same way as D compensations with reference to block search and REPOS. The behavior on Reset and Power On can be defined by machine data.

If the setting of the following machine data indicates that the last active tool compensation number (D) is to be activated after Power On, the last active DL number is then no longer active:

MD20110 \$MC_RESET_MODE_MASK (definition of control default settings after reset / part program end)

3.20.4 Examples

Example 1

That no compensation and no sum offset will come into effect must be defined during tool change via the machine data:

- MD20270 \$MC_CUTTING_EDGE_DEFAULT=0 (Basic setting of tool cutting edge without programming)
- MD20272 \$MC_SUMCORR_DEFAULT=0 (default setting sum offset without program).

```

T5 M06          ; Tool number 5 is loaded - no compensation active.
D1 DL=3         ; Compensation D1 + sum offset 3 of D1 are activated.
X10
DL=2           ; Compensation D1 + sum offset 2 are activated.
X20
DL=0           ; Sum offset deselection, only compensation D1 is now active.
D2            ; Compensation D2 is activated - the sum offset is not included
              ; in the compensation.
X1
DL=1           ; Compensation D2 + sum offset 1 are activated.
X2
D0            ; Compensation deselection
X3
DL=2           ; No effect - DL2 of D0 is zero (same as programming T0 D2).
```

Example 2

During tool change it has to be defined that offset D2 and sum offset DL=1 are activated via the machine data:

MD20270 \$MC_CUTTING_EDGE_DEFAULT=2 (Basic setting of tool cutting edge without programming)

MD20272 \$MC_SUMCORR_DEFAULT=1 (default setting sum offset without program)

```

T5 M06          ; Tool number 5 is loaded - D2 + DL=1 are active (= values of
              ; machine data)
D1 DL=3         ; Compensation D1 + sum offset 3 of D1 are activated.
X10
DL=2           ; Compensation D1 + sum offset 2 are activated.
X20
DL=0           ; Sum offset deselection, only compensation D1 is now active.
D2            ; Compensation D2 is activated - sum offset DL=1 is activated.
X1
DL=2           ; Compensation D2 + sum offset 2 are activated.
D1            ; Compensation D1 + sum offset 1 are activated.
```

3.20.5 Extensions for tool length determination

3.20.5.1 Calculation of compensation values on a location-specific and workpiece-specific basis

Composition of the effective tool length

For a tool offset without active kinematic transformation, the effective tool length consists of up to 8 vectors:

- Tool length (geometry) (\$TC_DP3 - \$TC_DP5)
- Wear (\$TC_DP12 - \$TC_DP14)
- Tool base dimension (see note) (\$TC_DP21 - \$TC_DP23)
- Adapter dimension (see note) (\$TC_ADPT1 - \$TC_ADPT3)
- Total offsets fine (\$TC_SCPx3 - \$TC_SCPx5)
- Sum offsets coarse or setup offsets (\$TC_ECPx3 - \$TC_ECPx5)
- Offset vector I_1 of toolholder with orientation capability (\$TC_CARR1 - \$TC_CARR3)
- Offset vector I_2 of toolholder with orientation capability (\$TC_CARR4 - \$TC_CARR6)
- Offset vector I_3 of toolholder with orientation capability (\$TC_CARR15 - \$TC_CARR17)

Note

The tool base dimension and adapter dimension can only be applied as alternatives.

Type of action of the individual vectors

The type of action of the individual vectors or groups of vectors depends on the following further quantities:

Influencing quantity	Operating principle
G commands	Active machining plane
Tool type	Milling tool or turning/grinding tools
Machine data	Tool management active/not active, toolholder with orientation capability available/not available
Setting data	Behavior of tool length components when mirroring or when changing the plane
Tool carriers with orientation capability	Set values of toolholder with orientation capability
Adapter transformations	Transformed tool offset values

Distribution over the geometry-axis components

How the three vector components of partial totals of the vectors involved are distributed over the three geometry-axis components is determined by the following quantities:

Influencing quantity	Dependencies
Active processing level: G17 X/Y direction G18 Z/X direction G19 Y/Z direction	Infeed plane: Z Y X
Tool type: Milling tools, drilling tools, grinding tools, turning tools	See Section "Tool parameter 1: Tool type (Page 40)", Table "Minimum number of required tool parameters"
SD42900 \$SC_MIRROR_TOOL_LENGTH SD42910 \$SC_MIRROR_TOOL_WEAR SD42920 \$SC_WEAR_SIGN_CUTPOS SD42930 \$SC_WEAR_SIGN SD42940 \$SC_TOOL_LENGTH_CONST SD42950 \$SC_TOOL_LENGTH_TYPE	See Section "Special handling of tool offsets (Page 225)".
Adapter transformations	Further information: Function Manual Tool Management

The resulting tool orientation always remains parallel to one of the three axis directions X, Y or Z and exclusively depends on the active machining plane G17–G19, since it has not yet been possible to assign the tool an orientation.

Stepless variation of the tool orientation

The toolholder with orientation capability also enables the tool orientation to be varied steplessly, in addition to providing further offsets or linear expansion fluctuations with the aid of offset vectors I_1 - I_3 .

For further explanations, see Section "Tool carriers with orientation capability (Page 119)".

Minor operator compensations

Minor compensations, however, must also be modified during the normal production mode.

The reasons for this are, for example:

- Tool wear
- Clamping errors
- Temperature sensitivity of the machine:

These compensations are defined as follows:

Definition	Wear components
Wear	\$TC_DP12 - \$TC_DP14,
Total offsets fine	\$TC_SCPx3 - \$TC_SCPx5,
Sum offsets coarse or setup offsets	\$TC_ECPx3 - \$TC_ECPx5

In particular, compensations, which affect the tool length calculation, should be entered in the coordinates used for measurement.

These workpiece-specific compensations can be achieved more simply using the G group 56 with the three values `TOWSTD`, `TOWMCS` and `TOWWCS` and the setting data:

SD42935 \$SC_WEAR_TRANSFORM (transformation of tool components)

SD42935

Which of the wear components:

- Wear (\$TC_DP12 - \$TC_DP14)
- Setup offsets or sum offsets coarse (\$TC_ECPx3 - \$TC_ECPx5)
- Sum offsets fine (\$TC_SCPx3 - \$TC_SCPx5)

are to be transformed in the transformations:

- Adapter transformation
- Toolholder with orientation capability

are to be or not to be transformed, can be defined via the setting data:

SD42935 \$SC_WEAR_TRANSFORM (transformation of wear values)

With the setting data in its initial state, all wear values are transformed.

The setting data is considered in the following functions:

- Wear values in the machine coordinate system
Part program instruction: TOWMCS
- Wear values in the workpiece coordinate system
Part program instruction: TOWWCS

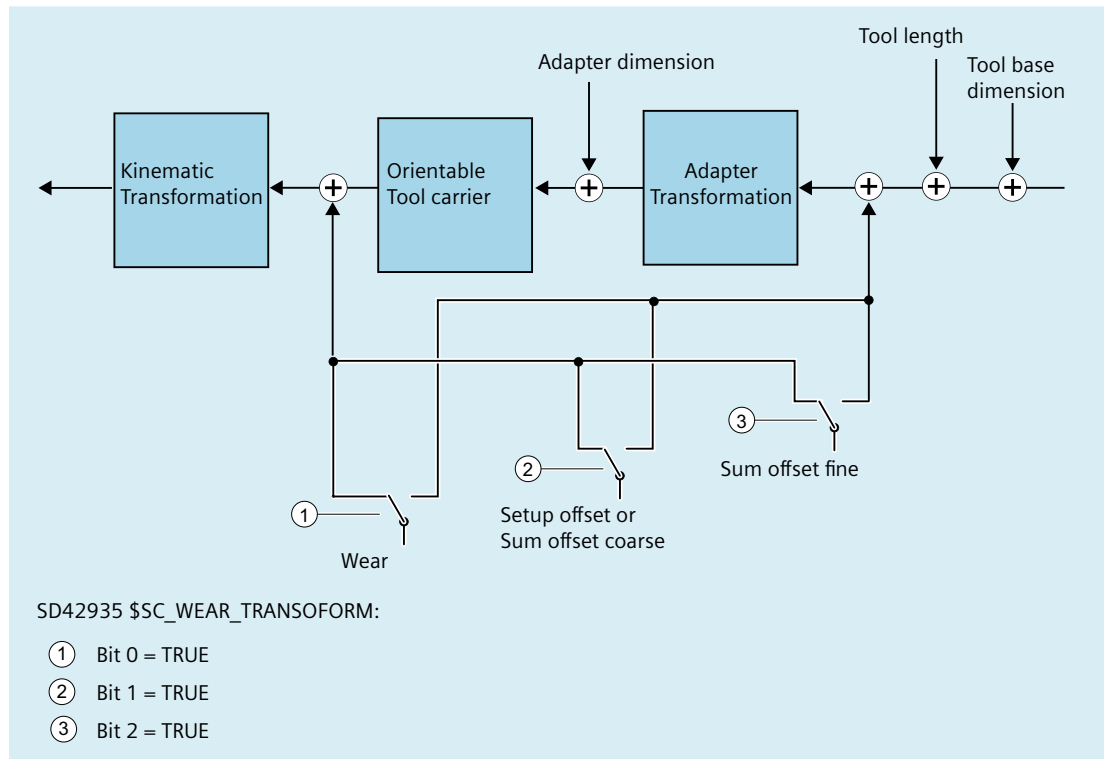


Figure 3-50 Transformation of wear data dependent on SD42935

Programming

G-code group 56 can be used to define the following values:

Syntax	Corrections
TOWSTD	Basic setting value for offsets in tool length
TOWMCS	Wear values in the machine coordinate system (MCS)
TOWWCS	Wear values in the workpiece coordinate system (WCS)
TOWBCS	Wear values in the basic coordinate system (BCS)
TOWTCS	Wear values in the TCS (Tool Coordinate System) at the toolholder (tool carrier reference point T)
TOWKCS	Wear values in tool coordinate system for kinematic transformation (KCS) of tool head

Coordinate systems for offsets in tool length

G commands `TOWMCS`, `TOWWCS`, `TOWBCS`, `TOWTCS` and `TOWKCS` can be used, e.g. to measure the wear tool length component in five different coordinate systems.

1. Machine coordinate system	MCS
1. Basic coordinate system	BCS
1. Workpiece coordinate system	WCS
1. Tool coordinate system of kinematic transformation	KCS
1. Tool coordinate system	TCS

The calculated tool length or a tool length component can be represented and read out using the predefined `GETTCOR` (Page 257) function in one of these coordinate systems.

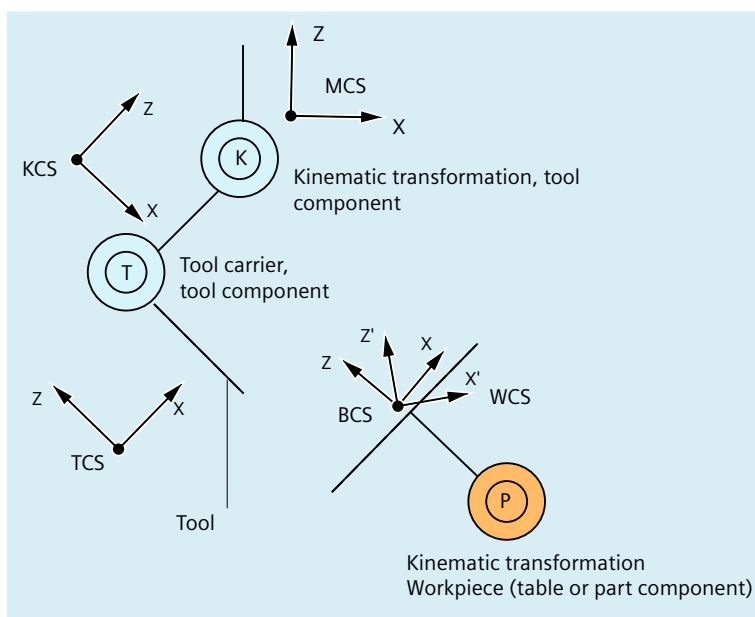


Figure 3-51 Coordinate system for the evaluation of tool lengths

3.20.5.2 Functionality of the individual wear values

TOWSTD

Basic setting (default behavior):

- The wear values are added to the other tool length components. The resulting total tool length is then used in further calculations.

In the case of an active tool carrier with orientation capability:

- The wear values are subjected to the appropriate rotation.

TOWMCS

Wear data in the MCS (machine coordinate system):

In the case of an active rotation by means of a tool carrier with orientation capability:

- The tool carrier only rotates the vector of the resultant tool length. Wear is ignored. Then the tool length vector rotated in this way and the wear are added. The wear is not subjected to the rotation.

If **no** tool carrier with orientation capability is active or this does not result in a rotation, TOWMCS and TOWSTD are identical.

Linear transformation

The tool length can be uniquely defined in the MCS only if the MCS is generated by linear transformation from the BCS.

This would also be the case under one of the following conditions:

- No kinematic transformation active.
- Orientation transformations (3-axis, 4-axis, and 5-axis transformations) are active.

TOWWCS

Wear values in WCS (workpiece coordinate system):

- If a tool carrier with orientation capability is active, the tool vector is calculated as for TOWMCS, without taking the wear into account.
- The wear data is interpreted in the workpiece coordinate system.

The wear vector in the workpiece coordinate system is converted to the machine coordinate system and added to the tool vector.

TOWBCS

Wear values in BCS (basic coordinate system):

- If a tool carrier with orientation capability is active, the tool vector is calculated as for TOWMCS, without taking the wear into account.
- The wear data is interpreted in the workpiece coordinate system.

The wear vector in the basic coordinate system is converted to the workpiece coordinate system and added to the tool vector.

Non-linear transformation

If a non-linear transformation is active, e.g. with TRANSMIT, and the MCS is specified as the desired coordinate system, the BCS is automatically used instead of the MCS.

Tool carrier with orientation capability

A table component of the tool carrier with orientation capability, if available, is not applied directly to the coordinate systems, unlike a table (or part) component of the kinematic transformation. A rotation described by such a component is represented in a basic frame or system frame and is thus included in the transition from WCS to BCS.

Kinematic transformation

The table (or part) component of the kinematic transformation is described by the transition from BCS to MCS.

TOWTCS

Wear values in TCS (tool coordinate system):

- If a tool carrier with orientation capability is active, the tool vector is calculated as for TOWMCS, without taking the wear into account.
- The wear data is interpreted in the tool coordinate system.

The wear vector in the TCS (Tool Coordinate System) is converted to the machine coordinate system by way of the tool coordinate system of the kinematic transformation (KCS) and added to the tool vector.

TOWKCS

The wear value specifications for the kinematic transformation are interpreted in the associated TCS (Tool Coordinate System).

The wear vector is converted to the machine coordinate system by way of the tool coordinate system of the kinematic transformation and added to the tool vector.

G command change when a tool is active

Changing the G command in the group TOWSTD, TOWMCS, TOWWCS, TOWBCS, TOWTCS, and TOWKCS does not affect an already active tool, and does not become effective until the next tool is selected.

A new G command of this group will also come into effect if it is programmed in the same block, in which a tool is selected.

Evaluation of individual wear components

Evaluation of individual wear components (assignment to geometry axes, sign evaluation) is influenced by:

- The active plane
- The adapter transformation
- The five setting data shown in the table below

Setting data	Wear components		
SD42910 \$SC_MIRROR_TOOL_WEAR	TOWSTD	TOWMCS	TOWWCS
SD42920 \$SC_WEAR_SIGN_CUTPOS	X	X	—
SD42930 \$SC_WEAR_SIGN	X	—	—
SD42940 \$SC_TOOL_LENGTH_CONST	X	X	X
SD42950 \$SC_TOOL_LENGTH_TYPE	X	X	X

Note

Wear components which are subjected to an active rotation by an adapter transformation or a tool carrier with orientation capability are referred to as non-transformed wear components.

Special features

If TOWMCS or TOWWCS is active, the following setting data does not affect the non-transformed wear components:

SD42920 \$SC_WEAR_SIGN_CUTPOS (Sign of wear for tools with cutting edge systems)

The following setting data also does not affect the non-transformed wear components in case of TOWWCS:

SD42910 \$SC_MIRROR_TOOL_WEAR (Sign change tool wear when mirroring)

In this case, a possibly active mirroring is already contained in the frame, which is referred to for evaluating the wear components.

On a plane change, the assignment between the non-transformed wear components and the geometry axes is retained, i.e. these are not interchanged as with other length components. The assignment of components depends on the active plane for tool selection.

Example

Let's assume a milling tool is used where only the wear value \$TC_DP12 assigned to length L1 is not equal to zero.

If G17 is active, this length is effective in the direction of the z axis.

This measure always acts in the Z-direction also upon a plane change after the tool selection, when TOWMCS or TOWWCS are active and the bit 1 is set in the setting data:

SD42935 \$SC_WEAR_TRANSFORM (transformations for tool components)

If, for example, G18 is active on tool selection, the component is always effective in the Y direction instead.

3.21 Working with tool environments**Overview of functions**

- Save tool environment (TOOENV) (Page 253)
- Delete tool environment (DELTOOENV) (Page 255)
- Read T, D and DL number (GETTENV) (Page 256)
- Read tool lengths and/or tool length components (GETTCOR) (Page 257)
- Change tool components (SETTCOR) (Page 263)

System variables overview

- Read information about the saved tool environments (\$P_TOOLENVN, (\$P_TOOLENV) (Page 257)

3.21.1 Save tool environment (TOOLENV)

The TOOLENV function is used to save any current states needed for the evaluation of tool data stored in the memory.

The individual data are as follows:

- The active G command of group:
 - 6 (G17, G18, G19)
 - 56 (TOWSTD, TOWMCS, TOWWCS, TOWBCS, TOWTCS, TOWKCS)
- The active transverse axis
- Machine data:
 - MD18112 \$MN_MM_KIND_OF_SUMCORR (properties of the summed offsets in the TO area)
 - MD20360 \$MC_TOOL_PARAMETER_DEF_MASK (definition of tool parameters).
- Setting data:
 - SD42900 \$SC_MIRROR_TOOL_LENGTH (sign change tool length when mirroring)
 - SD42910 \$SC_MIRROR_TOOL_WEAR (sign change tool wear when mirroring)
 - SD42920 \$SC_WEAR_SIGN_CUTPOS (sign of the tool wear with cutting edge systems)
 - SD42930 \$SC_WEAR_SIGN (sign of wear)
 - SD42935 \$SC_WEAR_TRANSFORM (transformations for tool components)
 - SD42940 \$SC_LENGTH_CONST (change of the tool length components for a plane change)
 - SD42942 \$SC_TOOL_LENGTH_CONST_T (change of tool length components for turning tools at change of plane)
 - SD42950 \$SC_TOOL_LENGTH_TYPE (allocation of the tool length components independent of tool type)
 - SD42954 \$SC_TOOL_ORI_CONST_M (change of tool orientation components for milling tools at change of plane)
 - SD42956 \$SC_TOOL_ORI_CONST_T (change of tool orientation components for turning tools at change of plane)
- The orientation component of the current complete frame (rotation and mirroring, no work offsets or scaling)
- The orientation component and the resulting length of the active toolholder with orientation capability
- The orientation component and the resulting length of an active transformation

In addition to the data describing the environment of the tool, the T number, D number and DL number of the active tool are also stored, so that the tool can be accessed later in the same environment as the TOOLENV call, without having to name the tool again.

Syntax

```
<Status> = TOOLENV (<name>)
```

Meaning

TOOLENV (. . .) :		Predefined function to save a tool environment	
		Alone in the block:	Yes
<Status>:		Function return value. Negative values indicate error states.	
		Data type:	INT
		Value:	0 Function OK
		-1	No memory reserved for tool environments: MD18116 \$MN_MM_NUM_TOOL_ENV = 0 This means that the "tool environments" functionality is not available.
		-2	No more free memory locations for tool environments available.
		-3	Null string illegal as name of a tool environment.
		-4	No parameter (<name>) specified.
Parameters			
1	<name>:	Name, under which the current data set should be saved. If a data set of the same name already exists, then it is overwritten. In this case, the status is "0".	
		Data type:	STRING

Further information

Base dimension/adaptor dimension – tool length compensation

When the tool magazine management is active (only available with the "Tool management" option!), the value of the following machine data defines whether the adapter length or the tool base dimension (cutting edge-specific parameters \$TC_DP21, \$TC_DP22 and \$TC_DP23) is incorporated in the calculation of the tool length:

MD18104 \$MN_MM_NUM_TOOL_ADAPTER (tool adapter in TO area).

Since a change to this machine data only takes effect after the control system has powered up, it is not saved in the tool environment.

Resulting length of toolholders with orientation capability and transformations:

Note

Both toolholders with orientation capability and transformations can use system variables or machine data, which act as additional tool length components, and which can be subjected partially or completely to the rotations performed. The resulting additional tool length components must also be saved when TOOLENV is called, because they represent part of the environment, in which the tool is used.

Adapter transformation

The adapter transformation is a property of the tool adapter and thus of the complete tool. It is, therefore, not part of a tool environment, which can be applied to another tool.

By saving the complete data necessary to determine the overall tool length, it is possible to calculate the effective length of the tool at a later point in time, even if the tool is no longer active or if the conditions of the environment (e.g. G codes or setting data) have changed. Similarly, the effective length of a different tool can be calculated assuming that it would be used under the same conditions as the tool, for which the status was saved.

Maximum number of data sets for tool environments

Machine data MD18116 \$MN_MM_NUM_TOOL_ENV is used to define the maximum number of data sets that can be saved to describe the tool environments. The data are in the TOA area. They are kept even when the control system is switched off.

Data cannot be backed up. This means that this data cannot be transferred between the different control systems.

3.21.2 Delete tool environment (DELTOOLENV)

The DELTOOLENV function is used to delete the data sets that are used to describe tool environments. Deletion means that the set of data stored under a particular name can no longer be accessed (an access attempt triggers an alarm).

Note

Data sets can only be deleted using the DELTOOLENV function, by an INITIAL.INI download or by a cold start (NC power up with default machine data). There are no additional automatic deletion operations.

Syntax

```
<Status> = DELTOOLENV (<name>)
<Status> = DELTOOLENV ()
```

Meaning

DELTOOLENV (. . .):		Predefined function to delete a tool environment	
		Alone in the block:	Yes
<Status>:		Function return value. Negative values indicate error states.	
		Data type:	INT
		Value:	0 Function OK
			-1 No memory reserved for tool environments: MD18116 \$MN_MM_NUM_TOOL_ENV = 0 This means that the "tool environments" functionality is not available.
		-2	A tool environment with the specified name does not exist.
Parameters			
1	<name>:	Name of data set to be deleted	
		Data type:	STRING

3.21 Working with tool environments

DELTOOLENV () :	DELTOOLENV () deletes data sets describing tool environments without specifying a name
------------------	---

3.21.3 Read T, D and DL number (GETTENV)

The GETTENV function is used to read the T, D and DL numbers stored in a tool environment.

Syntax

<Status> = GETTENV (<name>, <TDDL>)

Meaning

GETTENV (. . .) :		Predefined function to read T, D and DL numbers in a data set to describe a tool environment	
		Alone in the block:	Yes
<Status>:		Function return value. Negative values indicate error states.	
		Data type:	INT
		Value:	0 Function OK
			-1 No memory reserved for tool environments: MD18116 \$MN_MM_NUM_TOOL_ENV = 0 This means that the "tool environments" functionality is not available.
			-2 A tool environment with the specified name does not exist.
Parameters			
1	<name>:	Name of the data set from which the T, D and DL numbers are to be read	
		Data type:	STRING
2	<TDDL>:	The field of this result parameter contains the T, D and DL numbers of the tool, whose tool environment is saved in the specified data set:	
		• <TDDL> [0]: T number • <TDDL> [1]: D number • <TDDL> [2]: DL number	
		Data type:	INT[3]
GETTENV (, <TDDL>), GETTENV (" ", <TDDL>) :		When calling function GETTENV, it is permissible to omit the first parameter – or to transfer the null string as first parameter. In these two special cases, in <TDDL>, the T, D and DL numbers of the active tool are returned.	

3.21.4 Read information about the saved tool environments (\$P_TOOLENVN, \$P_TOOLENV)

Information regarding the saved tool environments can be read using the following system variables:

\$P_TOOLENVN:	Supplies the number of data sets (which have still not been deleted) – defined using TOOLENV – to describe tool environments			
	Syntax:	<n> = \$P_TOOLENVN		
	Meaning:	<n>:	Number of defined data sets	
			Data type:	INT
			Value range:	0 ... MD18116 \$MN_MM_NUM_TOOL_ENV
This system variable can be accessed even if no tool environments are possible (MD18116 = 0). In this case, the return value is "0".				
\$P_TOOLENV:	Supplies the name of the <i>-th data set to describe a tool environment			
	Syntax:	<Name> = \$P_TOOLENV [<i>]		
	Meaning:	<name>:	Name of the data set with number <i>	
			Data type:	STRING
		<i>:	Number of the data set	
			Data type:	INT
	Value range:	1 ... \$P_TOOLENVN		
	The assignment of numbers to data sets is not fixed, but can be changed as a result of deleting or creating data sets. The data sets are numbered internally. If <i> refers to a data set that has not been defined, then the null string is returned. If index <i> is not valid, i.e. <i> is less than 1 or higher than that the maximum number of data sets for tool environments (MD18116 \$MN_MM_NUM_TOOLENV), then the following alarm is output: Alarm 17020 "inadmissible array index 1"			

3.21.5 Read tool lengths and/or tool length components (GETTCOR)

The GETTCOR function is used to read out tool lengths or tool length components.

The parameters can be used to specify which components are considered and the conditions under which the tool is used.

Syntax

```
<Status> = GETTCOR(<Len>[, <Comp>, <Stat>, <T>, <D>, <DL>])
```

Meaning

GETTCOR(...):	Predefined function to read tool lengths or to read tool length components	
	Alone in the block:	Yes

3.21 Working with tool environments

<Status>:		Function return value. Negative values indicate error states.		
		Data type: INT		
		Value:	0	Function OK
			-1	No memory reserved for tool environments: MD18116 \$MN_MM_NUM_TOOL_ENV = 0 This means that the "tool environments" functionality is not available.
			-2	A tool environment with the name specified under <Stat> does not exist.
			-3	Invalid string in parameter <Comp>. Causes of this error can be invalid characters or characters programmed twice.
			-4	Invalid T number
			-5	Invalid D number
			-6	Invalid DL number
			-7	Attempt to access a non-existent memory module.
			-8	Attempt to access a non-existent option (programmable tool orientation, tool management).
			-9	The <Comp> string contains a colon (identifier for the specification of a coordinate system), but it is not followed by a valid character denoting the coordinate system.
Parameters				
1	<Len>:	Result vector		
		Data type:	REAL[11]	
		The vector components are arranged in the following order: • <Len> [0]: Tool type • <Len> [1]: Cutting edge position • <Len> [2]: Abscissa • <Len> [3]: Ordinate • <Len> [4]: Applicate • <Len> [5]: Tool radius The coordinate system defined in <Comp> and <Stat> is used as the reference coordinate system for the length components. If a coordinate system is not defined in <Comp>, then tool lengths are displayed in the machine coordinate system. The assignment of the abscissa, ordinate and applicate to the geometry axes depends on the active plane used in the tool environment. This means, for G17, the abscissa is parallel to X, with G18 it is parallel to Z, etc. Components <Len>[6] to <Len>[10] contain the additional parameters, which can be used to specify the geometry description of a tool (e.g. \$TC_DP7 to \$TC_DP11 for the geometry and the corresponding components for wear or sum and setup off-sets). These 5 additional elements and the tool radius are only defined for components E, G, S, and W. Their evaluation does not depend on <Stat>. The corresponding values in <Len>[6] to <Len>[10] can thus only be not equal to zero if at least one of the four specified components is involved in the tool length calculation. The remaining components do not influence the result. The dimensions refer to the control's basic system (inch or metric).		

2	<Comp>:	Tool length components (optional)	
		Data type:	STRING
		The character string consists of two substrings, which are separated from one another by a colon.	
		General form: "<SubStr_1> [: <SubStr_2>]"	
		<SubStr_1>:	The first substring designates the tool length components to be taken into account when calculating the tool length. The order of the characters in the substrings, and their notation (upper or lower case), is arbitrary. Any number of blanks or white spaces can be inserted between the characters. Note: It is not permissible that the characters in the substring are programmed twice.
			Charac- ters:
			- Minus symbol (only allowed as first character) The complete tool length is calculated, minus the components specified in the next string.
			C Adapter or tool base dimension (whichever of the two alternative components is active for the tool in use)
			E Setup offsets
			G Geometry
			K Kinematic transformation (is only evaluated for generic 3, 4 and 5-axis transformation)
			S Summed offsets
			T Toolholder with orientation capability
			W Wear
		If the first substring is empty (except for white spaces), the complete tool length is calculated allowing for all components. This applies even if the <Comp> parameter is not specified.	
		<SubStr_2>:	The optional second substring identifies the coordinate system, in which the tool length is to be output. The second substring only comprises one single relevant character.
			Charac- ters:
			A Adjustable coordinate system (ACS)
			B Basic coordinate system (BCS)
			K Tool coordinate system of kinematic transformation (KCS)
			M Machine coordinate system (MCS)
			T Tool coordinate system (TCS)
			W Workpiece coordinate system (WCS)
		If no coordinate system is specified, the evaluation is performed in the MCS (machine coordinate system). If any rotations are to be taken into account, they are specified in the tool environment defined in <Stat>.	
3	<_Stat>:	Name of the data set for describing a tool environment (optional)	
		Data type:	STRING
		If the value of this parameter is the null string (""), or is not specified, then the current status is used. The current tool is used if a tool is not specified.	

3.21 Working with tool environments

4	<T>:	Internal T number of the tool (optional).	
		Data type:	INT
		If this parameter is not specified or if its value is "0", then the tool stored in <Stat> is used.	
		If the value of this parameter is "-1", then the T number of the active tool is used. It is also possible to explicitly specify the number of the active tool. Note: If <Stat> is not specified, the actual status is used as the tool environment. Since <T> = 0 refers to the T number saved in the tool environment, the active tool is used in this environment, i.e. parameters <T> = 0 and <T> = -1 have the same meaning in this special case.	
5	<D>:	Cutting edge of the tool (optional).	
		Data type:	INT
		If this parameter is not specified, or if its value is "0", then the D number used is based on the source of the T number. If the T number from the tool environment is used, then the D number of the tool environment is also read, otherwise the D number of the currently active tool is read.	
6	<DL>:	Number of the offset dependent on the location (optional).	
		Data type:	INT
		If this parameter is not specified, then the DL number used is based on the source of the T number. If the T number from the tool environment is used, then the D number of the tool environment is also read, otherwise the D number of the currently active tool is read.	

Examples

GETTCOR (_LEN)	Calculates the tool length of the currently active tool in the machine coordinate system allowing for all components.
GETTCOR (_LEN, "CGW:W")	Calculates the tool length for the active tool, consisting of the adapter or tool base dimension, geometry and wear. Further components, such as toolholder with orientation capability or kinematic transformation, are not considered. Output in the workpiece coordinate system.
GETTCOR (_LEN, "-K:B")	Calculates the complete tool length of the active tool without allowing for the length components of a possibly active kinematic transformation. Output in the basic coordinate system.
GETTCOR (_LEN, ":M", "Testenv1", , 3)	Calculates the complete tool length in the machine coordinate system for the tool stored in the tool environment named "Testenv1". However, the calculation is performed for cutting edge number D3, regardless of the cutting edge number stored.

Additional information

Adapter transformation/toolholder with orientation capability/kinematic transformation

Any rotations and component exchanges initiated by the adapter transformation, toolholder with orientation capability and kinematic transformation, are part of the tool environment. They are thus always performed, even if the corresponding length component is not supposed to be included. If this is undesirable, tool environments must be defined, in which the corresponding transformations are not active. In many cases (i.e. any time a transformation or toolholder with orientation capability is not used on a machine), the data sets stored for the tool environments automatically fulfill these conditions, with the result that the user does not need to make special provision.

Turning and grinding tools: Calculating the tool length depending on MD20360 \$MC_TOOL_PARAMETER_DEF_MASK

The following machine data defines how the wear and tool length are to be evaluated if a diameter axis is used for turning and grinding tools.

MD20360 \$MC_TOOL_PARAMETER_DEF_MASK (definition of tool parameters).

Bit	Value
0	For turning and grinding tools, the wear parameter of the transverse axis is taken into account as the diameter value:
	= 0 (default) No
	= 1 Yes
1	For turning and grinding tools, the tool length component of the transverse axis is taken into account as the diameter value:
	= 0 (default) No
	= 1 Yes

If the bits involved are set, the associated entry is weighted with a factor of 0.5. This weighting is reflected in the tool length returned by GETTCOR.

Example:

MD20360 \$MC_TOOL_PARAMETER_DEF_MASK = 3

MD20100 \$MC_DIAMETER_AX_DEF (geometry axis with transverse axis function) = "X"

X is diameter axis (standard turning machine configuration)

Program code	Comment
N30 \$TC_DP1[1,1]=500	
N40 \$TC_DP2[1,1]=2	
N50 \$TC_DP3[1,1]=3.0	; geometry L1
N60 \$TC_DP4[1,1]=4.0	
N70 \$TC_DP5[1,1]=5.0	
N80 \$TC_DP12[1,1]=12.0	; wear L1
N90 \$TC_DP13[1,1]=13.0	
N100 \$TC_DP14[1,1]=14.0	
N110 T1 D1 G18	
N120 R1=GETTCOR(_LEN,"GW")	
N130 R3=_LEN[2]	; 17.0 (= 4.0 + 13.0)

3.21 Working with tool environments

Program code	Comment
N140 R4=_LEN[3]	; 7.5 (= 0.5 * 3.0 + 0.5 * 12.0)
N150 R5=_LEN[4]	; 19.0 (= 5.0 + 14.0)
N160 M30	

Length components of the kinematic transformation and toolholder with orientation capability

If a **toolholder with orientation capability** is taken account of during the tool length calculation, the following vectors are included in that calculation:

Type	Vectors
M	I1 and I2
T	I1, I2 and I3
P	The tool length is not influenced by the toolholder with orientation capability.

In generic **5-axis transformation**, the following machine data are included in the tool length calculation for transformer types 24 and 56:

Transformation type	Machine data
24	MD24550/24650 \$MC_TRAFO5_BASE_TOOL_1/2 MD24560/24660 \$MC_TRAFO5_JOINT_OFFSET_1/2 MD24558/24658 \$MC_TRAFO5_PART_OFFSET_1/2
56	MD24550/24650 \$MC_TRAFO5_BASE_TOOL_1/2 MD24560/24660 \$MC_TRAFO5_JOINT_OFFSET_1/2

Transformation type 56 (moving tool and moving workpiece) corresponds to type M for toolholders with orientation capability.

For this 5-axis transformation, in the previous software releases, vector MD24560/24660 \$MC_TRAFO5_JOINT_OFFSET_1/2 (vector of kinematic offset of the 1st/2nd 5-axis transformation in the channel) corresponds to the sum of the two vectors I_1 and I_3 for a type M tool carrier with orientation capability.

Only the sum is relevant for the transformation in both cases. The way, in which the two individual components are composed, is insignificant. However, when calculating the tool length, it is relevant which component is assigned to the tool and which is assigned to the tool table. This is the reason that machine data MD24558/24658 \$MC_TRAFO5_JOINT_OFFSET_PART_1/2 (vector kinematic offset in table) was introduced. It corresponds to vector I_3 . Machine data:MD24560/24660 \$MC_TRAFO5_JOINT_OFFSET_1/2 no longer corresponds to the sum of I_1 and I_3 , but only to vector I_1 . If machine data MD24558/24658 \$MC_TRAFO5_JOINT_OFFSET_PART_1/2 is equal to zero, the behavior is the same as before.

Compatibility

The GETTCOR function is used in conjunction with the TOOLENV and SETTCOR functions to replace parts of the functionality, which were previously implemented externally in the measuring cycles.

Only some of the parameters, which actually determine the effective tool length, were implemented in the measuring cycles. The functions mentioned above can be configured to reproduce the behavior of the measuring cycles in relation to the tool length calculation.

3.21.6 Change tool components (SETTCOR)

The SETTCOR function is used to change tool components taking into account all general conditions that can be involved when evaluating the individual components.

Note

Regarding the terminology: If in the following, in conjunction with the tool length, tool components are involved, then the components considered from a vectorial perspective are meant, which make up the complete tool length (e.g. geometry or wear). Such a component comprises three individual values (L1, L2, L3), which are called coordinate values in the following.

The tool component "geometry" therefore comprises three coordinate values \$TC_DP3 to \$TC_DP5.

Syntax

```
<Status> = SETTCOR(<CorVal>, <Comp>, [<CorComp>, <CorMode>,
<GeoAx>, <Stat>, <T>, <D>, <DL>])
```

Meaning

SETTCOR(...):	Predefined function to change tool components	
	Alone in the block:	Yes

3.21 Working with tool environments

<Status>:	Function return value. Negative values indicate error states.		
	Data type:		INT
	Value:	0	Function OK
		-1	No memory reserved for tool environments: MD18116 \$MN_MM_NUM_TOOL_ENV = 0 This means that the "tool environments" functionality is not available.
		-2	A tool environment with the name specified under <Stat> does not exist.
		-3	Invalid string in parameter <Comp>. Causes of this error can be invalid characters or characters programmed twice.
		-4	Invalid T number.
		-5	Invalid D number.
		-6	Invalid DL number.
		-7	Attempt to access a non-existent memory module.
		-8	Attempt to access a non-existent option (programmable tool orientation, tool management).
		-9	Illegal numerical value for parameter <CorComp>.
		-10	Illegal numerical value for parameter <CorMode>.
		-11	The contents of parameters <Comp> and <CorComp> are contradictory.
		-12	The contents of parameters <Comp> and <CorMode> are contradictory.
		-13	The content of the <GeoAx parameter does not designate a geometry axis.
-14		Write attempt to a non-existent setup offset.	
Parameters			
1	<CorVal>:	Correction vector In the workpiece coordinate system (WCS) defined by <Stat>, the following assignment applies: • <CorVal> [0]: Abscissa • <CorVal> [1]: Ordinate • <CorVal> [2]: Applicate If only one tool component is to be corrected (i.e. no vectorial correction, see parameter <CorMode>), the correction value is always in <CorVal>[0], independent of the axis on which it acts. The contents of the other two components are then not evaluated. If <CorVal> or a component of <CorVal> refers to the transverse axis, then the data is evaluated as radius dimension. This means that a tool is, for example, "longer" by the specified dimension; this correspondingly results in a change to the workpiece diameter that is twice as large. The dimensions refer to the basic system (inch or metric) of the control system.	
		Data type:	REAL[3]

2	<Comp>:	Tool component(s)			
		Data type:		STRING	
		The character string consists of two substrings, which are separated from one another by a colon.			
		General form: "<SubStr_1> [: <SubStr_2>]"			
		<SubStr_1>:	The first substring must always be available, and can either comprise one or two characters. The first or only character for the 1st component (Val ₁) and the second character for the 2nd component (Val ₂), which are processed according to the subsequent parameters <CorComp> and <CorMode>.		
			Charac- ters:	C	Adapter or tool base dimension (whichever of the two alternative components is active for the tool in use)
				E	Setup offsets
				G	Geometry
				S	Sum offsets
				W	Wear
		<Substr_2>:	The second substring is optional. Alternatively, it can comprise (individual) letters "W" or "T".		
			Charac- ters:	W	If the second substring is empty or contains the letter "W", then the offset values are taken into account as if they had been measured in the workpiece coordinate system (WCS).
T	If the second substring contains the letter "T", then the offset values are taken into account as if they had been measured in the tool coordinate system (Tool Coordinate System, TCS).				
The notation of the characters in the string (upper or lower case) is arbitrary. Any number of spaces or tabs (white spaces) can be inserted.					

3.21 Working with tool environments

3	<CorComp>:	Specifies the component(s) of the tool data sets that are to be described (optional).		
		Data type: INT		
		Value:	0	Offset value <CorVal>[0] refers to the geometry axis transferred in parameter <GeoAx> in the workpiece coordinate system – or in the tool coordinate system (also see a description of parameter <Comp>). This means that the offset value must be calculated in the designated tool components so that, taking account all the parameters that can influence the tool length calculation, a change of the total tool length by the specified value in the specified axis direction is obtained. This change should be achieved by correcting the component specified in <Comp> and the symbolic algorithm specified in <CorMode> (see the following parameters). The resulting correction can therefore have an effect on all three axis components.
			1	Like "0", however, vectorial. The content of vector <CorVal> refers to abscissa, ordinate and applicate in the workpiece coordinate system or tool coordinate system (see the description of parameter <Comp>). Subsequent parameter <GeoAx> is not evaluated.
			2	Vectorial offset, i.e. L1, L2 and L3 can change simultaneously. In contrast to the versions from "0 and "1", the offset values contained in <CorVal> refer to the coordinates of Val ₁ components (see following parameter <CorMode>) of the tool. Any possible inclination of an existing tool compared with the workpiece coordinate system has no influence on the offset.
			3 - 5	Correction of tool lengths L1 to L3 (\$TC_DP3 to \$TC_DP5) or the corresponding values for wear, setting up or additive offsets. The offset value is contained in <CorVal>[0]. It is measured in the coordinates of the Val ₁ component (see following parameter <CorMode>) of the tool. Any possible inclination of an existing tool compared with the workpiece coordinate system has no influence on the offset.
			6	Correction of the tool radius (\$TC_DP6) or the corresponding values for wear, setting up or additive offsets. Bits 10 and 11 (evaluation of the diameter and/or diameter wear data, either specified as a radius or diameter) in machine data MD20360 \$MC_TOOL_PARAMETER_DEF_MASK are taken into account.
			7 – 11	Correction of \$TC_DP7 to \$TC_DP11 or the corresponding values for wear, setting up or additive offsets. These parameters are treated just like the tool radius.
		If this parameter is not specified then its value is "0".		

4	<CorMode>:	Specifies the type of write operation to be executed (optional).		
		Data type:	INT	
		Value:	0	$Val_{1new} = \text{<CorVal>}$
			1	$Val_{1new} = Val_{1old} + \text{<CorVal>}$
			2	$Val_{1new} = \text{<CorVal>}$ $Val_{2new} = 0$
	3		$Val_{1new} = Val_{1old} + Val_{2old} + \text{<CorVal>}$ $Val_{2new} = 0$	
	The notation $Val_{1old} + Val_{2old}$ is symbolic. If the two components (due to the status of <_Stat>) are evaluated in different ways, i.e. if a rotation is effective between the two components, then Val_{2old} is transformed prior to addition so that the resulting tool length after deleting Val_{2new} and prior to the addition of <CorVal> remains unchanged. <CorVal> always refers to Val_1. <CorVal> is a value, which dependent on the second part of parameter <Comp>, is measured in the workpiece coordinate system (WCS) or in the tool coordinate system (TCS). It is therefore already transformed with respect to the tool components, in which it should be calculated. Therefore, it cannot be directly calculated together with the saved value, but must be transformed back prior to adding to Val_1 or Val_2. This can mean that the offset acts on an axis different than the one defined by <CorComp> – or that it acts on several axes. For the case <CorComp> = 0, i.e. when <CorVal> does not contain a vector, but only an individual value, then the described operations are executed in the coordinates in which <CorVal> was measured (WCS/TCS). In particular, this also applies to setting Val_{2new} to zero in variants 2 and 3. This result is then transformed back into the coordinates of the tool. This can mean that none of the coordinate values to be set to zero (L1, L2, L3) become zero, or coordinate values, that were previously zero, are now not equal to zero. However, if the corresponding operations are successively executed for all three geometry axes, then it is guaranteed that all three coordinate values of the components to be deleted are zero. If the tool is not rotated with respect to the workpiece coordinate system or is rotated so that all tool components remain parallel to the coordinate axes (axis exchange operations), then this also ensures that only one tool coordinate changes. The successive execution of the same operation (<CorMode>) with <CorComp> = 0 for all three coordinate axes in any sequence is identical with the single execution of the same operation with <CorComp>=1. For parameter values "0" and "1", parameter <Comp> must contain one character, and for parameter values "2" and "3", two characters. Example: <Comp> contains string "ES", <CorMode> the value "2" ⇒ Setup offset_{new} = <CorVal>, summed offset_{new} = 0 If parameter <CorMode> is not specified, then its value is "0".			
	<GeoAx>:	Specifies the index of the geometry axis in which the offset value <CorVal>[0] was read (optional)		
		Data type:	INT	
		Value range:	0 ... 2	
Indices 0 to 2 refer to abscissa, ordinate and applicate in the active plane (G17/G18/G19) of the current tool environment.				
The content of this parameter is only evaluated if parameter <CorComp> has a value of "0".				

3.21 Working with tool environments

6	<Stat>:	Name of the data set for describing a tool environment (optional)	
		Data type:	STRING
		If the value of this parameter is the null string (""), or is not specified, then the current status is used. The current tool is used if a tool is not specified.	
7	<T>:	Internal T number of the tool (optional).	
		Data type:	INT
		If this parameter is not specified or if its value is "0", then the tool stored in <Stat> is used. If the value of this parameter is "-1", then the T number of the active tool is used. It is also possible to explicitly specify the number of the active tool. Note: If <Stat> is not specified, the actual status is used as the tool environment. Since <T> = 0 refers to the T number saved in the tool environment, the active tool is used in this environment, i.e. parameters <T> = 0 and <T> = -1 have the same meaning in this special case.	
8	<D>:	Cutting edge of the tool (optional).	
		Data type:	INT
		If this parameter is not specified, or if its value is "0", then the D number used is based on the source of the T number. If the T number from the tool environment is used, the D number of the tool environment is also read, otherwise the D number of the currently active tool is read.	
9	<TL>:	Number of the offset dependent on the location (optional).	
		Data type:	INT
		If this parameter is not specified, then the DL number used is based on the source of the T number. If the T number from the tool environment is used, the D number of the tool environment is also read, otherwise the D number of the currently active tool is read. If T, D and DL specify a tool without location-dependent offsets, no summed or setup offsets may be specified in parameter <Comp> (error code in <Status>).	

Note

Not all possible combinations of the three parameters <Comp>, <CorComp> and <CorMode> make sense. For example, algorithm 3 in <CorComp> requires that two characters are specified in <Comp>. If an invalid parameter combination is specified, then a corresponding error code is returned in the <Status>.

Examples**Example 1**

Program code	Comment
N10 DEF REAL _CORVAL[3]	
N20 \$TC_DP1[1,1]=120	; Milling tool
N30 \$TC_DP3[1,1]=10.0	; Geometry L1
N40 \$TC_DP12[1,1]=1.0	; Wear L1
N50 _CORVAL[0]=0.333	
N60 T1 D1 G17 G0	

Program code	Comment
N70 R1=SETTCOR(_CORVAL,"G",0,0,2)	
N80 T1 D1 X0 Y0 Z0	; ==> MCS position X0.000 Y0.000 Z1.333
N90 M30	

<CorComp> is "0", therefore, the coordinate value of the geometry component acting in the Z direction must be replaced by the offset value 0.333.

The resulting total tool length is thus: $L1 = 0.333 + 1.000 = 1.333$

Example 2

Program code	Comment
N10 DEF REAL _CORVAL[3]	
N20 \$TC_DP1[1,1]=120	; Milling tool
N30 \$TC_DP3[1,1]=10.0	; Geometry L1
N40 \$TC_DP12[1,1]=1.0	; Wear L1
N50 _CORVAL[0]=0.333	
N60 T1 D1 G17 G0	
N70 R1=SETTCOR(_CORVAL,"W",0,1,2)	
N80 T1 D1 X0 Y0 Z0	; ==> MCS position X0.000 Y0.000 Z11.333
N90 M30	

<CorComp> is "1", this means that the offset value of 0.333 – acting in the Z axis – is added to the wear value of 1.0.

The resulting total tool length is thus: $L1 = 10.0 + 1.333 = 11.333$

Example 3

Program code	Comment
N10 DEF REAL _CORVAL[3]	
N20 \$TC_DP1[1,1]=120	; Milling tool
N30 \$TC_DP3[1,1]=10.0	; Geometry L1
N40 \$TC_DP12[1,1]=1.0	; Wear L1
N50 _CORVAL[0]=0.333	
N60 T1 D1 G17 G0	
N70 R1=SETTCOR(_CORVAL,"GW",0,2,2)	
N80 T1 D1 X0 Y0 Z0	; ==> MCS position X0.000 Y0.000 Z0.333
N90 M30	

<CorComp> is "2", therefore, the offset effective in the Z axis is entered in the geometry component (the old value is overwritten) and the wear value is deleted.

The resulting total tool length is thus: $L1 = 0.333 + 0.0 = 0.333$

Example 4

Program code	Comment
N10 DEF REAL _CORVAL[3]	
N20 \$TC_DP1[1,1]=120	; Milling tool
N30 \$TC_DP3[1,1]=10.0	; Geometry L1

3.21 Working with tool environments

Program code	Comment
N40 \$TC_DP12[1,1]=1.0	; Wear L1
N50 _CORVAL[0]=0.333	
N60 T1 D1 G17 G0	
N70 R1=SETTCOR(_CORVAL,"GW",0,3,2)	
N80 T1 D1 X0 Y0 Z0	; ==> MCS position X0.000 Y0.000 Z11.333
N90 M30	

<CorComp> is "3", therefore, the wear value and compensation value are added to the geometry component and the wear component is deleted.

The resulting total tool length is thus: $L1 = 11.333 + 0.0 = 11.333$

Example 5

Program code	Comment
N10 DEF REAL _CORVAL[3]	
N20 \$TC_DP1[1,1]=120	; Milling tool
N30 \$TC_DP3[1,1]=10.0	; Geometry L1
N40 \$TC_DP12[1,1]=1.0	; Wear L1
N50 _CORVAL[0]=0.333	
N60 T1 D1 G17 G0	
N70 R1=SETTCOR(_CORVAL,"GW",0,3,0)	
N80 T1 D1 X0 Y0 Z0	; ==> MCS position X0.333 Y0.000 Z11.000
N90 M30	

<CorComp> is "3", as in the previous example, but the compensation is now effective on the geometry axis with index "0" (X axis), which for a milling tool, is assigned to tool component L3 due to G17. As a consequence, when calling SETTCOR, tool parameters \$TC_DP3 and \$TC_DP12 are not influenced. Instead, the compensation value is entered in \$TC_DP5.

Example 6

Program code	Comment
N10 DEF REAL _CORVAL[3]	
N20 \$TC_DP1[1,1]=500	; Turning tool
N30 \$TC_DP3[1,1]=10.0	; Geometry L1
N40 \$TC_DP4[1,1]=15.0	; Geometry L2
N50 \$TC_DP12[1,1]=10.0	; Wear L1
N60 \$TC_DP13[1,1]=0.0	; Wear L2
N70 _CORVAL[0]=5.0	
N80 ROT Y-30	
N90 T1 D1 G18 G0	
N100 R1=SETTCOR(_CORVAL,"GW",0,3,1)	
N110 T1 D1 X0 Y0 Z0	; ==> MCS position X24.330 Y0.000 Z17.500
N120 M30	

The tool is a turning tool. A frame rotation is activated in N80, causing the basic coordinate system (BCS) to be rotated in relation to the workpiece coordinate system (WCS). In the WCS, the compensation value (N70) acts on the geometry axis with index 1, i.e. on the X axis

because G18 is active. Since $\langle \text{CorMode} \rangle = 3$, the tool wear in the direction of the X axis of the WCS must become zero once N100 has been executed.

The contents of the relevant tool parameters at the end of the program are thus:

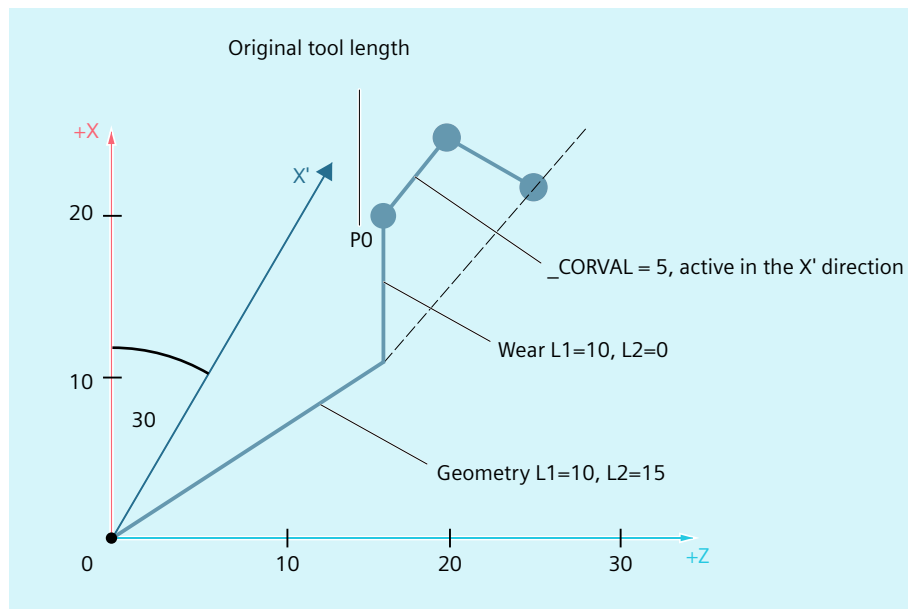
\$TC_DP3[1,1]: 21.830 ; geometry L1

\$TC_DP4[1,1] : 21.830 ; geometry L2

\$TC_DP12[1,1] : 2.500 ; wear L1

\$TC_DP13[1,1] : -4.330 ; wear L2

The geometrical relationships are shown in the figure below: The total wear including $_CORVAL$ is mapped onto the X' direction in the WCS. This produces point P2. The coordinates of this point (measured in X/Y coordinates) are entered in the geometry component of the tool. The difference vector $P_2 - P_1$ remains in the wear. The wear thus no longer has a component in the direction of $_CORVAL$.



If the program example is continued after N110 with the following instructions, then the remaining wear is included completely in the geometry because the compensation is now effective in the Z' axis (parameter $\langle \text{GeoAx} \rangle = 0$):

```
N120 _CORVAL[0]=0.0
N130 R1=SETTCOR(_CORVAL,"GW",0,3,0)
N140 T1 D1 X0 Y0 Z0 ; ==> MCS position X24.330 Y0.000 Z17.500
```

Since the new compensation value is "0", the total tool length and thus the position approached in N140 may not change. If $_CORVAL$ were not equal to "0" in N120, a new total tool length and thus a new position in N140 would result, however, the wear component of the tool length would always be zero, i.e. the total tool length is subsequently always contained in the geometry component of the tool.

3.21 Working with tool environments

The same result as that achieved by calling the SETTCOR function with the <CorComp> = 0 parameter twice can also be reached by calling <CorComp> = 1 (vectorial compensation) just once:

Program code	Comment
N10 DEF REAL _CORVAL[3]	
N20 \$TC_DP1[1,1]=500	; Turning tool
N30 \$TC_DP3[1,1]=10.0	; Geometry L1
N40 \$TC_DP4[1,1]=15.0	; Geometry L2
N50 \$TC_DP12[1,1]=10.0	; Wear L1
N60 \$TC_DP13[1,1]=0.0	; Wear L2
N70 _CORVAL[0]=0.0	
N71 _CORVAL[1]=5.0	
N72 _CORVAL[2]=0.0	
N80 ROT Y-30	
N90 T1 D1 G18 G0	
N100 R1=SETTCOR(_CORVAL,"GW",1,3,1)	
N110 T1 D1 X0 Y0 Z0	; ==> MCS position X24.330 Y0.000 Z17.500
N120 M30	

In this case, all wear components of the tool are set to zero immediately after the first call of SETTCOR in N100.

Example 7

Program code	Comment
N10 DEF REAL _CORVAL[3]	
N20 \$TC_DP1[1,1]=500	; Turning tool
N30 \$TC_DP3[1,1]=10.0	; Geometry L1
N40 \$TC_DP4[1,1]=15.0	; Geometry L2
N50 \$TC_DP12[1,1]=10.0	; Wear L1
N60 \$TC_DP13[1,1]=0.0	; Wear L2
N70 _CORVAL[0]=5.0	
N80 ROT Y-30	
N90 T1 D1 G18 G0	
N100 R1=SETTCOR(_CORVAL,"GW",3,3)	
N110 T1 D1 X0 Y0 Z0	; ==> MCS position X25.000 Y0.000 Z15.000
N120 M30	

When compared to example 6, parameter <CorComp> = 3, and so the <GeoAx> parameter can be omitted. The value contained in _CORVAL[0] now acts immediately on the tool length component L1, the rotation in N80 has no effect on the result, the wear components in \$TC_DP12 are included in the geometry component together with _CORVAL[0], with the result that the total tool length is stored in the geometry component of the tool, due to \$TC_DP13, after the first SETTCOR call in N100.

The geometry component of the tool remains unchanged due to $\langle \text{CorMode} \rangle = 1$. The compensation vector defined in the WCS (rotation around the y axis) must be included in the wear component such that the total tool length in Fig. 3 refers to point P_2 . Therefore, the resulting wear component of the tool is given by the distance of the two points P_1 and P_2 . However, since the wear is evaluated negatively, due to setting data SD42930 \$SC_WEAR_SIGN, the compensation determined in this way has to be entered in the compensation memory with a negative sign. The contents of the relevant tool parameters at the end of the program are thus:

```
$TC_DP3[1,1]: 10.000 ; geometry L1 (unchanged)
$TC_DP4[1,1] : 15.000 ; geometry L2 (unchanged)
$TC_DP5[1,1]: 10.000 ; geometry L3 (unchanged)
$TC_DP12[1,1] : 2.010 ; wear L1 (= 10 - 15 * cos(30) + 10 * sin(30))
$TC_DP13[1,1] : -16.160 ; wear L2 (= -15 * sin(30) - 10 * cos(30))
$TC_DP14[1,1] : -5.000 ; wear L3
```

The effect of setting data SD42930 \$SC_WEAR_SIGN on the L3 component in the Y direction can be recognized without the additional complication caused by the frame rotation.

Additional information

Turning/grinding tools: Calculating the tool length depending on MD20360 \$MC_TOOL_PARAMETER_DEF_MASK

The following machine data defines how the wear and tool length are to be evaluated if a diameter axis is used for turning/grinding tools:

MD20360 \$MC_TOOL_PARAMETER_DEF_MASK.<Bit> = <Value>

<Bit>	<Value>	Meaning
0	0	For turning/grinding tools, the wear parameter of the transverse axis is taken into account in the radius value :
	1	For turning/grinding tools, the wear parameter of the transverse axis is taken into account as the diameter value :
1	0	For turning/grinding tools, the tool length component of the transverse axis is taken into account as the radius value :
	1	For turning/grinding tools, the tool length component of the transverse axis is taken into account as the diameter value :

If the bits involved are set, the associated entry is weighted with a factor of 0.5. The correction using SETTCOR is executed so that the total effective tool length change is equal to the value transferred in $\langle \text{CorVal} \rangle$. If, when calculating the length, a length is evaluated with a factor of 0.5 as a result of machine data MD20360 \$MC_TOOL_PARAMETER_DEF_MASK, then the compensation of this component must be realized with twice the value transferred.

Example

MD20360 \$MC_TOOL_PARAMETER_DEF_MASK = 2 (tool length must be evaluated in the diameter axis using a factor of 0.5)

3.22 Read the assignment of the tool lengths L1, L2, L3 to the coordinate axes (LENTOAX)

Axis X is the diameter axis.

Program code	Comment
N10 DEF REAL _LEN[11]	
N20 DEF REAL _CORVAL[3]	
N30 \$TC_DP1[1,1]=500	; Tool type
N40 \$TC_DP2[1,1]=2	; Cutting edge position
N50 \$TC_DP3[1,1]=3.	; Geometry - length 1
N60 \$TC_DP4[1,1]=4.	; Geometry - length 2
N70 \$TC_DP5[1,1]=5.	; Geometry - length 3
N80 _CORVAL[0]=1.	
N90 _CORVAL[1]=1.	
N100 _CORVAL[2]=1.	
N110 T1 D1 G18 G0 X0 Y0 Z0	; ==> MCS position X1.5 Y5 Z4
N120 R1=SETTCOR(_CORVAL,"G",1,1)	
N130 T1 D1 X0 Y0 Z0	; ==> MCS position X2.5 Y6 Z5
N140 R3=\$TC_DP3[1,1]	; = 5. = (3.000 + 2.*1.000)
N150 R4=\$TC_DP4[1,1]	; = 5. = (4.000 + 1.000)
N160 R5=\$TC_DP5[1,1]	; = 6. = (5.000 + 1.000)
N170 M30	

In each axis, the tool length compensation should be 1 mm (N80 to N100). 1 mm is thus added to the original length in lengths L2 and L3. Twice the compensation value (2 mm) is added to the original tool length in L1, in order to change the total length by 1 mm as required. If the positions approached in blocks N110 and N130 are compared, it can be seen that each axis position has changed by 1 mm.

3.22 Read the assignment of the tool lengths L1, L2, L3 to the coordinate axes (LENTOAX)

The LENTOAX function provides information about the assignment of tool lengths L1, L2 and L3 of the **active** tool to the abscissa, ordinate and applicate. The assignment of abscissa, ordinate and applicate to the geometry axes is affected by frames and the active plane (G17 - G19).

Only the geometry component of a tool (\$TC_DP3[<t>,<d>] to \$TC_DP5[<t>,<d>]) is considered, i.e. a different axis assignment for other components (e.g. wear) has no effect on the result.

Syntax

```
<Status> = LENTOAX(<AxInd>, <Matrix>[, <Coord>])
```

Principle

LENTOAX(...):	Predefined function to read the assignment of tool lengths L1, L2 and L3 of the active tool to the coordinate axes	
	Alone in the block:	Yes

3.22 Read the assignment of the tool lengths L1, L2, L3 to the coordinate axes (LENTOAX)

<Status>:		Function return value. Negative values indicate error states.		
		Data type: INT		
		Value:	0	Function OK Information provided in <AxInd> is sufficient for the description (all tool length components are in parallel to the geometry axes).
			1	Function is OK, however, the content of <Matrix> must be evaluated for a correct description (the tool length components are not parallel to the geometry axes).
			-1	Invalid string in parameter <Coord>.
			-2	No tool active.
Parameters				
1	<AxInd>:	If the tool length components are parallel to the geometry axes, the axis indices assigned to length components L1 to L3 are returned in the <AxInd> array. • <AxInd> [0]: Abscissa • <AxInd> [1]: Ordinate • <AxInd> [2]: Applicate		
Data type: INT[3]				
Value:		0	No assignment exists (axis does not exist)	
		1 ... 3 or	Number of the length effective in the corresponding coordinate axis.	
		-1 ... -3	The sign is negative if the tool length component is pointing in the negative coordinate direction.	
		If not all length components are parallel or antiparallel to the geometry axes, the index of the axis, which contains the largest part of a tool length component, is returned in <AxInd>. In this case (if the function does not return an error for a different reason), then the return value is <Status> = 1. The mapping of tool length components L1 to L3 to geometry axes 1 to 3 is then described completely by the content of the 2nd parameter <Matrix>.		
2	<Matrix>:	Matrix which represents the vector of the tool lengths (L1=1, L2=1, L3=1) to the vector of the coordinate axes (abscissa, ordinate, applicate), i.e. the tool length components are assigned to the columns in the order L1, L2, L3 and the axes are assigned to the lines in the order abscissa, ordinate, applicate.		
Data type: REAL				
All elements are always valid in the matrix, even if the geometry axis belonging to the coordinate axis is not available, i.e. if the corresponding entry in <AxInd> is 0.				

3.22 Read the assignment of the tool lengths L1, L2, L3 to the coordinate axes (LENTOAX)

3	<Coord>:	coordinate system applicable for the assignment (optional)		
		Data type:		STRING
		Charac- ters:	MCS M	The tool length is represented in the machine coordinate sys- tem.
			BCS B	The tool length is represented in the basic coordinate system.
			WCS W	The tool length is represented in the workpiece coordinate system (default setting).
			KCS K	The tool length is represented in the tool coordinate system of the kinematic transformation.
			TCS T	The tool length is represented in the tool coordinate system.
The notation of the characters in the string (upper or lower case) is arbitrary. If the parameter <Coord> is not specified, then WCS is used (default setting).				

Note

In the TCS, all tool length components are always parallel or antiparallel to the axes.

The components can only be antiparallel when mirroring is active and the following setting data is activated:

SD42900 \$SC_MIRROR_TOOL_LENGTH (sign change tool length when mirroring)

Example

Standard application, milling tool for G17.

L1 applies in Z (applicate), L2 applies in Y (ordinate), L3 applies in X (abscissa).

Function call in the form:

```
<Status>=LENTOAX (<AxInd>, <Matrix>, "WCS")
```

The result parameter <AxInd> then contains the values:

```
<AxInd>[0] = 3
```

```
<AxInd>[1] = 2
```

```
<AxInd>[2] = 1
```

Or, in short: (3, 2, 1)

In this case, the associated matrix (<Matrix>) is:

$$\text{<Matrix>} = \begin{pmatrix} 0 & 0 & 1 \\ 0 & 1 & 0 \\ 1 & 0 & 0 \end{pmatrix}$$

A change from G17 to G18 or G19 does not alter the result, because the assignment of the length components to the geometry axes changes in the same way as the assignment of the abscissa, ordinate and applicate.

A frame rotation of Z through 60 degrees is now programmed with G17 active, e.g.

3.23 Boundary conditions

ROT Z60

The direction of the applicator (Z direction) remains unchanged; the main component of L2 now lies in the direction of the new X axis; the main component of L1 now lies in the direction of the negative Y axis. As a consequence, the return value (<Status>) is "1", <AxInd> contains the values (2, -3, 1).

In this case, the associated matrix (<Matrix>) is:

$$\langle \text{Matrix} \rangle = \begin{pmatrix} 0 & \sin 60^\circ & \cos 60^\circ \\ 0 & \cos 60^\circ & -\sin 60^\circ \\ 1 & 0 & 0 \end{pmatrix}$$

3.23 Boundary conditions

3.23.1 Tools with a relevant tool point direction

The following must be observed for tools with relevant cutting edge position:

- The straight line between the tool edge center points at the block start and block end is used to calculate intersection points with the approach and retraction block. The difference between the tool edge reference point and the tool edge center point is superimposed on this movement.
For approach and/or retraction with **KONT**, the movement is superimposed in the linear subblock of the approach or retraction movement. Therefore, the geometric conditions for tools with or without relevant cutting edge position are identical.
- In circle blocks and in motion blocks containing rational polynomials with a denominator degree > 4, it is not permitted to change a tool with active tool radius compensation in cases where the distance between the tool edge center point and the tool edge reference point changes. With other types of interpolation, it is now possible to change when a transformation is active (e.g. **TRANSMIT**).
- For tool radius compensation with variable tool orientation, the transformation from the tool edge reference point to the tool edge center point can no longer be performed by means of a simple zero offset. Tools with a relevant cutting edge position are therefore not permitted for 3D peripheral milling (an alarm is output).

Note

The subject is irrelevant with respect to face milling as only defined tool types without relevant cutting edge position are permitted for this operation anyway. (A tool with a type, which has not been explicitly approved, is treated as a ball end mill with the specified radius. A cutting edge position parameter is ignored).

3.23.2 SD42935 expansions

Transformation of wear values

Which wear components in conjunction with the instructions `TOWMCS` (wear values in the machine coordinate system) and `TOWWCS` (wear values in the workpiece coordinate system) are transformed or not transformed can be defined using:

SD42935 `$SC_WEAR_TRANSFORM` (transformation of wear values)

3.23.3 Scratching

Scratching

If the "Scratch" function is used in the reset state, then regarding the initial setting value, the following must be taken into consideration:

- The wear components are evaluated depending on the initial settings of the G groups `TOWSTD`, `TOWMCS` and `TOWWCS`.
- If a value other than the initial setting is needed to ensure correct calculation, scratching may be performed only in the STOP state.

3.23 Boundary conditions

3D tool radius compensation

4.1 Function

4.1.1 Introduction

3D circumferential milling and 3D face milling

3D tool radius compensation (3D-TRC) is used to process contours with tools that can be controlled in their orientation independently of the tool path and shape.

SINUMERIK provides the 3D-TRC in different variants, which are used in combination with the following manufacturing processes:

- Circumferential milling (Page 282) for structural components
- Face milling (Page 288) for free-form surfaces

Tool orientation

For the 3D-TRC, a distinction must be made between the two scenarios of a tool with a fixed orientation and a tool with a changeable orientation.

The tool radius compensation with a fixed tool for circumferential milling is designated as a 2½D-TRC. This is the conventional tool radius compensation. Circumferential milling with a functionality that goes beyond this scope of performance is not possible for the compensation with a fixed tool. For circumferential milling, there is only a 3D-TRC if the tool orientation is changeable. This means, in addition to the 3 degrees of freedom for positioning the tool (generally 3 linear axes), two additional degrees of freedom (2 rotary axes) are also needed for setting the tool orientation (5-axis machining).

During face milling, the tool orientation can remain constant or be variable. The advantages of variable orientation are:

- better approximation of the end contour
- increased chip performance
- more freedom in selecting the shape of the tool
- a wider range of surfaces that can be processed (undercuts)

Note

5D-WRK

The 3D-TRC is also called 5D-TRC below, because 5 degrees of freedom are available in this case for defining the position of the tool in space relative to the workpiece.

4.1 Function

Parameterization

The machine and setting data set for the 2D-TRC are also in effect for the 3D-TRC.

Further information

Tool offset (Page 19)

In addition to this, there is special system data that is only relevant to 3D-TRC. See Chapter "Parameter assignment (Page 296)" regarding settings of this data.

Activation/deactivation

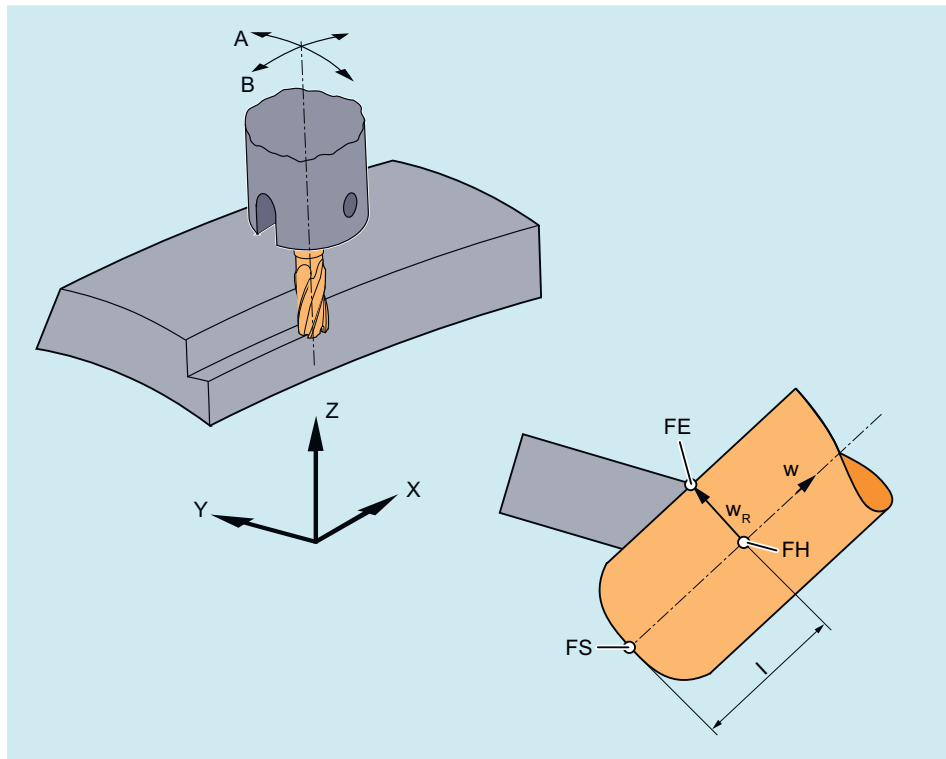
The selection of the 3D-TRC variant is made in the NC program via G commands of Group 22 (tool offset type). The actual activation and deactivation are done via the G commands of Group 7 (tool radius compensation).

See Section "Programming (Page 297)".

4.1.2 Circumferential milling

The 3D circumferential milling is used for machining contours with tools having an orientation that is not constant during processing. Processing is done using the peripheral surface of the tool. If the tool is cylindrical, ruled surfaces (e.g. cones, cylinders, etc.) are created during this processing. Other forms of milling machines can also be used, however (e.g. form cutters). The tool radius compensation needed for this type of processing is an extension of the conventional 2½D-TRC.

The ratios for 3D circumferential milling are shown in the following figure:



FE: Milling tool machining point

FS: Milling tool tip

FH: Milling tool reference point

l: Insertion depth

w: Tool vector

w_r : Vector from the milling tool reference point to the milling tool machining point with length of shank radius R

Figure 4-1 Circumferential milling

Insertion depth

The insertion depth of the milling tool is the distance of the milling tool reference point from the tip of the tool.

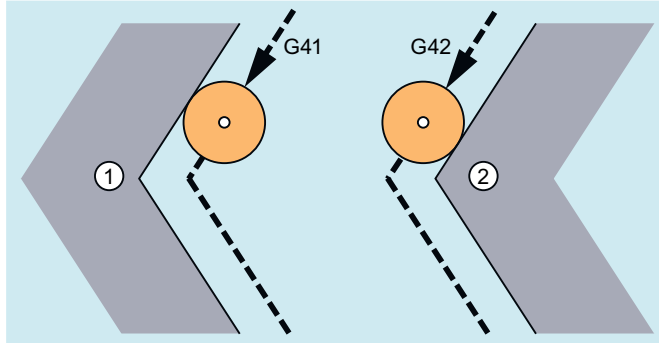
The milling tool reference point is the vertical projection of the milling tool machining point on the programmed path to the longitudinal axis of the tool.

The position of the machining point on the peripheral surface of the tool is set with the insertion depth.

Via the address ISD (InSertion Depth), the insertion depth of the tool for circumferential milling and **active** 3D tool radius compensation can be changed (see Chapter "Programming (Page 297)").

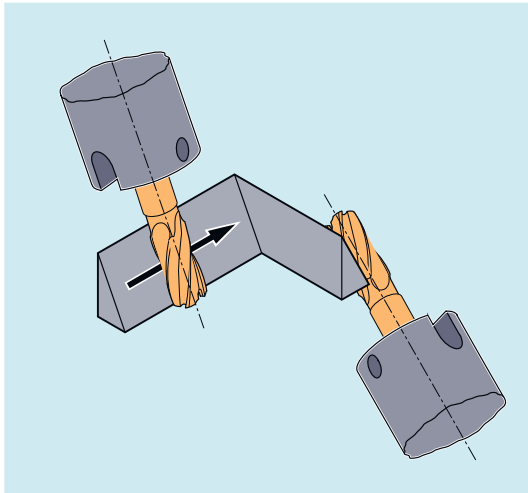
4.1.2.1 Corners in circumferential milling

Inside and outside corners are handled separately. The terms inside corner and outside corner are dependent on the tool orientation.



- ① Inside corner
- ② Outside corner

When the orientation changes at a corner, for example, the corner type may change while machining is in progress:



Whenever this occurs, the machining operation is aborted with alarm 10770.

4.1.2.2 Behavior at outer corners

At outside corners during circumferential milling with 3D-TRC analogous to the conditions in the 2½D-TRC, the G commands of Group 18 (corner behavior tool offset) are evaluated:

- **G450: Transition circle (tool travels around workpiece corners on a circular path)**
Outside corners are treated like circles with a radius of 0, where the circular plane stretches from the end tangent of the first block and the start tangent of the second block. Thus, an orientation change during the block transition is also possible. Thus, a block is always inserted at an outside corner.
In contrast to the solution for the 2½D-TRC, the contour element inserted at an outside corner is always a circle with a radius of 0, which is affected by the tool radius compensation like on any other programmed path. It is not possible to insert conical sections in place of the circles. The address DISC therefore has no relevance in this case and is not evaluated.
- **G451: Intersection of equidistant paths (tool backs off from the workpiece corner)**
The intersection is determined by extending the offset curves of the two participating blocks and defining the intersection of the two blocks at the corner in the plane perpendicular to the tool orientation.
If an intersection is found in the plane perpendicular to the tool, this does not mean that the curves also intersect in space. Rather the curves in the direction of the tool longitudinal axis are considered, which are generally a certain distance apart. The positional offset is eliminated over the entire block length in direction of the tool.
The way this offset is processed in tool direction at outside corners is the same as for inside corners.

For nearly tangential transitions, the behavior for an active G450 is the same as for G451 (limit angle can be set via MD20210 \$MC_CUTCOM_CORNER_LIMIT). Conversely, if G451 is active, a circle is also inserted (procedure as for G450) if there is no intersection or if the corner angle exceeds a specific value (MD20230 \$MC_CUTCOM_CURVE_INSERT_LIMIT).

Behavior during orientation changes

The intersection procedure (G451) is not used when at least one block containing a change to the tool orientation was inserted between the traversing blocks in question. In this case a circle is always inserted at the corner.

The G commands of Group 27 (too offset for orientation change at outside corners) can be used to define whether orientation changes, which have been programmed between the two traversing blocks that form the corner, are executed before the beginning of the inserted circle set or at the same time.

- **ORIC:** Orientation changes at outside corners are superimposed on the circle block to be inserted.
- **ORID:** Orientation changes are performed before the circle block.

If two or more blocks with orientation changes (e.g. A2=... B2=... C2=...) are programmed between two traversing blocks and ORIC is active, then the circular block to be inserted is divided between these intermediate blocks corresponding to the amount of the individual angle changes. Further programmed intermediate blocks without traversing and orientation motions are executed at the programmed positions.

If ORID is active, all inserted blocks (both those with and without orientation movement) are executed at the end of the first of the two traversing blocks. The tangent in the end point

of the first traversing block is used for the offset calculation. The circle block with constant orientation is inserted immediately before the second traversing block.

See also "Example 1: Orientation change at outside corner during 3D circumferential milling (Page 313)".

Intermediate blocks without relevant traversing information

Intermediate blocks without relevant traversing information (neither tool orientation nor position of geometry axes are changed) are permissible. The intersection procedure is applied to the adjacent blocks as if these intermediate blocks did not exist. In the same manner, tool direction motions in the tool direction may also be programmed in intermediate blocks.

4.1.2.3 Behavior at inside corners

For the 3D-TRC, only adjacent traversing blocks are taken into consideration for the calculation of the intersecting point.

If the orientation is not changed at the block limit, then the contour need only be considered in the plane vertical to the tool axis. In this case, the tool cross-section is a circle which touches the two contours. The geometric relations in this plane are identical to those for $2\frac{1}{2}$ D-TRC.

Behavior during orientation changes

If the orientation changes on a block transition, the tool moves in the inside corner so that it is constantly in contact with the two blocks forming the corner.

Path segments must be long enough to ensure that the contact points of the tool do not cross the block limits into other blocks when the orientation changes at an inside corner.

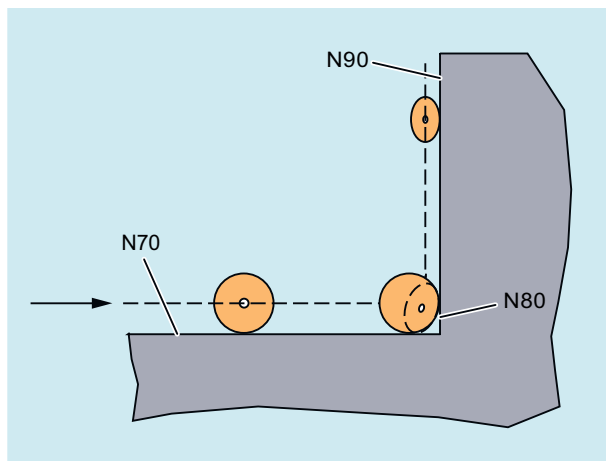


Figure 4-2 Orientation changes at inside corner

When the orientation changes in a block that is one of the two blocks forming the inside corner, then it is no longer possible to adhere to the programmed relationship between path position and associated orientation. This is because the orientation must reach its end value even though the path end position is not reached. This response is identical to the response of synchronized axes with $2\frac{1}{2}$ D tool radius compensation.

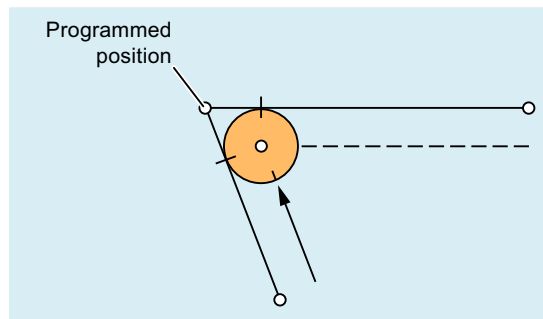


Figure 4-3 Path end position and change in orientation at inside corners

See also "Example 2: Orientation change at inside corner during 3D circumferential milling (Page 315)".

Change in insertion depth

Generally speaking, the contour elements that form an inside corner are not positioned on the plane perpendicular to the tool. This means that the contact points between the two blocks and the tool are at different distances from the tool tip.

This means: The insertion depth (ISD) changes abruptly from the 1st to the 2nd block at an inside corner.

To ensure that this difference in depth is not an abrupt step change, it is distributed continuously among the blocks involved during interpolation. The depth-compensating motion is executed in the current tool direction.

This solution prevents the contour from being violated by cylindrical tools if the length of the tool prevents the cutter contact point on the lateral surface of the cutter leaving the range in which machining is possible.

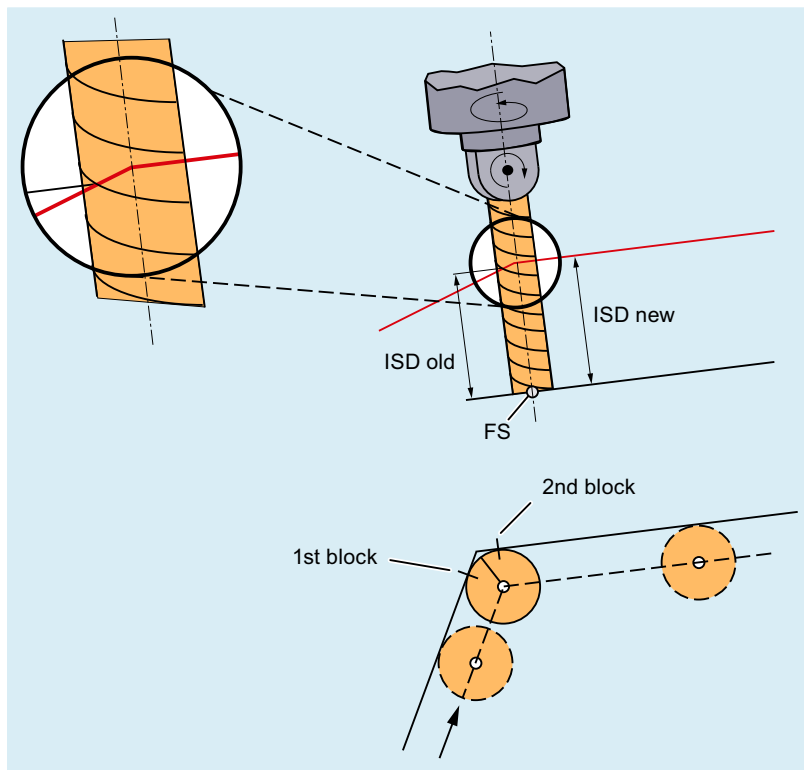


Figure 4-4 Change in insertion depth

4.1.2.4 Monitoring of path curvature

The path curvature is monitored for an unchangeable orientation in the following way:

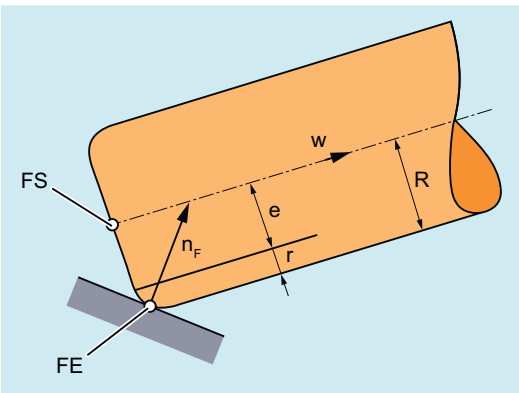
1. The contour in each block is projected on the plane that is orthogonal to the tool orientation.
2. Then the curvature of the projection is analyzed. Areas with too much curvature (tool does not fit in the curve) are cut out.
3. The remaining offset contours overlap each other and thus form the global offset contour.

4.1.3 Face milling

Face milling is used to process curved surfaces of any kind. In this case, the longitudinal axis of the tool and the surface normal vector are more or less parallel. In contrast, the longitudinal axis of the tool and the surface normal vector of the surface to be machined in a 3D circumferential milling operation are at right angles to one another.

Information about the surface to be machined is absolutely essential for face milling operations, i.e. a description of the linear path in space is not sufficient. The tool shape must also be known in order to implement the tool offset (the term "Tool radius compensation" is not appropriate in this case).

The conditions for face milling are shown in the following figure:

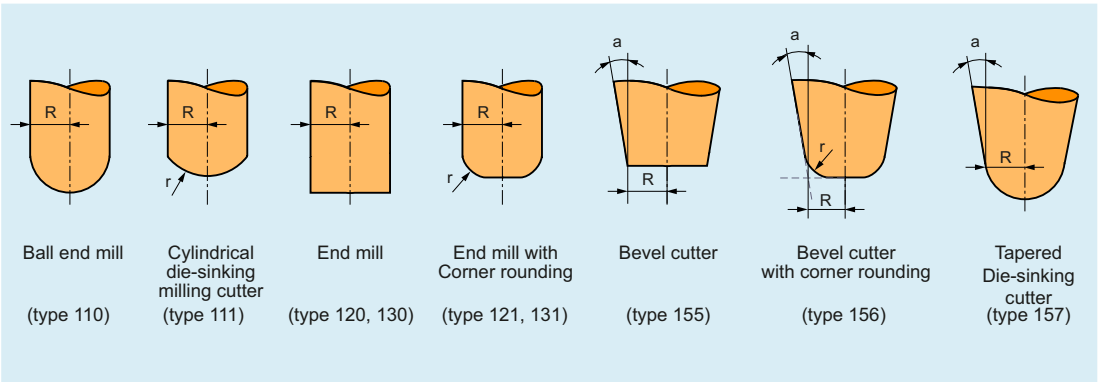


- FE: Milling tool machining point
- FS: Milling tool tip
- n_F : Surface normal vector
- w : Tool vector
- R : Shaft radius
- r : Corner radius
- e : Difference between the shank radius R and corner radius r

Figure 4-5 Face milling with a torus cutter

4.1.3.1 Tool shapes and tool data for face milling

An overview of the tool shapes, which may be used for face milling operations and relevant tool data are listed in the following. The shape of the tool shaft is not taken into account. Therefore, the tool types 120 and 155, for example, have an identical effect.



- R = shank radius (tool radius)
- r = corner radius
- a = angle between the tool longitudinal axis and upper end of the torus surface

Cutter type	Type No.	R	r	a
Ball end mill	110	> 0	-	-
Cylindrical die-sinking milling cutter	111	> 0	$> R$	-
End mill, angle head cutter	120, 130	> 0	-	-

4.1 Function

Cutter type	Type No.	R	r	a
End mill, angle head cutter with corner rounding	121, 131	> r	> 0	-
Bevel cutter	155	> 0	-	> 0
Bevel cutter with corner rounding	156	> 0	> 0	> 0
Tapered die-sinking cutter	157	> 0	-	> 0

- : is not evaluated.

Tool data	Tool parameters	
Tool dimensions	Geometry	Wear
R	\$TC_DP6	\$TC_DP15
r	\$TC_DP7	\$TC_DP16
a	\$TC_DP11	\$TC_DP20

Note

If, in the NC program, a type number is specified that differs from that in the diagram, then the system automatically uses tool type 110 (ball end mill).

Note

An alarm is output if the tool data limit values are violated.

Geometry and wear values of tool data have an additive effect, i.e. the tool form that is actually used for the offset calculation is always described by the sum values.

In the CUT3DFD function (differential tool), only the wear value is considered as an offset value. The geometry values are only necessary in this case for describing the tool form.

Tool length offset

The tool tip (intersecting point between the tool longitudinal axis and the workpiece surface) is considered the reference point for the length offset.

Tool change

A new tool with changed dimensions (R, r, a) or other form can only be specified during the first activation of the tool offset (i.e. during the transition from G40 to G41 or G42) or for an already active offset only for new programming of G41 or G42. This rule does not apply to any other tool data, e.g. tool lengths, so that tools can be loaded without reprogramming G41 or G42.

4.1.3.2 Face milling with a specified surface normal vector

Expanded options for orientation programming are provided for 3D face milling.

The tool offset for face milling cannot be calculated simply by specifying the path (e.g. of a line in space). The surface to be machined must also be known. The necessary information about the surface is provided to the controller with the surface normal vector.

The surface normal vector at the block beginning is programmed with A4, B4, C4, and the vector at the block end with A5, B5, C5. Components of the surface normal vectors

which are not programmed are set to zero. The length of a vector programmed in this way is irrelevant. A vector of zero length (all three components are zero) is ignored, i.e. the direction programmed beforehand remains valid, no alarm is generated.

If only the start vector is programmed (A4, B4, C4) in a block, then the programmed surface normal vector remains constant over the entire block. If only the end vector is programmed (A5, B5, C5), then large-circle interpolation is used to interpolate between the end value of the preceding block and the programmed end value. If both start and end vectors are programmed, interpolation according to the large circle principle is also performed between the two directions. The fact that the start vector may be reprogrammed in a block means that the direction of the surface normal vector can change irregularly on a block transition. Irregular transitions of the surface normal vector always occur in cases where there is no tangential transition between the surfaces (planes) involved, i.e. if they form an edge.

Once a surface normal vector is programmed, it remains valid until a new vector is programmed. In the basic position, the surface normal vector is set to equal the vector in the Z direction. This basic setting direction is independent of the active plane (G17 - G19). If ORIWKS is active, surface normal vectors refer to the active frame, i.e. when the frame is rotated, the vectors rotate simultaneously. This applies both to programmed orientations and to those that have been derived from the active plane. If ORIWKS is active, the surface normal vectors are brought along during a frame change. An orientation modified as the result of frame rotations is not returned to its original state on switchover from ORIWKS to ORIMKS.

It must be noted that the programmed surface normal vectors may not necessarily be the same as those used internally. This is always the case if the programmed surface normal vector is not vertical on the path tangent. A new surface normal vector is then formed, which lies within the plane spanned by the path tangent and programmed surface normal vector, but which is vertical on the path tangent vector. This orthogonality is required, because the path tangent vector and the surface normal vectors (must) always be vertical to one another in a real surface. However, since the two variables can be programmed independently of one another, they can contain information that is mutually contradictory. Orthogonalization ensures that the information contained in the path tangent vector has priority over the data in the surface normal vector. An alarm is output if the angle between the path tangent vector and the programmed surface normal vector undershoots the limit value, which is defined in the following machine data:

MD21084 \$MC_CUTCOM_PLANE_PATH_LIMIT (minimal angle between surface normal vector and path tangent vector)

If a block is shortened (inside corner), then the interpolation range of the surface normal vector is reduced accordingly, i.e. the end value of the surface normal vector is not reached as it would be with other interpolation quantities such as, for example, the position of an additional synchronized axis.

In addition to the usual methods of programming orientation, it is also possible to refer the tool orientation to the surface normal vector and path tangent vector using the addresses LEAD (lead or camber angle) and TILT (side angle). Interpretation of the angle specifications is thus dependent on the setting in MD21094 \$MC_ORIPATH_MODE (see Function Manual "Transformations", Section "Multi-axis transformations"). The specification of the angle relative to the surface normal is only an expanded option of orientation programming at the end of the block. It does not imply that the lead and side angles reach their programmed values before the path end point is reached.

The resulting tool orientation is determined from the path tangent, surface normal vector, lead angle and side angle at the end of the block. This orientation is always implemented by

the end of the block, particularly in cases where the block is shortened (at an inside corner). If the omitted path section is not a straight line in a plane, the lead and side angles generally deviate from their programmed values at the path end point. This is because the orientation has changed relative to the surface normal vector or path tangent vector when the absolute orientation of the tool is the same as at the original path end point.

4.1.3.3 Compensation on path

A special case must be examined with respect to face milling operations, i.e. that the machining point on the tool surface moves around. This may be the case on a torus cutter whenever surface normal vector n_F and tool vector w become collinear (i.e. the tool is at exact right angles to the surface) since it is not a single point on the tool that corresponds to this direction, but the entire circular surface on the tool end face. The contact point is therefore not defined for this orientation. A path point in which tool longitudinal axis and surface normal are parallel is therefore referred to below as a singular point or a singularity.

The above case is also meaningful in practical terms, e.g. in cases where a convex surface, which may have a vertical surface normal (e.g. hemisphere), must be machined with a perpendicular tool (e.g. face milling with constant orientation). The machining point on the contour remains fixed, but the machine must be moved to bring the machining point from one side of the tool to the other.

The problem described is only a borderline case (lead angle $\beta = 0$ and side angle $\gamma = 0$). If the lead angle $\beta = 0$ and the side angle γ has a low value, then the tool must be moved very rapidly (in borderline case in steps) to keep the machining point resulting from the milling conditions close to the arc-line forming the end face (see the following figure).

In principle, it is resolved by the smoothing of the surface normal and the smoothing of the contour.

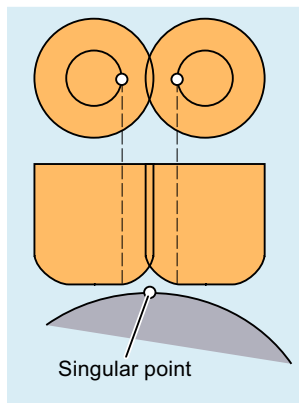


Figure 4-6 Switching of the processing point on the tool surface in the vicinity of a point in which the surface normal vector and the tool orientation are parallel.

Singularities do not only occur in isolated points, but can also occur over entire curves. This is the case, for example, if the curve to be interpolated is a plane curve (i.e. a curve with a constant osculating plane) and the tool is constantly aligned in parallel to the binormal vector (perpendicular to the osculating plane). A simple example of this is an arc in the X-Y plane, which is processed with a tool that is aligned parallel to the Z axis. For paths of this type, the tool offset is reduced to a tool length compensation. I.e. the tool is positioned in such a way that its tip lies on the programmed path.

For a jump-like transition between singular and non-singular curves, linear blocks must be inserted, just as in the handling of individual points, so that the processing point on the tool can migrate from the tool tip to the periphery (at outside corners or convex surfaces) or the paths must be shortened to prevent contour violations (at inside corners or concave surfaces).

It is therefore necessary to activate both the smoothing of the contours ("Top Surface" function) and the smoothing of the surface normal to ensure that the transitions run smoothly.

4.1.3.4 Corners in face milling

Two surfaces that do not tangentially transition into one another form an edge. The paths defined on the surfaces form a corner. This corner is a point of the edge.

The type of corner (inside or outside corner) is defined by the surface normals of the participating surfaces and the paths defined on them.

The surface normals of the two surfaces forming the edge may point in opposite directions of the overall surface (the front edge of one surface is continued on the rear edge of the second surface), see also the following Figure. Transitions of this type are not allowed and are rejected with an alarm.

The scalar product of the surface normal vector and (possibly variable) tool orientation on one corner/path must be positive at each point, i.e. it is not permissible to machine from the rear face of the surface. If this condition is not fulfilled, an alarm is output. The permissible ranges of validity of tool orientation for inside and outside corners are illustrated in the following Figure. These ranges are further restricted by the condition that the angle between the surfaces to be machined and the "steepest" surface line of the tool surface must not be lower than a particular machine data setting. The "steepest" surface line is a line at angle α to the tool longitudinal axis (this line is in the same direction as the tool longitudinal axis on cylindrical tools). This restriction must be imposed to ensure that the contact point on the tool does not leave the permissible range.

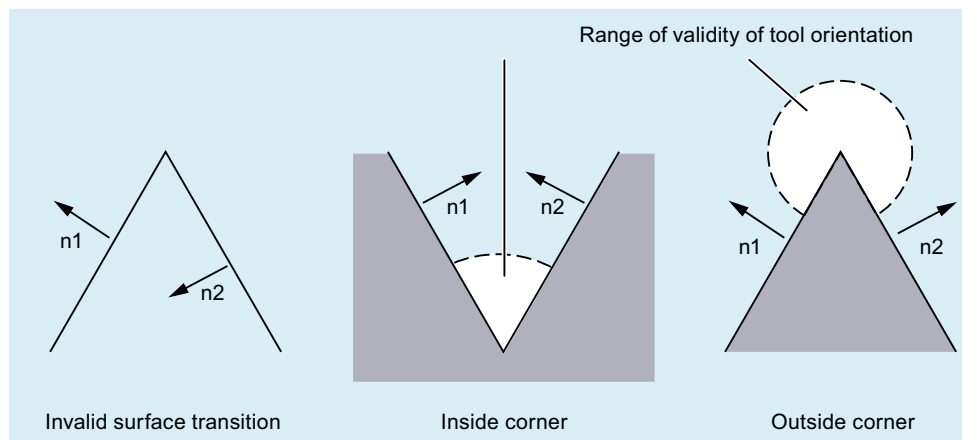


Figure 4-7 Corners in face milling

It is possible to insert blocks that contain no motion commands (e.g. auxiliary function outputs) and/or that include motions of geometry axes not involved in the path between two blocks which contain a path definition. In particular, orientation changes can also be

programmed in such blocks. The only exception applies to the activation and deactivation of the 3D tool radius compensation function: Intermediate blocks with orientation changes are not permitted between the activation block and the first corrected block or between the last corrected block and the deactivation block. Other intermediate blocks are allowed, however.

4.1.3.5 Behavior at outer corners

In face milling, outside corners are treated like circles with a radius of 0, where the circular plane stretches from the end tangent of the first block and the start tangent of the second block. Thus, an orientation change during the block transition is also possible. Thus, a circle is always inserted as a contour element at an outside corner. The intersecting point process is not available for face milling.

Behavior during orientation changes

The G commands of Group 27 (too offset for orientation change at outside corners) can be used to define whether orientation changes, which have been programmed between the two traversing blocks that form the corner, are executed before the beginning of the inserted circle set or at the same time.

- ORIC: Orientation changes at outside corners are superimposed on the circle block to be inserted.
- ORID: Orientation changes are performed before the circle block.

If two or more blocks with orientation changes (e.g. A2=... B2=... C2=...) are programmed between two traversing blocks and ORIC is active, then the circular block to be inserted is divided between these intermediate blocks corresponding to the amount of the individual angle changes. Further programmed intermediate blocks without traversing and orientation motions are executed at the programmed positions.

If ORID is active, all inserted blocks (both those with and without orientation movement) are executed at the end of the first of the two traversing blocks. The tangent in the end point of the first traversing block is used for the offset calculation. The circle block with constant orientation is inserted immediately before the second traversing block.

See also "Example 1: Orientation change at outside corner during 3D circumferential milling (Page 313)".

4.1.3.6 Behavior at inside corners

The position of the tool in which it is in contact with the two surfaces forming the corner must be determined at an inside corner. The contact points must be on the paths defined on both surfaces. It is not usually possible to solve this problem exactly since, when the tool is moved along the path on the first surface, it normally touches a point on the second surface which is not on the path.

For this reason, the tool is not moved along the path on the first surface, but deviates from the path in such a way as to ensure that the distance between the contact points and the relevant contours in the position in which the tool contacts both surfaces is minimal, see also the following Figure.

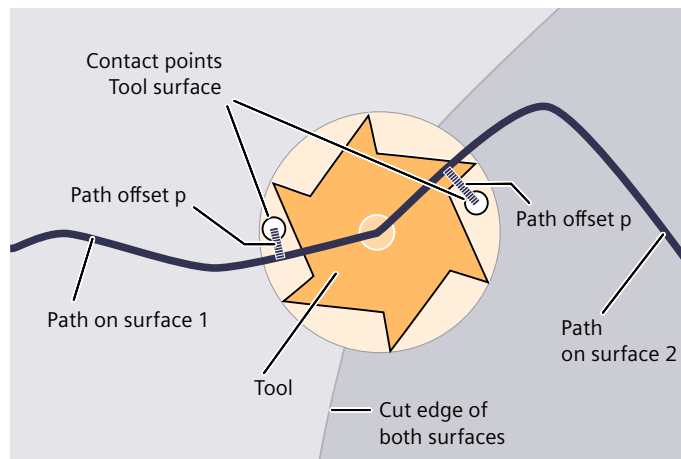


Figure 4-8 Inside corner with face milling (view in direction of longitudinal axis of tool)

Note

The amount by which the contact points deviate from the programmed contour will generally be small since the explanatory example shown in the Figure, in which the machining point "changes" the cutter side at an inside corner (the value of the angular difference Ψ about the tool longitudinal axis between the two contact points on the tool surface is approximately 180°) is more likely to be the exception (see also the following Figure, on the right). The angle Ψ will normally stay almost constant so that the distance between the contact points on the tool surface will be relatively small (see also the following Figure, on the left).

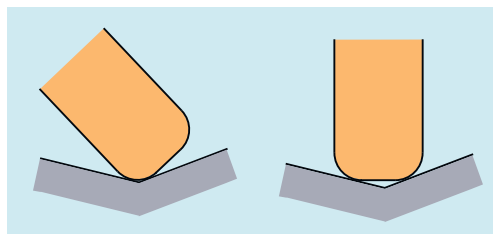


Figure 4-9 Machining at inside corners

The difference between the programmed point on the path and the point actually to be approached (path offset p) is eliminated linearly over the entire block length. Differences resulting from inside corners at the block start and block end are overlaid. The current difference in a path point is always perpendicular to the path and in the surface defined by the surface normal vector.

If the tool orientation at an inside corner is not constant, the change in orientation is implemented in the same way as described in Subsection "Behavior at inside corners" for 3D circumferential milling, i.e. the tool is moved in the corner so that it contacts the two adjacent surfaces at the block start, block end and at two points $1/3$ and $2/3$ of the change in orientation. A 3rd-degree polynomial is used to interpolate between these 4 points.

A variable tool orientation in a block that is shortened owing to an inside corner is also treated in the same way as described in Subsection "Behavior at inside corners" for 3D milling (Page 286), i.e. the entire change in orientation is executed in the shortened block. Consequently, the functional relationship between path tangent, surface normal and tool orientation also changes. This results in new, previously nonexistent singularities

or impermissible side angles (at points which are virtually singular) occurring in the shortened block. If this type of situation is detected during processing of an inside corner, the machining operation is aborted with an alarm. No block division takes place at the singular points since the compensatory motions this would involve frequently cause contour violations and the change in machining side on the tool is not generally intended or even foreseen by the user. The alarm is also output during examination of an inside corner if the singularity occurs in the second of the two blocks without the transition to the next block being considered. The system does not therefore detect that a block of this type will form an inside corner in conjunction with the following block and that the singularity would be eliminated again by the second block reduction.

Remedy: Create new/different NC program, e.g. with a smaller tolerance.

The surface normal vector $\mathbf{n}_F$ is not affected by the reduction of a block. This means that in contrast to the tool orientation, the change in orientation that may need to be executed for this vector will not be imaged onto the reduced traversing interval. This is necessary because a surface other than that programmed would be machined. Unlike the tool orientation, no problems arise as the result of an abrupt change in the surface normal vector at a block transition since it does not reflect any axis motions.

Analogously to 3D circumferential milling, (see Section "Behavior at inside corners (Page 286)"), the two traversing blocks that form an inside corner must contain contact points. There is no evaluation of several traversing blocks (i.e. no bottleneck detection), CDON/CDOF are not evaluated. If no contact point can be found, the machining operation is aborted with an alarm (risk of collision).

4.1.3.7 Monitoring of path curvature

The path curvature is only monitored for circumferential milling. As a rule, it is therefore not detected for CUT3DF* functions when an attempt is made to process a concave surface which is so intensively curved that machining is not possible with the tool being used. A possible exception are blocks that are split owing to a singularity. The transition between the two partial blocks created after the split is then treated like an inside corner. Apart from these special cases, it is the responsibility of the user to only use tools that make it possible to process every part of the contour without contour violation. It could be helpful to use the CUT3DFD function for this, where the differential tool is only minimally different from the original tool.

4.2 Commissioning

4.2.1 Parameter assignment

Minimal angle between path tangent and tool orientation

For the 3D-TRC, the angle between the path tangent and the tool orientation cannot undershoot a specific limit angle.

This angle is defined in the machine data:

MD21080 \$MC_CUTCOM_PARALLEL_ORI_LIMIT

The lower the value of MD21080, the higher the computation effort generally is, which is needed to check adherence to the cited conditions. Exceptions apply to linear blocks with constant orientation.

Minimal angle between surface normal vector and tool orientation

In 3D face milling, the following machine data specifies the minimum angle which the surface normal vector and tool orientation must form in each point of the path if you are working with a side angle that is not equal to zero and the tool is not a cherry:

MD21082 \$MC_CUTCOM_PLANE_ORI_LIMIT

If this value is undershot, processing is canceled with an alarm.

The lower the value of MD21082, the higher the computation effort generally is, which is needed to check adherence to the cited conditions.

MD21082 has no effect in linear blocks with constant orientation. In this case, angles of any size are permitted, even if the side angle is not equal to zero.

Minimal angle between surface normal vector and path tangent vector

During 3D face milling, the following machine data specifies the minimum angle which the surface normal vector and path tangent vector must form in each point of the path:

MD21084 \$MC_CUTCOM_PLANE_PATH_LIMIT

If this value is undershot, processing is canceled with an alarm.

The lower the value of MD21084, the higher the computation effort generally is, which is needed to check adherence to the cited conditions.

Interpolation of the surface normals via polynomials

In order to use smoothing of the surface normal in 3D face milling, the function "Interpolation of the surface normal via polynomials" must be enabled:

MD28291 \$MC_MM_SMOOTH_SURFACE_NORMALS = TRUE

If MD28291 is not set, processing is canceled with an alarm.

4.3 Programming

4.3.1 Selecting 3D tool radius compensation for 3D circumferential milling (CUT3DC, CUT3DCD, ISD)

The 3D tool radius compensation (3D TRC) for the 3D circumferential milling without taking limitation surfaces into account is selected with the modally effective G command CUT3DC or CUT3DCD.

The actual activation is performed with G41 or G42 The tool radius compensation is deactivated with G40.

Syntax

```
G41/G42 ORIC/ORID ISD=... CUT3DC/CUT3DCD CDOF2 X... Y... Z...
...
G40 X... Y... Z...
```

Meaning

CUT3DC:	3D TRC for circumferential milling (only when 5-axis transformation is active)	
CUT3DCD:	3D TRC in relation to a differential tool for circumferential milling (only when 5-axis transformation is active) The radius difference is specified by the tool parameter \$TC_DP15.	
G41/G42 X... Y... Z...:	Activate tool radius compensation	
	G41:	Tool radius compensation left of the contour
	G42:	Tool radius compensation right of the contour
	Note: The activation must be performed in a linear block (G0/G1).	
CDOF2:	Deactivate collision detection for 3D circumferential milling	
ORIC/ORID:	The behavior for orientation changes at outside corners is specified via the G commands ORIC and ORID.	
	ORIC:	Orientation changes at outside corners are superimposed on the circle block to be inserted.
	ORID:	Orientation changes at outside corners are executed before the circle block to be inserted.
ISD=<Value>:	With the ISD address, the insertion depth of the tool can be changed for circumferential milling and active 3D tool radius compensation.	
	<Value>:	Length of the insertion depth
G40 X... Y... Z...:	Deactivate tool radius compensation Note: The deactivation must be performed in a linear block (G0/G1) with geometry axis movements.	

Note

The G commands for selecting the 3D TRC are evaluated in the approach block, i.e. typically in the block that contains G41 or G42.

G41 or G42 can also be programmed in blocks without traversing movement in geometry axes relevant for the compensation. In this case, the approach block is the first traversing block following such a block.

A change of the 3D TRC variant with active tool radius compensation is ignored without alarm.

Example

Program code	Comment
	; Definition of tool D1:
\$TC_DP1[1,1]=120	; Type (end mill)
\$TC_DP3[1,1]=20	; Length compensation vector
\$TC_DP6[1,1]=8	; Radius
N10 X0 Y0 Z0 T1 D1 F12000	; Selection of the tool.
N20 TRAORI(1)	; Activation of the transformation.
N30 G42 ORIC ISD=10 CUT3DC G64 X30	; Activation of the 3D circumferential milling, ; continuous orientation changes at outside corners, ; insertion depth: 10 mm
N40 ORIWKS A30 B15	; Orientation change at a corner through specification of axis positions.
N50 Y20 A3=1 C3=1	; Traversing block with orientation change, ; specification of the orientation with direction vector.
N60 X50 Y30	; Traversing block with constant orientation.
N70 Y50 A3=0.5 B3=1 C3=5	; Traversing block with orientation change.
N80 M63	; Block without traversing information.
N90 X0 ISD=20	; Traversing block with change of the insertion depth.
N100 G40 Y0	; Deactivation of the tool radius compensation.
N110 M30	

Further information

Path and orientation

The type of circumferential milling used here is implemented by defining a path (guide line) and the associated orientation. In this type of machining, the shape of the tool on the path is not relevant. Only the radius at the tool intervention point is decisive.

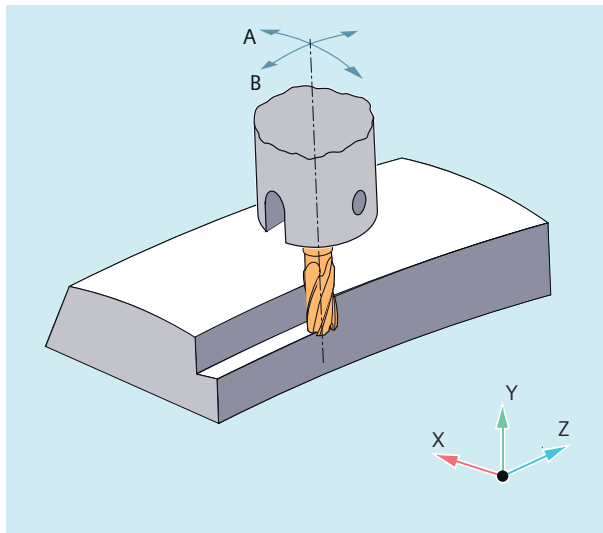


Figure 4-10 Circumferential milling

Approach behavior

The approach behavior is always NORM for the 3D variant of the tool radius compensation.

Behavior at outside corners

The G commands of group 18 (corner behavior, tool offset) are evaluated at the outside corners for circumferential milling with 3D TRC in the same way as for the conditions for the 2½D TRC:

- G450: Transition circle (tool travels round workpiece corners on a circular path)
In contrast to the solution for the 2½D TRC, the inserted contour element at an outside corner is always a circle with a 0 radius, on which the tool radius compensation acts in the same way as on any other programmed path. It is not possible to insert conics instead of circles. In this case, the DISC address has no significance and is therefore not evaluated.
- G451: Intersection of equidistant paths (tool backs off from the workpiece corner)
The intersection is determined by extending the offset curves of the two participating blocks and defining the intersection of the two blocks at the corner in the plane perpendicular to the tool orientation.

The intersection procedure (G451) is not used when at least one block containing a change to the tool orientation was inserted between the relevant traversing blocks. In this case, a circle is always inserted at the corner.

Behavior for changes in orientation at outside corners

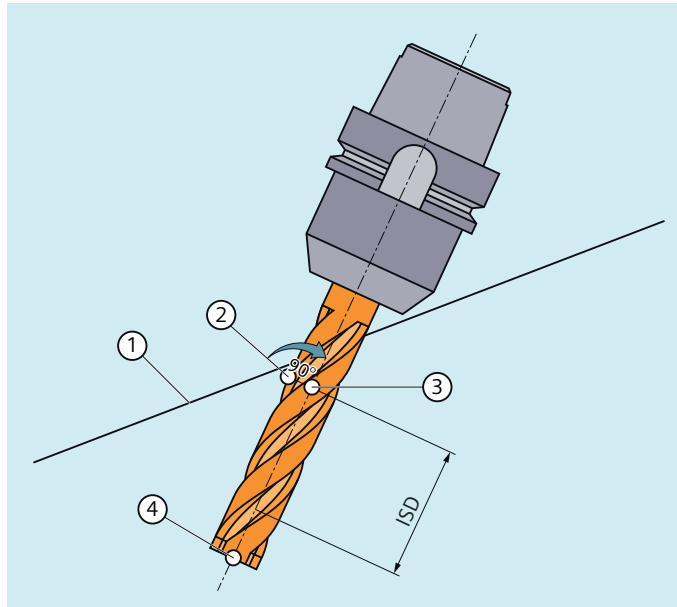
The ORIC and ORID G commands are used to determine whether changes in orientation programmed between two blocks forming the corner are executed before the inserted circle block (ORID) is processed or at the same time (ORIC).

Insertion depth

The insertion depth of the milling tool is the distance of the milling tool reference point from the tip of the tool.

The milling tool reference point is the vertical projection of the milling tool machining point on the programmed path to the longitudinal axis of the tool.

The position of the machining point on the peripheral surface of the tool is set with the insertion depth.



- ① Programmed path
 - ② Milling tool machining point
 - ③ Milling tool reference point
 - ④ Milling tool tip
- ISD Insertion depth (InSersion Depth)

Figure 4-11 Insertion depth

Tool radius compensation referred to a differential tool

3D TRC for circumferential milling referred to a differential tool is activated via the CUT3DCD command. It should be applied if the programmed contour refers to the center-point path of a standard tool, and a tool other than a differential tool is used for machining. When calculating the 3D tool radius compensation, only the wear value of the radius of the active tool (\$TC_DP15) and any programmed tool offsets OFFN and TOFFR/TOFFLR are taken into account. The basic radius (\$TC_DP6) of the active tool is **not** taken into account.

Pocket milling with inclined side walls for circumferential milling with CUT3DC

In this 3D tool radius compensation, a deviation of the mill radius is compensated by infeed toward the surface normals to be machined. The plane, in which the milling tool face is located, remains unchanged if the insertion depth ISD has remained the same. For example, a milling tool with a smaller radius than a standard tool would not reach the pocket base, which is also the limitation surface. For automatic tool infeed, this limitation surface must be known to the control, see Section "3D circumferential milling taking into account a limitation surface (CUT3DCC, CUT3DCCD) (Page 308)".

Advanced Surface / Top Surface**Note**

When applying tool radius compensation CUT3DCD in combination with the "Advanced Surface" or "Top Surface" option (requiring a license), the setting recommendations regarding "Advanced Surface" / "Top Surface" must be observed.

A special test program is provided in the SIOS portal for checking the set data.

→ Test program for Advanced Surface / Top Surface (<https://support.industry.siemens.com/cs/ww/en/view/109738423>)

4.3.2 Selecting 3D tool radius compensation for the 3D face milling (CUT3DF, CUT3DFS, CUT3DFF, CUT3DFD)

The 3D tool radius compensation (3D TRC) for the 3D face milling is selected with the modally effective G command CUT3DF, CUT3DFS, CUT3DFF or CUT3DFD.

The actual activation is performed with G41 or G42

With 3D face milling, the surface normals of the plane to be machined must be defined in order to calculate the tool radius compensation. This must be performed in the block with G41 or G42 via the A4, B4, C4 and A5, B5, C5 addresses.

The tool radius compensation is deactivated with G40.

Syntax

```
G41/G42 ORIC/ORID CUT3DF/CUT3DFS/CUT3DFF/CUT3DFD X... Y... Z... A4=... B4=...
C4=... A5=... B5=... C5=...
...
G40 X... Y... Z...
```

Meaning

CUT3DFS:	3D TRC for face milling with constant orientation. The tool orientation is defined by G17 - G19 and is not influenced by frames.
CUT3DFF:	3D TRC for face milling with constant orientation. The tool orientation is the direction defined by G17 - G19 and, in some cases, rotated by a frame.
CUT3DF:	3D TRC for face milling with orientation change (only with active 5-axes transformation)

CUT3DFD:	3D TRC in relation to a differential tool for face milling with change in orientation (only with active 5-axis transformation) The radius difference is specified by the tool parameter \$TC_DP15. Note: CUT3DFD is only possible in combination with "Smoothing of surface normals in 3D face milling". This is activated by calling the "Top Surface" function (requires a license) via CY-CLE832(...).	
G41/G42 X... Y... Z...:	Activate tool offset The behavior with G41 and with G42 is identical for 3D face milling. Note: The activation must be performed in a linear block (G0/G1).	
A4/5=... B4/5=... C4/5=...:	Definition of the surface normals of the plane to be machined	
	A4=... B4=... C4=...:	Definition at start of block
	A5=... B5=... C5=...:	Definition at end of block
ORIC/ORID:	The behavior for orientation changes at outside corners is specified via the G commands ORIC and ORID.	
	ORIC:	Orientation changes at outside corners are superimposed on the circle block to be inserted.
	ORID:	Orientation changes are performed before the circle block.
G40 X... Y... Z...:	Deactivate tool radius compensation Note: The deactivation must be performed in a linear block (G0/G1) with geometry axis movements.	

Note

G41 or G42 can also be programmed in blocks without traversing movement in geometry axes relevant for the compensation. In this case, the approach block is the first traversing block following such a block.

A change of the 3D TRC variant with active tool radius compensation is ignored without alarm.

Examples**Example 1: 3D face milling with CUT3DF**

Program code	Comment
N10	; Definition of tool D1:
N20 \$TC_DP1[1,1]=121	; Tool type (toroidal miller)
N30 \$TC_DP3[1,1]=20	; Length compensation
N40 \$TC_DP6[1,1]=5	; Radius
N50 \$TC_DP7[1,1]=3	; Rounding radius
N60	
N70	

4.3 Programming

Program code	Comment
N80 X0 Y0 Z0 A0 B0 C0 G17 T1 D1 F12000	; Selection of the tool.
N90 TRAORI(1)	; Select orientation transformation.
N100 B4=-1 C4=1	; Definition of the plane.
N110 G41 ORID CUT3DF G64 X10 Y0 Z0	; Activate tool offset.
N120 X30	
N130 Y20 A4=1 C4=1	; Outside corner, new plane definition.
N140 B3=1 C3=5	; Change in orientation with ORID.
N150 B3=1 C3=1	; Change in orientation with ORID.
N160 X-10 A5=1 C5=2 ORIC	
N170 A3=-2 C3=1	; Change in orientation with ORIC.
N180 A3=-1 C3=1	; Change in orientation with ORIC.
N190 Y-10 A4=-1 C4=3	; New plane definition.
N200 X-20 Y-20 Z10	; Inside corner with previous block.
N210 X-30 Y10 A4=1 C4=1	; Inside corner, new plane definition.
N220 A3=1 B3=0.5 C3=1.7	; Change in orientation with ORIC.
N230 X-20 Y30 A4=1 B4=-2 C4=3 ORID	
N240 A3 = 0.5 B3=-0.5 C3=1	; Change in orientation.
N250 X0 Y30 C4=1	; Path movement, new plane,
	; orientation with relative programming.
N260 BSPLINE X20 Z15	; Spline begin, relative programming of the orientation
N270 X30 Y25 Z18	; remains active during spline.
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, orientation with axis programming.
N320 G1 X0 A3=-0.123 B3=0.456 C3=2.789 B4=-1 C4=5 B5=-1 C5=2	; Path movement, orientation, non-constant plane.
N330 X-20 G40	; Deactivation of the tool radius compensation.
N340 M30	

Example 2: NC program (section) generated from a CAD system with CUT3DFD

Program code	Comment
N01 G710	
N03 T="12"	
N06 S5305 M03	
N07 G642	
	; Approaching the starting position in the MCS taking the tool length into account.
G00 G90 X-250.62787 Y-38.37944 A=DC(253.12719) B-12.49543	
G00 G90 Z251.80052	
	; End of positioning in the MCS.
	;

Program code	Comment
TRAORI (1)	; Select orientation transformation.
G500	
D1	
CYCLE832(0.01, _TOP_SURFACE_SMOOTH_ON + _ORI_FINISH, 1)	; Call CYCLE832 with: ; Contour tolerance = 0.01 mm, ; Processing type: Top Surface with smoothing, ; Finishing with input of an orientation tolerance, ; Orientation tolerance = 1 degree
CUT3DFD	
N08 G90 G94	
N09 G00 X-269.21195 Y128.32027 Z1.18577 A3=-.216361688 B3=.934284397 C3=-.283373051	; The blocks N09 to N10 are the fast portion of the approach movement to the workpiece with constant orientation.
N10 G00 X-251.90301 Y53.57752 Z23.85561	
N11 G01 X-247.57578 Y34.89183 Z29.52308 F50000.00000	; In the blocks N11 to N21, the slow portion of the approach movement is realized. The tool is already close to the workpiece; furthermore, the mold making settings (e.g. COMPSURF) from the CYCLE832 are now active (due to active G01). The path of this so-called transient phase for the mold-making behavior should be about 1000 times the contour tolerance (10 mm in this example).
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	; From N22, the surface normal is completely defined over the entire block for the first time (i.e. surface normal at the end of the previous block N21 is present, thus the surface normal at the beginning of the block N22 and the surface normal at the end of the block N22). Thus the prerequisite for switching on the tool offset with G41/G42 is fulfilled.

4.3 Programming

Program code	Comment
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	
...	

Note

For 3D face milling with CUT3DFD, the definition of the surface normal is required for activating the tool offset with G41/G42. Programming of G41/G42 without definition of the surface normals leads to the output of an alarm.

Further information**3D face milling**

For this type of 3D milling, you require the line-by-line description of the 3D paths on the workpiece surface. The tool shape and dimensions are taken into account in the calculations, which are normally performed in CAM. The postprocessor writes to the part program - in addition to NC blocks - the tool orientations (for active 5-axis transformation) and the G command for the required 3D tool offset. This means that the machine operator has the possibility of using tools that are slightly smaller - deviating from the tool used to calculate the NC paths.

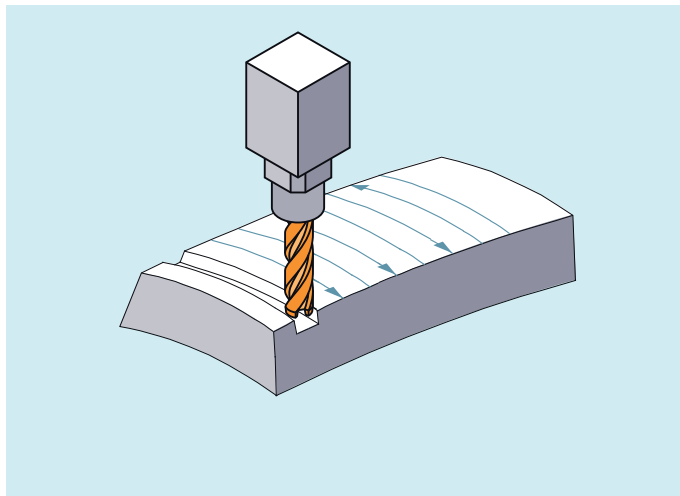


Figure 4-12 Face milling

Approach behavior

The approach behavior is always NORM for the 3D variant of the tool radius compensation.

Behavior at outside corners

Outside corners are treated as circles with radius 0 for face milling, whereby the circle plane extends from the end tangent of the first block to the start tangent of the second block. In this way, the orientation can be changed during block transition. A circle is therefore always inserted as contour element at an outside corner. The intersection procedure is not available with face milling.

Behavior for changes in orientation at outside corners

The ORIC and ORID G commands are used to determine whether changes in orientation programmed between two blocks forming the corner are executed before the inserted circle block (ORID) is processed or at the same time (ORIC).

Tool radius compensation referred to a differential tool

3D tool radius compensation referring to a differential tool is selected using the CUT3DFD command. It should be applied if the programmed contour refers to the center-point path of a standard tool, and a tool other than a differential tool is used for machining. When calculating the 3D tool radius compensation, only the wear value of the radius of the active tool (\$TC_DP15) and any programmed tool offsets OFFN and TOFFR/TOFFLR are taken into account. The basic radius (\$TC_DP6) of the active tool is **not** taken into account.

3D face milling with CUT3DFD is only possible in combination with "Smoothing of surface normals in 3D face milling". This is activated by calling the "Top Surface" function (requires a license) via CYCLE832(...). Activation must take place **before** the tool offset is activated with G41/G42; not directly before tool intervention, but rather one path length before that, which corresponds to approx. 1000 times the contour tolerance (e.g. 1000 x 0.01 mm = 10 mm). Deactivation must be executed in reverse order: First switch off the tool offset with G40, then deactivate with e.g. CUT2D (or similar) after a path length which corresponds with approximately 1000 times the contour tolerance.

In order to be able to use the "Smoothing of surface normals in 3D face milling", the "Interpolation of surface normals via polynomials" function must also be enabled:

MD28291 \$MC_MM_SMOOTH_SURFACE_NORMALS = TRUE

Note

For 3D face milling with CUT3DFD in combination with "Top Surface", the setting recommendations regarding "Top Surface" must be observed.

A special test program is provided in the SIOS portal for checking the set data.

→ Test program for Top Surface (<https://support.industry.siemens.com/cs/ww/en/view/109738423>)

4.3.3 3D circumferential milling taking into account a limitation surface (CUT3DCC, CUT3DCCD)

In 3D circumferential milling with a continuous or constant change in tool orientation, the tool center-point path is frequently programmed for a defined standard tool. Because in practice suitable standard tools are often not available, a tool that does not deviate too much from a standard tool ($\leq 5\%$) can be used.

CUT3DCCD takes account of a limitation surface for a real differential tool that the programmed standard tool would define. The NC program defines the center-point path of a standard tool.

CUT3DCC with the use of cylindrical tools takes account of a limitation surface that the programmed standard tool would have reached. The NC program defines the contour on the machining surface.

The surface normal vector of the limitation surface is specified with A4, B4, C4 and A5, B5, C5 for 3D face milling.

Syntax

```
G41/G42 CUT3DCCD/CUT3DCC CDOF2 X... Y... Z... A4=... B4=... C4=... A5=... B5=...
C5=...
...
G40 X... Y... Z...
```

Meaning

CUT3DCCD:	3D TRC for circumferential milling taking into account a limitation surface with a differential tool on the tool center-point path: Infeed to the limitation surface	
CUT3DCC:	3D TRC for circumferential milling taking into account a limitation surface with 3D radius compensation: Contour on the machining surface	
G41/G42 X... Y... Z...:	Activate tool radius compensation	
	G41:	Tool radius compensation left of the contour
	G42:	Tool radius compensation right of the contour
	Note: The activation must be performed in a linear block (G0/G1).	
CDOF2:	Deactivate collision detection for 3D circumferential milling	
A4/5=... B4/5=... C4/5=...:	Definition of the surface normals of the limitation surface	
	A4=... B4=... C4=...:	Definition at start of block
	A5=... B5=... C5=...:	Definition at end of block
G40 X... Y... Z...:	Deactivate tool radius compensation	
	Note: The deactivation must be performed in a linear block (G0/G1) with geometry axis movements.	

Note

The G commands for selecting the 3D TRC are evaluated in the approach block, i.e. typically in the block that contains G41 or G42.

G41 or G42 can also be programmed in blocks without traversing movement in geometry axes relevant for the compensation. In this case, the approach block is the first traversing block following such a block.

A change of the 3D TRC variant with active tool radius compensation is ignored without alarm.

Example

Program code	Comment
N10 \$TC_DP1[1,1]=120	; Cylindrical milling tool
N20 \$TC_DP6[1,1]=10	
N30 \$TC_DP15[1,1]=-3	
...	
; Processing with cylindrical milling tool and CUT3DCCD	
N110 TRAORI	; Activation of the transformation.
N120 A4=0 B4=0 C4=1	; Definition of the surface normal of the limitation surface at the start of the block.
N130 X0 Y0 Z0 A0 C0 T1 D1 F20000	
N140 X10 Y0 Z0 G41 CUT3DCCD CDOF2 G64	; Activate 3D circumferential milling, taking into account the limitation surface + switching off collision detection.
N150 X20	
N160 X30 A45	; Obtuse angle ==> no infeed
N170 X40 A-45	; Acute angle ==> infeed
N180 X55	
N190 Y10 Z10	; Movement in the tool direction.
N200 Y20	
N210 C45	; Pure change in orientation.
N220 Y30 C90	
N230 A5=-1 B5=0 C5=2 Y40	; Change of the surface.
N240 Y50 G40	; Deactivation of the tool radius compensation.
...	

Further information**Tool type**

The tool type (tool parameter \$TC_DP1) is evaluated. Only milling tools with cylindrical shank (cylinder or end mill, toroidal miller and, in the limit case, cylindrical die mill) are permitted.

This corresponds to the tool types 1 - 399, with the exception of the numbers 111 and 155 to 157.

Standard tools with corner rounding

Corner rounding with a standard tool is defined by the tool parameter \$TC_DP7. Tool parameter \$TC_DP16 describes the deviation of the corner rounding of the real tool compared with the standard tool.

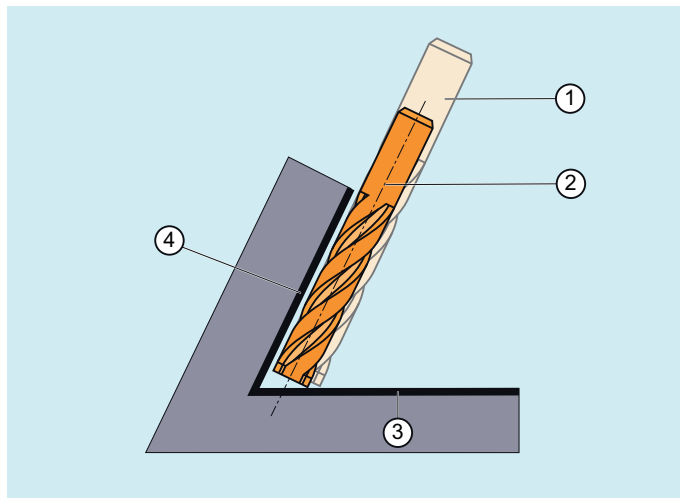
Example: Toroidal miller with reduced radius compared to the standard tool

Tool type	Shaft radius (R)	Corner radius (r)
Standard tool with corner rounding	$R = \$TC_DP6$	$r = \$TC_DP7$
Real tool with corner rounding Tool types 121 and 131 toroidal miller (end mill with corner rounding)	$R' = \$TC_DP6 + \$TC_DP15 + OFFN$	$r' = \$TC_DP7 + \$TC_DP16$

In this example, both \$TC_DP15 + OFFN and \$TC_DP16 are negative.

3D TRC with CUT3DCCD: Tool center-point path with infeed up to the limitation surface

If a tool with a smaller radius than the appropriate standard tool is used, machining is continued using a milling tool, which is fed in in the longitudinal direction until it reaches the bottom (base) of the pocket. The tool removes as much material from the corner formed by the machining surface and limitation surface. This involves a machining type combining circumferential and face milling. Analogous to a tool with reduced radius, for a tool with increased radius, the infeed is in the opposite direction.



- ① Standard tool
- ② Tool with smaller radius infeed up to the limitation surface
- ③ Limitation surface
- ④ Machining surface

Contrary to all other tool offsets of G group 22, tool parameter \$TC_DP6 specified for CUT3DCCD has no relevance for the tool radius and does not influence the resulting compensation. The compensation offset results from the sum of the wear value of the

tool radius (tool parameter \$TC_DP15) and a tool offset OFFN programmed to calculate the perpendicular offset to the limitation surface.

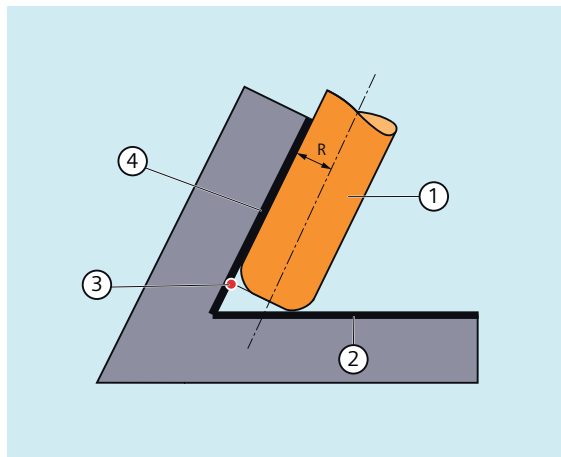
The generated part program does not specify whether the surface to be machined is to the right or left of the path. It is therefore assumed that the radius is a positive value and the wear value of the original tool is a negative value. A negative wear value always describes a tool with a reduced diameter.

Using cylindrical tools

When cylindrical tools are used, infeed is only necessary if the machining surface and the surface of limitation form an acute angle (less than 90 degrees). If a toroidal miller (end mill with corner rounding) is used, tool infeed in the longitudinal direction is required for both acute and obtuse angles.

3D TRC with CUT3DCC: Contour on the machining surface

If CUT3DCC is active with a toroidal miller, the programmed path refers to a fictitious cylindrical milling tool having the same diameter. The resulting path reference point is shown in the following diagram for a toroidal miller.



- ① Toroidal miller
- ② Limitation surface
- ③ Path reference point
- ④ Machining surface
- R Shaft radius (tool radius)

The angle between the machining and limitation surfaces may change from an acute to an obtuse angle and vice versa even within the same block.

The tool actually being used may either be larger or smaller than the standard tool. However, the resulting corner radius must not be negative and the sign of the resulting tool radius must be kept.

4.4 Boundary conditions

For CUT3DCC, the NC part program refers to the contour on the machining surface. As for conventional tool radius compensation, the total tool radius is used that comprises the following components:

- Tool radius (tool parameter \$TC_DP6)
- Wear value (tool parameter \$TC_DP15)
- A tool offset OFFN programmed to calculate the perpendicular offset to the limitation surface

The position of the limitation surface is defined from the following difference:

Dimensions of the standard tool - tool radius (tool parameter \$TC_DP6)

Advanced Surface / Top Surface

Note

When applying tool radius compensation CUT3DCC / CUT3DCCD in combination with the "Advanced Surface" or "Top Surface" function (requiring a license), the setting recommendations regarding "Advanced Surface" / "Top Surface" must be observed. Special test programs are provided in the SIOS portal for checking the set data.

→ Test program for Advanced Surface / Top Surface (<https://support.industry.siemens.com/cs/ww/en/view/109738423>)

4.4 Boundary conditions

Boundary conditions for face milling

Functions with high-order geometry information

Functions with high-order geometry information (e.g. polynomials or spline interpolation) cannot be used.

Top Surface

3D face milling with CUT3DFD is only possible in conjunction with the "Top Surface" function, which requires a license.

Bottleneck detection

A "bottleneck detection" is not possible, and therefore the user is solely responsible.

Traversing movements in the orientation direction of the tool

For traversing motion in the orientation direction of the tool, an alarm is displayed and machining stopped.

4.5 Examples

4.5.1 Example 1: Orientation change at outside corner during 3D circumferential milling

Orientation change at outside corner with ORIC active

Program code	Comment
N10 A0 B0 X0 Y0 Z0 F5000	
N20 T1 D1	; Selection of the tool (radius=5).
N30 TRAORI(1)	; Activation of transformation.
N40 CUT3DC	; Selection of 3D TRC for circumferential milling.
N50 ORIC	; Orientation changes at outside corners are superimposed on the circle block to be inserted.
N60 G42 X10 Y10	; Activation of TRC.
N70 X60	
N80 A3=1 B3=0 C3=1	; change in orientation of external corner generated from N70 and N90
N90 Y60	
N100 X10	
N110 G40 X0 Y0	; Deactivation of TRC.
N120 M30	

The circular motion and change in orientation are executed in parallel in block N80 (ORIC active):

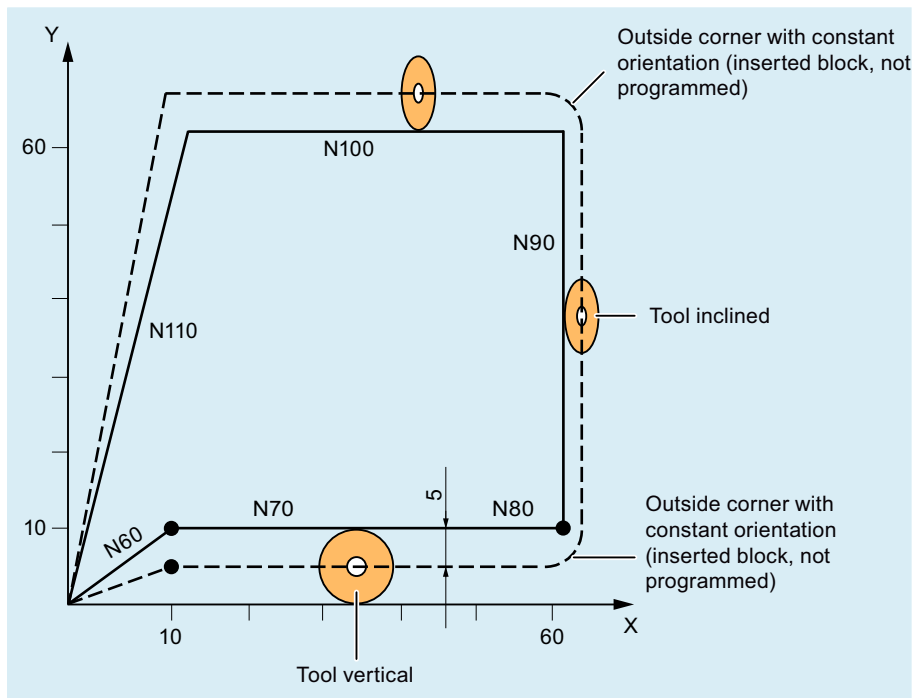


Figure 4-13 ORIC: Change in orientation and path movement in parallel

Special case:

Intermediate blocks without traversing and orientation motions are executed at the programmed positions, e.g. auxiliary functions.

Example:

Program code	Comment
N70 X60	
N75 M20	; auxiliary function call
N80 A3=1 B3=0 C3=1	; change in orientation of external corner generated from N70 and N90
N90 Y60	

Blocks N75 and N80 are executed after N70. The circle block is then executed with the current orientation.

Orientation change at outside corner with ORID active

Program code	Comment
N10 A0 B0 X0 Y0 Z0 F5000	
N20 T1 D1	; Selection of the tool (radius=5).
N30 TRAORI(1)	; Activation of transformation.
N40 CUT3DC	; Selection of 3D TRC for circumferential milling.
N50 ORID	; Orientation changes at outside corners are performed on the circle block to be inserted.
N60 G42 X10 Y10	; Activation of TRC.

Program code	Comment
N70 X60	
N80 A3=1 B3=0 C3=1	; change in orientation of external corner generated from N70 and N90
N90 Y60	
N100 X10	
N110 G40 X0 Y0	; Deactivation of TRC.
N120 M30	

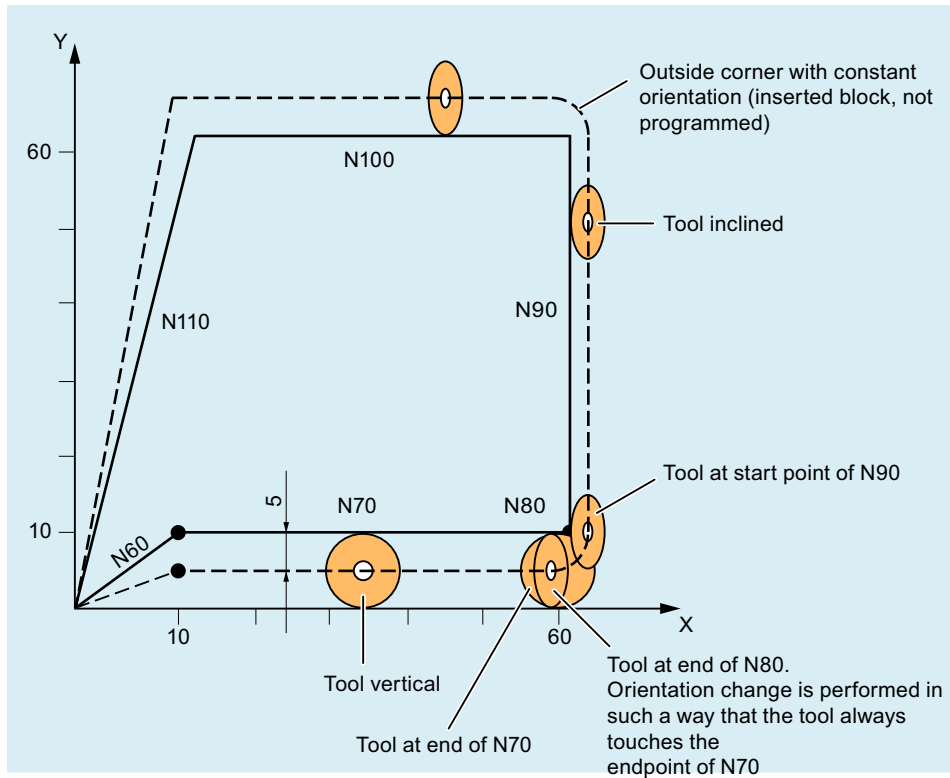


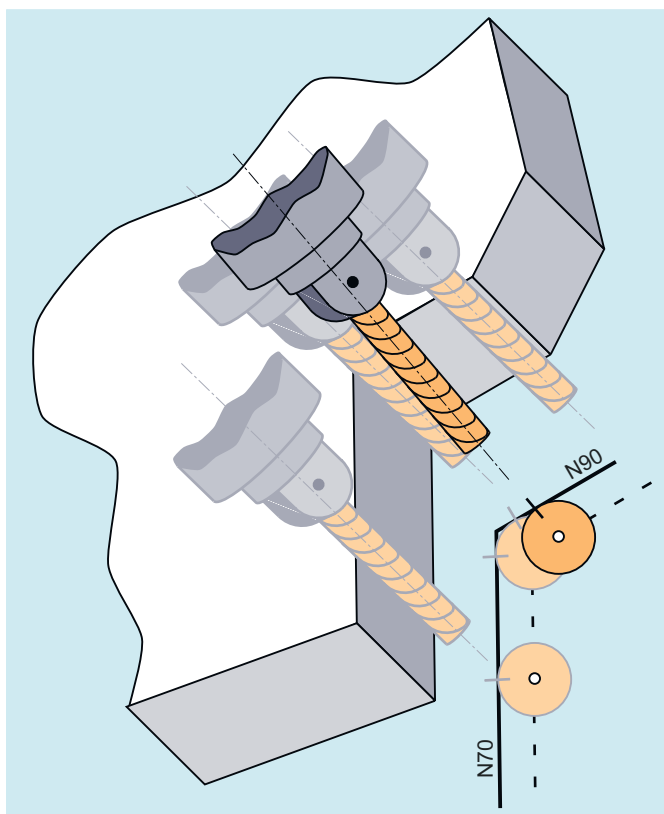
Figure 4-14 ORID: Change in orientation and path movement consecutively

4.5.2

Example 2: Orientation change at inside corner during 3D circumferential milling

Program code	Comment
N10 A0 B0 X0 Y0 Z0 F5000	
N20 T1 D1	; Selection of the tool (radius=5).
N30 TRAORI(1)	; Activation of transformation.
N40 CUT3DC	; Selection of 3D TRC for circumferential milling.
N50 ORID	
N60 G42 X10 Y10 G451	; Activation of TRC.

Program code	Comment
N70 Y60	
N80 A3=1 B3=0 C3=1	; change in orientation of inside corner generated from N70 and N90.
N90 X60 Y90	
N100 G40 X... Y...	
...	
N190 CDOF	
N200 M30	



Grinding-specific tool offset and tool monitoring

5.1 Grinding-specific tool data

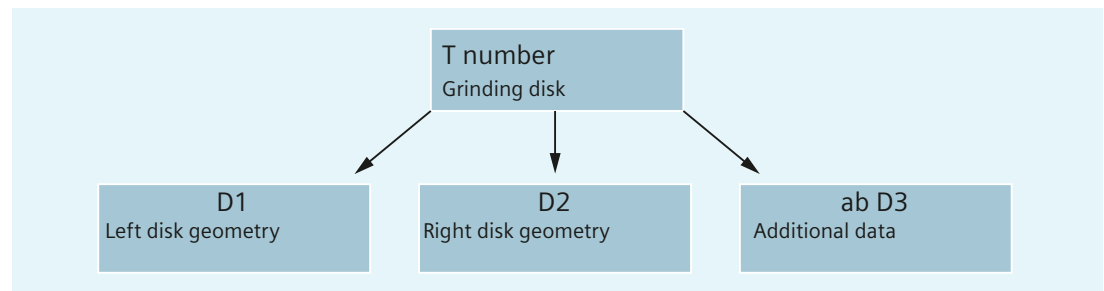
5.1.1 Structure of tool data

Grinding tools (tool type: 400 to 499) normally have specific tool and dresser data in addition to cutting edge data.

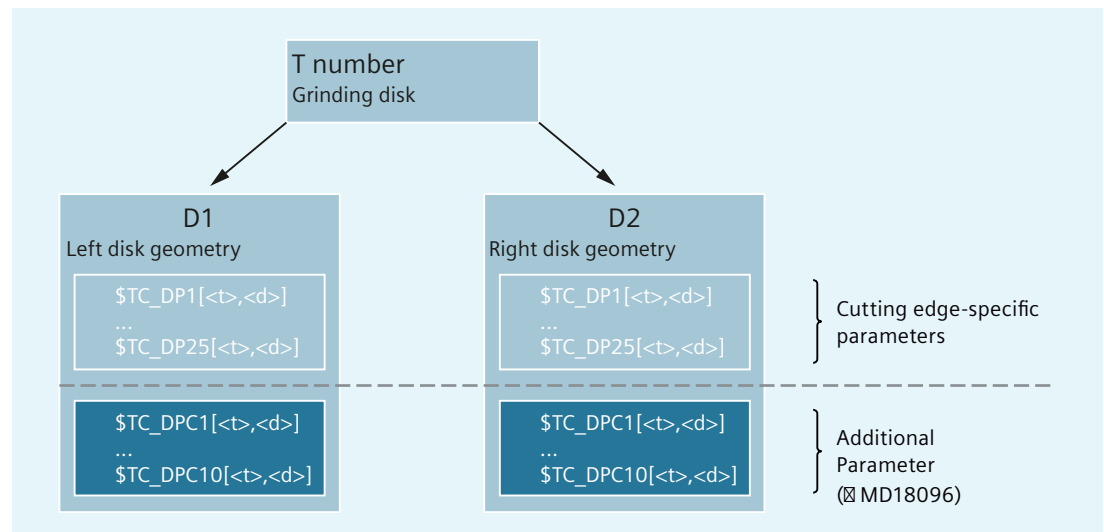
The grinding wheel-specific data for the left and right wheel geometry can be stored under a T number in the tool cutting edges D1 and D2.

If data is needed for the dresser geometry, it can be stored as follows:

- In the tool cutting edges from D3 of a T number



- In additional cutting edge-specific parameters (see MD18096 \$MN_MM_NUM_CC_TOA_PARAM)

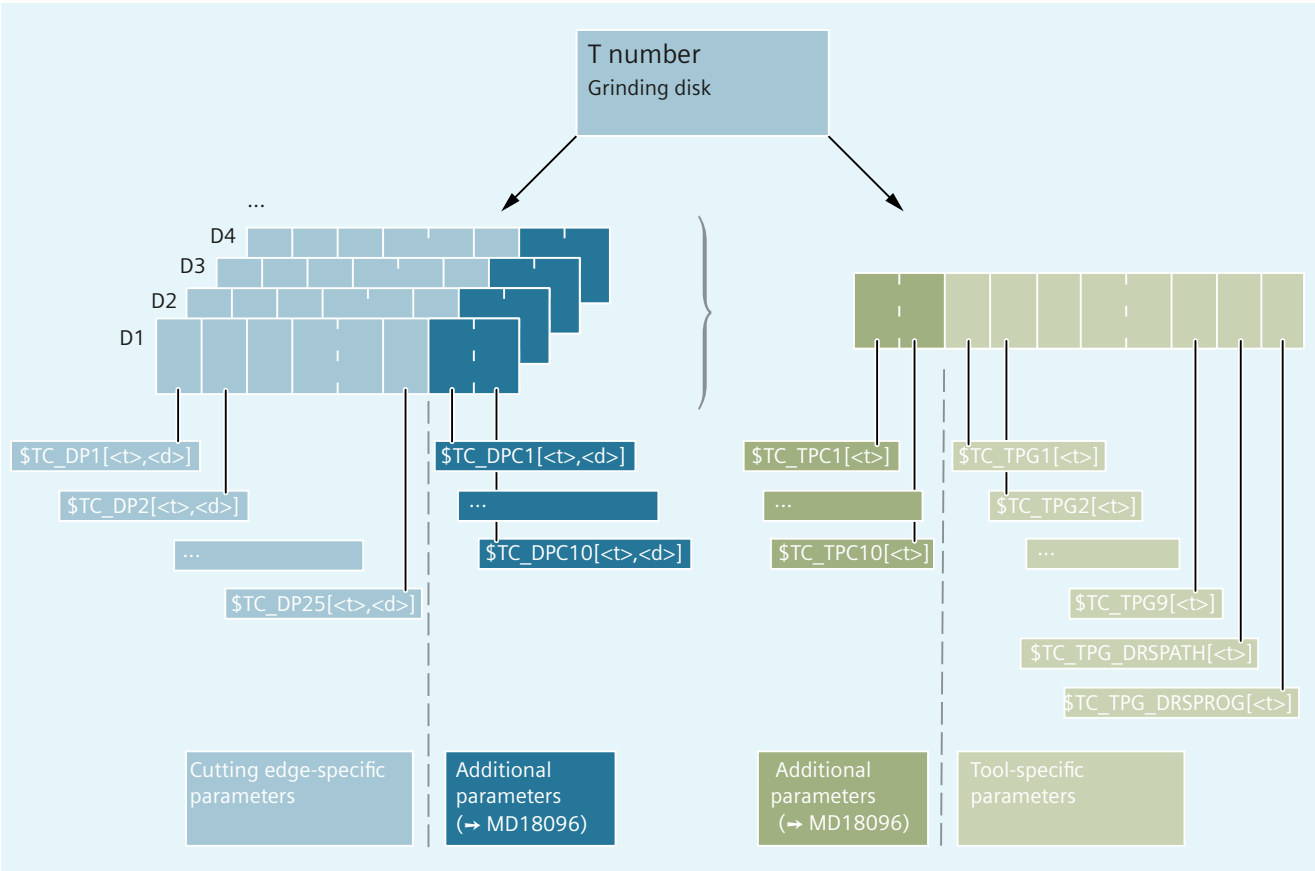


<t>: T number

<d>: D number

5.1 Grinding-specific tool data

The following figure provides an overview of the (possible) storage structure of the tool data for grinding tools:



<t>: T number
<d>: D number

Figure 5-1 Structure of the tool data for grinding tools

5.1.2 Cutting edge-specific parameters

5.1.2.1 List of cutting edge-specific parameters

The cutting edge-specific tool parameters for grinding tools have the same meaning as those for turning and milling tools.

Tool parameter	Meaning	Comment
\$TC_DP1	Tool type	
\$TC_DP2	Cutting edge position	For turning and grinding tools only
Geometry - tool length compensation		

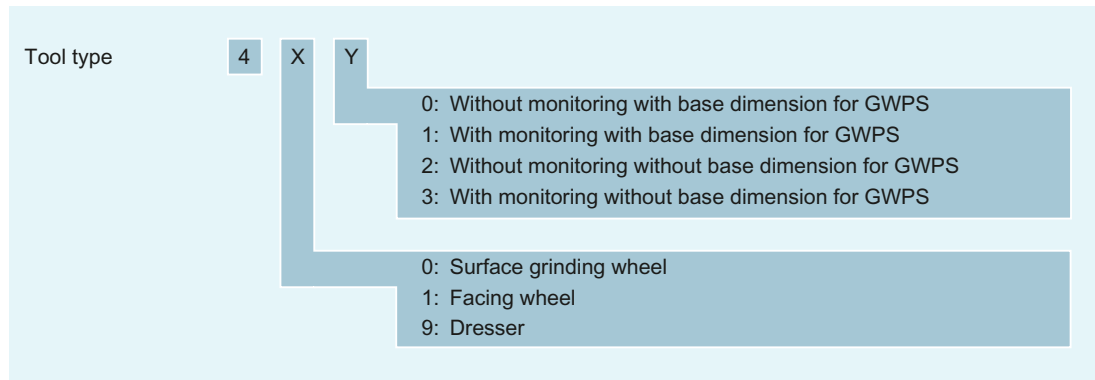
Tool parameter	Meaning	Comment	
\$TC_DP3	Length 1		
\$TC_DP4	Length 2		
\$TC_DP5	Length 3		
Geometry - tool radius compensation			
\$TC_DP6	Radius 1		
\$TC_DP7			Reserved
\$TC_DP8			Reserved
\$TC_DP9			Reserved
\$TC_DP10			Reserved
\$TC_DP11			Reserved
Wear - tool length compensation			
\$TC_DP12	Length 1		
\$TC_DP13	Length 2		
\$TC_DP14	Length 3		
Wear - tool radius compensation			
\$TC_DP15	Radius 1		
\$TC_DP16			Reserved
\$TC_DP17			Reserved
\$TC_DP18			Reserved
\$TC_DP19			Reserved
\$TC_DP20			Reserved
Base dimension/adaptor dimension - tool length compensation			
\$TC_DP21	Basic length 1		
\$TC_DP22	Basic length 2		
\$TC_DP23	Basic length 3		
Technology			
\$TC_DP24	Undercut angle	For turning tools only	
\$TC_DP25			Reserved

Note

The cutting edge data for D1 and D2 of a selected grinding tool can be chained, i.e. if a parameter in D1 or D2 is modified, then the same parameter in D1 or D2 is automatically overwritten with the new value (see \$TC_TPG2 (Page 322)).

5.1.2.2 \$TC_DP1

The parameter \$TC_DP1 contains the 3-digit number of the grinding tool type:



Number	Grinding tool type
400	Surface grinding wheel
401	Surface grinding wheel with monitoring with tool base dimension for GWPS
402	Surface grinding wheel without monitoring without tool base dimension for GWPS
403	Surface grinding wheel with monitoring without tool base dimension for GWPS
410	Facing wheel
411	Facing wheel with monitoring with tool base dimension for GWPS
412	Facing grinding wheel without monitoring without tool base dimension for GWPS
413	Facing wheel with monitoring without tool base dimension for GWPS
490	Dresser

5.1.2.3 Definition of additional parameters \$TC_DPC1 ... 10

For user-specific cutting edge data, the additional parameters \$TC_DPC1 to \$TC_DPC10 can be set up independent of the tool type via the following memory-configuring machine data:

MD18096 \$MN_MM_NUM_CC_TOA_PARAM

NOTICE**Loss of data due to reconfiguration**

Reconfiguration of the local static NC memory results in a loss of data on the active and passive file system. We therefore urgently recommend archiving or backing up all relevant data by creating a commissioning archive before activating a modified memory configuration.

5.1.3 Tool-specific parameters

5.1.3.1 List of tool-specific parameters

The tool-specific parameters are automatically set up with every new grinding tool (tool type: 400 to 499).

Note

Tool-specific parameters have the same characteristics as a cutting edge.

This is to be taken into account when the number of cutting edges is specified:

MD18100 \$MN_MM_NUM_CUTTING_EDGES_IN_TOA

When all the cutting edges of a tool are deleted, the associated tool-specific parameters are deleted at the same time.

Parameters

The parameters are assigned as follows:

Parameter	Meaning	Data type
\$TC_TPG1 (Page 322)	Spindle number	INT
\$TC_TPG2 (Page 322)	Chaining rule	INT
\$TC_TPG3 (Page 323)	Minimum wheel radius	REAL
\$TC_TPG4 (Page 323)	Minimum wheel width	REAL
\$TC_TPG5 (Page 323)	Current wheel width	REAL
\$TC_TPG6 (Page 324)	Maximum speed	REAL
\$TC_TPG7 (Page 324)	Maximum peripheral speed	REAL
\$TC_TPG8 (Page 324)	Angle of inclined wheel	REAL
\$TC_TPG9 (Page 325)	Parameter number for radius calculation	INT
\$TC_TPG_DRSPATH (Page 325)	Directory path to the dressing program	STRING
\$TC_TPG_DRSPROG (Page 325)	Dressing program name	STRING

Note

The monitoring data applies to both the left-hand and the right-hand cutting edge of the grinding wheel.

The tool-specific parameters take effect:

- when switching on the grinding-specific tool monitoring (Page 341)
- when switching on the constant grinding wheel peripheral speed (GWPS) (Page 344)

If a changed date is to take effect, these functions must be switched on again.

The length compensations always specify the distances between the tool carrier reference point and the tool tip in the Cartesian coordinates (must be noted for inclined grinding wheel).

5.1 Grinding-specific tool data

5.1.3.2 \$TC_TPG1

The spindle number is entered in parameter \$TC_TPG1. The grinding wheel circumferential velocity is then monitored for this spindle.

5.1.3.3 \$TC_TPG2

The tool parameters of the two cutting edges D1 and D2 are entered in parameter \$TC_TPG2, their values must always be the same. If, for a cutting edge, the value of one of these parameters is changed, then the corresponding parameter of the other cutting edge is automatically set to the same value.

The tool parameters are coded in parameter \$TC_TPG2:

Tool parameter	Meaning	Coding		
		Bit	Hex	Dec
\$TC_DP1	Tool type	0	0001	1
\$TC_DP2	Length of cutting edge	1	0002	2
Geometry - tool length compensation				
\$TC_DP3	Length 1	2	0004	4
\$TC_DP4	Length 2	3	0008	8
\$TC_DP5	Length 3	4	0010	16
\$TC_DP6	Radius	5	0020	32
\$TC_DP7	Reserved	6	0040	64
\$TC_DP8	Length 4	7	0080	128
\$TC_DP9	Length 5	8	0100	256
\$TC_DP10	Angle 1 etc.	9	0200	512
\$TC_DP11	Reserved	10	0400	1 024
Wear - tool length compensation				
\$TC_DP12	Length 1	11	0800	2 048
\$TC_DP13	Length 2	12	1000	4 096
\$TC_DP14	Length 3	13	2000	8 192
\$TC_DP15	Radius	14	4000	16 384
\$TC_DP16	Reserved	15	8000	32 768
\$TC_DP17	Wear length 4	16	1 0000	65 536
\$TC_DP18	Wear length 5	17	2 0000	131 072
\$TC_DP19	Wear angle 1 etc.	18	4 0000	262 144
\$TC_DP20	Reserved	19	8 0000	524 288
Base dimension/adaptor dimension - tool length compensation				
\$TC_DP21	Basic length 1	20	10 0000	1 048 576
\$TC_DP22	Basic length 2	21	20 0000	2 097 152
\$TC_DP23	Basic length 3	22	40 0000	4 194 304
Technology				
\$TC_DP24	Reserved	23	80 0000	8 388 608
\$TC_DP25	Reserved	24	100 0000	16 777 216

Change becomes effective

If parameter \$TC_TPG2 becomes effective at a later point in time, the values of the tool parameters of the cutting edges are not automatically aligned immediately. The alignment is only made the next time that a tool parameter of both cutting edges is changed.

Example

For a grinding tool (T1), the following parameters should be kept the same for cutting edges D1 and D2:

Tool parameter	Meaning	Coding: Hex	Coding: Decimal
\$TC_DP1	Tool type	0001	1
Geometry - tool length compensation			
\$TC_DP3	Length 1	0004	4
\$TC_DP4	Length 2	0008	8
\$TC_DP5	Length 3	0010	16
Wear - tool length compensation			
\$TC_DP12	Length 1	0800	2 048
\$TC_DP13	Length 2	1000	4 096
\$TC_DP14	Length 3	2000	8 192
Base dimension/adaptor dimension - tool length compensation			
\$TC_DP21	Basic length 1	10 0000	1 048 576
\$TC_DP22	Basic length 2	20 0000	2 097 152
\$TC_DP23	Basic length 3	40 0000	4 194 304

The sum of all parameter codings is entered in parameter \$TC_TPG2:

\$TC_TPG2 =	'H70381D'	'D7354397'
-------------	-----------	------------

5.1.3.4 \$TC_TPG3, \$TC_TPG4

For grinding-specific tool monitoring (Page 341), the **lower** limit values of the grinding wheel are entered in the parameters:

- Parameter \$TC_TPG3: Radius [mm] or [inch]
- Parameter \$TC_TPG4: Width [mm] or [inch]

Note

For inclined grinding wheels, the minimum radius must be entered in Cartesian coordinates.

5.1.3.5 \$TC_TPG5

The actual width of the grinding wheel - e.g. after dressing - is entered in parameter \$TC_TPG5.

5.1.3.6 \$TC_TPG6 and \$TC_TPG7

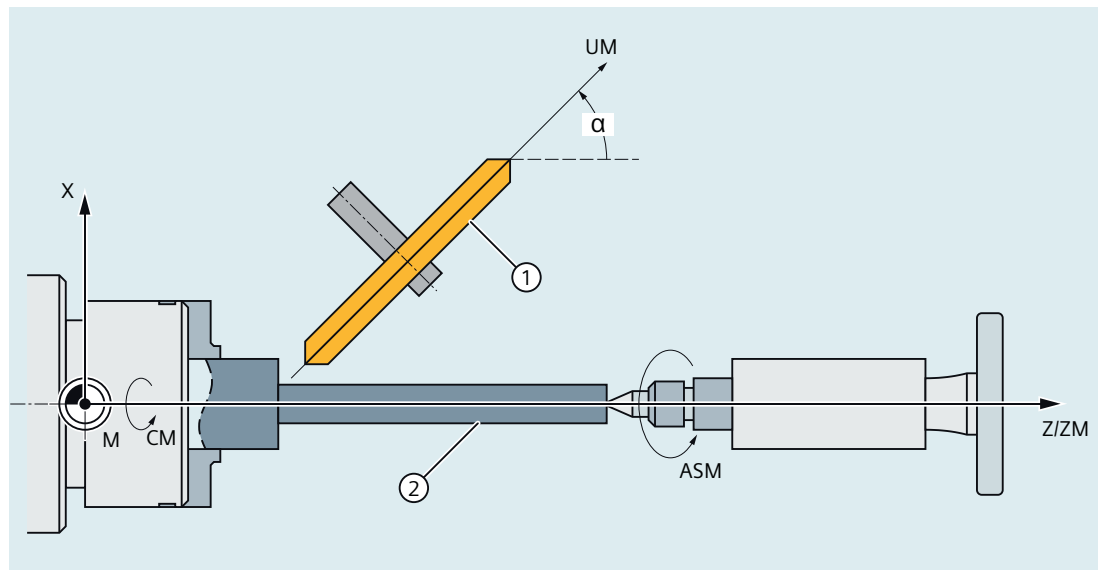
For grinding-specific tool monitoring (Page 341), the **upper** limit values of the grinding wheel are entered in the parameters:

- Parameter \$TC_TPG6: Speed [rpm]
- Parameter \$TC_TPG7: Circumferential/peripheral speed [$\text{m}\cdot\text{s}^{-1}$] or [$\text{inch}\cdot\text{s}^{-1}$]

5.1.3.7 \$TC_TPG8

For function "Constant grinding wheel peripheral speed (GWPS) (Page 344)", for an inclined grinding wheel, the inclination angle α referred to the actual active plane (work plane G17, G18, G19) is entered in parameter \$TC_TPG8.

The permissible range of inclination angle α is: $-90^\circ \leq \alpha < +90^\circ$



- ① Grinding disk
- ② Workpiece
- X Geometry axis
- Z Geometry axis
- ZM Machine axis
- UM Machine axis
- α Angle of inclined axis
- CM Machine axis (spindle)
- ASM Machine axis (tailstock)

Figure 5-2 Inclined grinding wheel

Note

The tool length is not automatically compensated when the angle in parameter \$TC_TPG8 is changed.

Note

For machines with inclined axis, the same angle must be entered for the inclined axis as for the inclined grinding wheel.

5.1.3.8 \$TC_TPG9

Parameter \$TC_TPG9 is used to define to which tool parameters the following functions should refer:

- Grinding-specific tool monitoring (Page 341)
- Constant grinding wheel peripheral speed (GWPS) (Page 344)

Examples:

\$TC_TPG9 = <value>

<Value>	Reference variable
3	Length 1 (geometry + wear + base, depending on tool type)
4	Length 2 (geometry + wear + base, depending on tool type)
5	Length 3 (geometry + wear + base, depending on tool type)
6	Radius

5.1.3.9 \$TC_TPG_DRSPATH and \$TC_TPG_DRSPROG

Parameter \$TC_TPG_DRSPATH (directory path to the dressing program) and \$TC_TPG_DRSPROG (dressing program) allow a dressing program to be assigned to a grinding wheel. In turn, this allows a generally valid dressing procedure to be created, in which the dressing program is executed using an indirect program call.

Example:

Program code	Comment
...	
	; Enter absolute dressing amount in \$TC_DP12:
\$TC_DP12[GETT("GRINDING_WHEEL",0),1]= \$TC_DP12[GETT("GRINDING_WHEEL",0),1] - 0.01	
G500	
CYCLE435("GRINDING_WHEEL",1,"DRESSER",1)	; Calculate dressing tool position
T="DRESSER" D1	; Activate dressing tool.
	; Indirectly call dressing program:
CALL "/"<<\$TC_TPG_DRSPATH[GETT("GRINDING_WHEEL",0)]<<"/"<<\$TC_TPG_DRSPROG[GETT("GRINDING_WHEEL",0)]	
M17	

5.1.3.10 Definition of additional parameters \$TC_TPC1 ... 10

For user-specific tool data, additional parameters \$TC_TPC1 to \$TC_TPC10 are set up independent of the tool type

NOTICE

Loss of data due to reconfiguration

Reconfiguration of the local static NC memory results in a loss of data on the active and passive file system. We therefore urgently recommend archiving or backing up all relevant data by creating a commissioning archive before activating a modified memory configuration.

5.1.3.11 Access to tool-specific parameters

Tool-specific parameters can be read and written from the part program.

Examples:

Program code	Comment
R10=\$TC_TPG5[2]	; read the current width of tool 2 and store in R10
\$TC_TPG6[3]=2000	; write value 2000 to the maximum speed of tool 3.

The following system variable allows the tool-specific parameters for the **current** tool to be accessed:

\$P_ATPG[<m>]

<m> = parameter numbewr (data type: REAL)

Example:

Parameter 3 (\$TPG3[<t>])

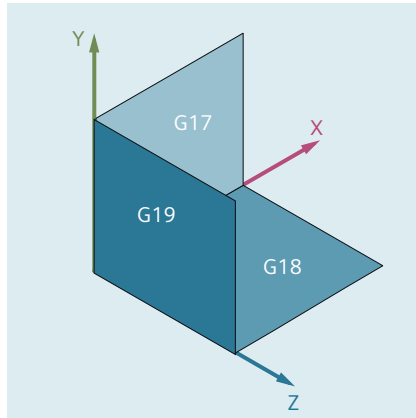
Program code	Comment
\$P_ATPG[3]=R10	; read the minimum wheel radius of the current tool and store in R10.

5.1.4 Planes and axis assignments

Tool length compensations for the geometry axes or radius compensation in the plane are assigned on the basis of the current machining plane.

The following planes and axis assignments are possible (abscissa, ordinate, applicate for 1st, 2nd and 3rd geometry axis):

Command	Plane (abscissa/ordinate)	Axis perpendicular to plane (applicate)
G17	X / Y	Z
G18	Z / X	Y
G19	Y / Z	X



- X Geometry axis 1
- Y Geometry axis 2
- Z Geometry axis 3

5.1.5 Examples

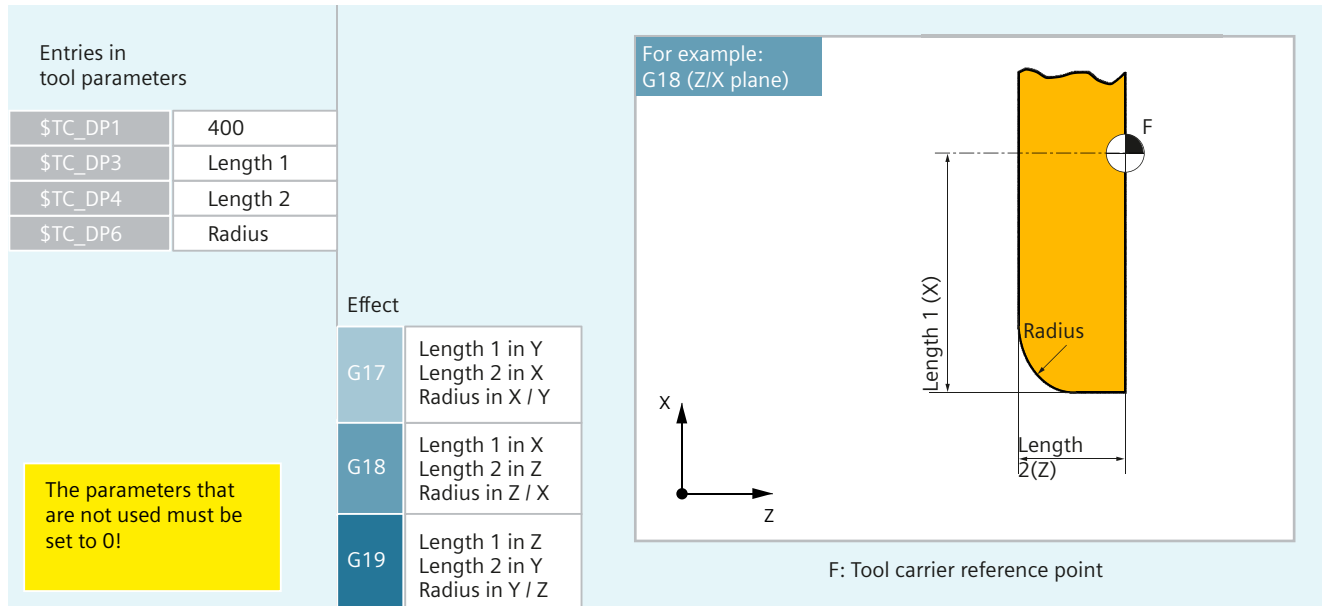


Figure 5-3 Required offset data of a surface grinding wheel

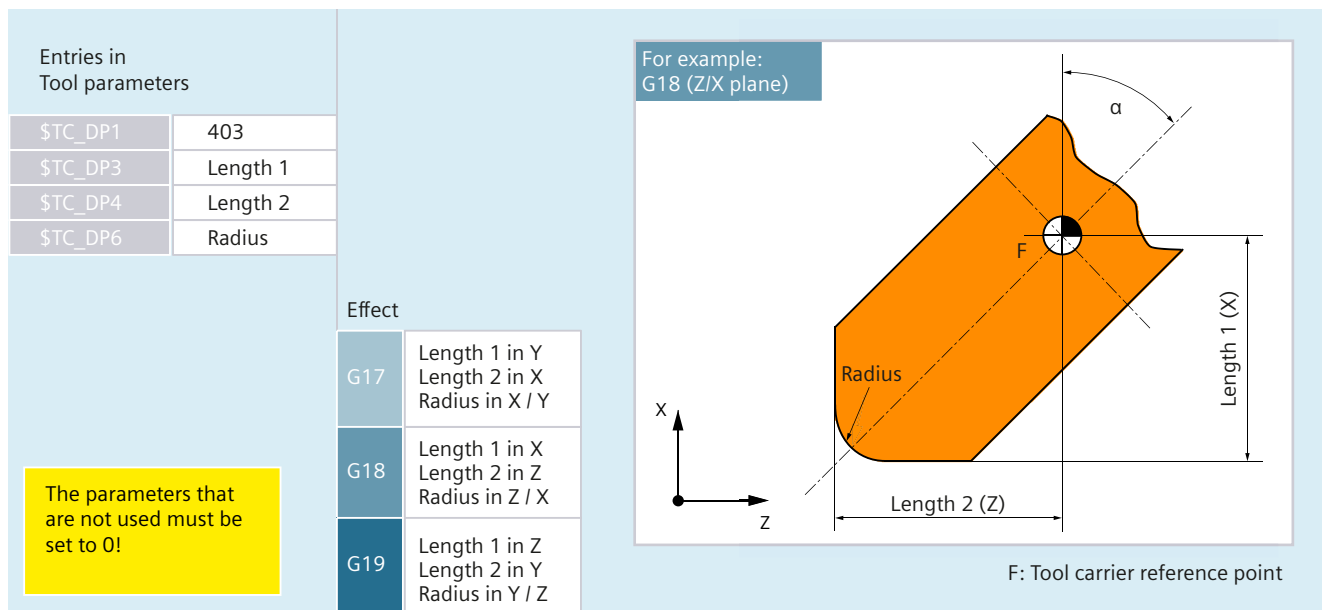


Figure 5-4 Required offset data for inclined wheel with implicit monitoring selection and without base dimension for GWPS

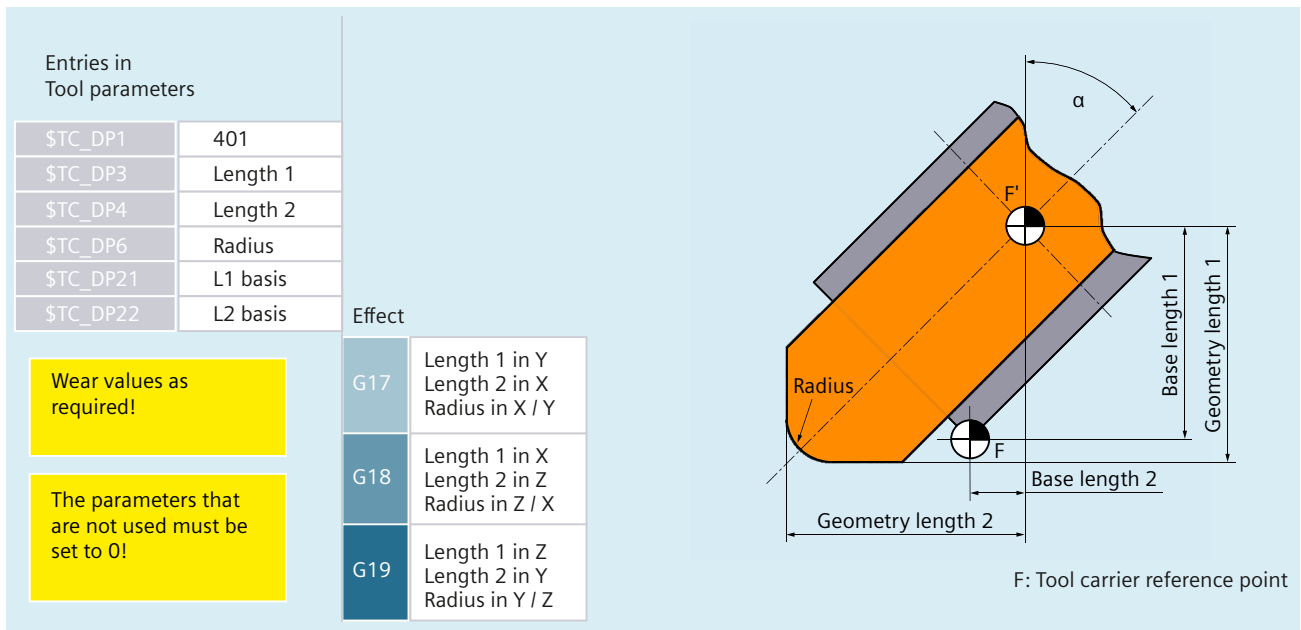


Figure 5-5 Required offset data for inclined wheel with implicit monitoring selection and with base dimension for GWPS

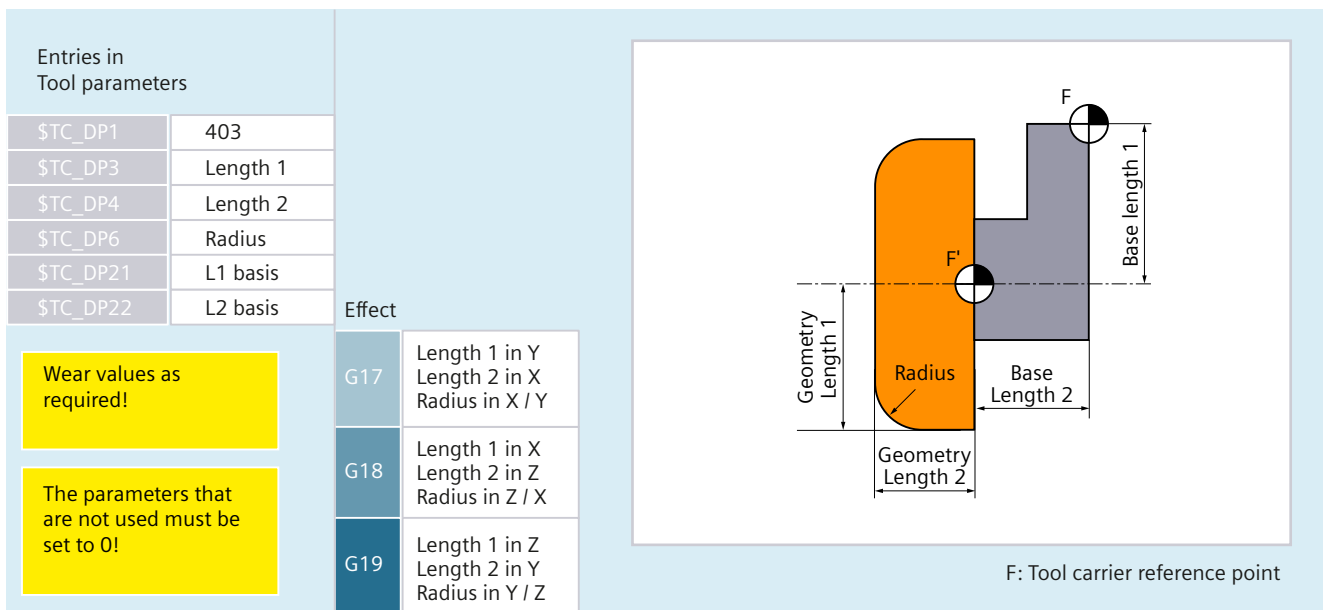


Figure 5-6 Required offset data of a surface grinding wheel without base dimension for GWPS

5.2 Online tool offset

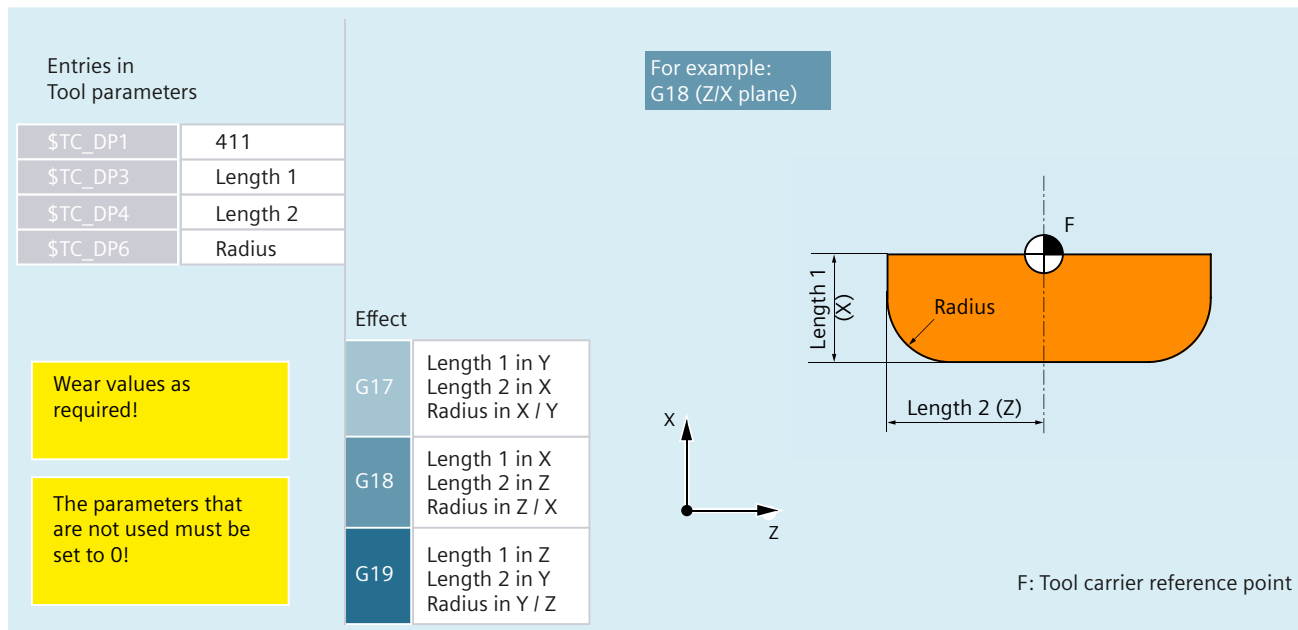


Figure 5-7 Required offset data of a facing wheel with monitoring parameters

5.2 Online tool offset

5.2.1 Function

A grinding operation involves both machining of a workpiece and dressing of the grinding wheel. These processes can take place in the same channel or in separate channels.

To allow machining to continue while the grinding wheel is being dressed, the reduction in the size of the grinding wheel caused by dressing must be transferred to the current tool in the machining channel as a tool offset that is applied immediately. This can be implemented by means of the "Online tool offset" function.

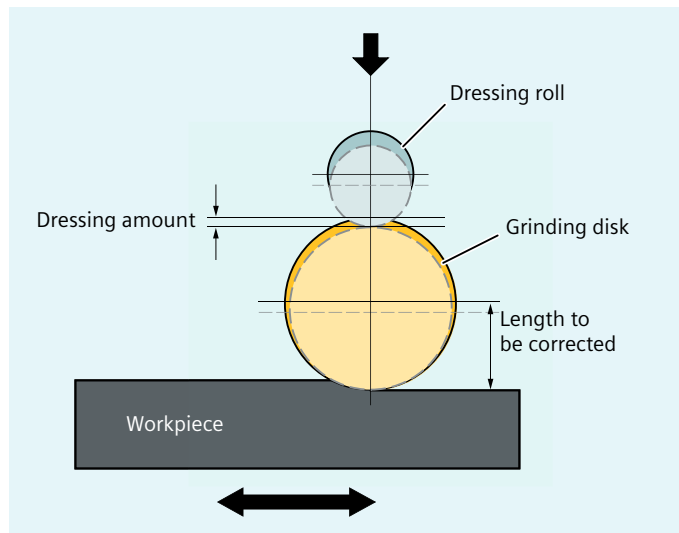


Figure 5-8 Dressing during machining using a dressing roller

The function supports the following methods of writing the tool offset:

- Continuous write, non-modal (PUTFTOCF (Page 334))
With PUTFTOCF, dressing is carried out at the same time as machining.
The tool offset is changed continuously in the machining channel according to a polynomial function of the first, second or third order, which must have been defined previously with FCTDEF (Page 333).
PUTFTOCF is always non-modal; in other words, it takes effect in the subsequent traversing block.
- Continuous write, modal: ID=1 DO FTOC (for further information, see Function Manual Synchronized Actions)
- Discrete write (PUTFTOC (Page 335))
With PUTFTOC, dressing is not carried out at the same time as machining from a parallel channel. The correction value specified with PUTFTOC takes effect immediately in the destination channel.

Activation / deactivation

The online tool offset in the part program is activated/deactivated by means of the G command FTOCON/FTOCOF (Page 336).

Additional properties

- An online tool offset can be activated for every grinding tool (tool type: 400 to 499) in each channel.
- The online tool offset is generally applied as a length compensation. Like geometry and wear data, lengths are assigned to geometry axes on the basis of the current plane as a function of the tool type.
- The grinding spindle monitoring function remains active when an online tool offset is selected.

5.2 Online tool offset

Note

The online tool offset may only be used for grinding tools.

The offset always corrects the wear parameters of the selected length. If the length compensation is identical for several cutting edges, then a chaining specification must be used to ensure that the values for the 2nd cutting edge are automatically corrected as well.

If online tool offset is active in the machining channel, then the wear values for the active tool in this channel may not be changed from the machining program or via operator inputs.

Modifications to the radius wear (P15) are not taken into account until the tool is reselected.

The online tool offset is also applied to the constant grinding wheel peripheral speed (GWPS), i.e. the spindle speed is corrected by the corresponding value.

5.2.2 Commissioning

Number of tool holders that can be defined

Function "Online tool offset" cannot be simultaneously used with the function "Orientable toolholder".

When using the "Online tool offset" function, the maximum number of tool holders that can be defined in the TO area should be set to "0":

MD18075 \$MN_MM_NUM_TOOLHOLDERS = 0

Note

For SINUMERIK 828D, the value of MD18075 can only be set in "Grinding" technology.

Enable online tool radius compensation

Online tool radius compensation (Page 340) must also be additionally enabled using machine data:

MD20254 \$MC_ONLINE_CUTCOM_ENABLE = 1

5.2.3 Programming

5.2.3.1 Defining a polynomial function (FCTDEF)

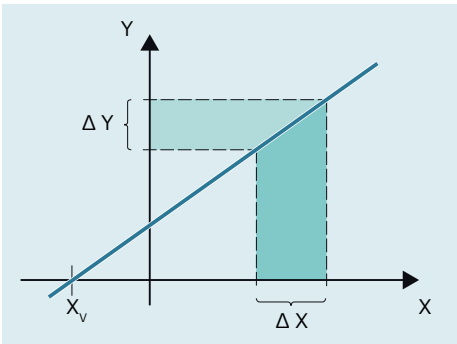
Certain dressing strategies (e.g. dressing roller) are characterized by the fact that the grinding wheel radius is continuously (linearly) reduced as the dressing roller is fed in. This strategy requires a linear function between infeed of the dressing roller and writing the wear value of each length. The linear function is defined using the predefined procedure FCTDEF(...) for up to third order polynomial functions.

Straight line equation

$$y = f(x) = a_0 + a_1 * x_1$$

a_1 : Gradient of the straight line, with $a_1 = \Delta x / \Delta y$

a_0 : Shift of the straight line along the X axis with $a_0 = -a_1 * X_v$



Syntax

FCTDEF (<Func>, <LLimit>, <ULimit>, <a0>, <a1>, <a2>, <a3>)

Meaning

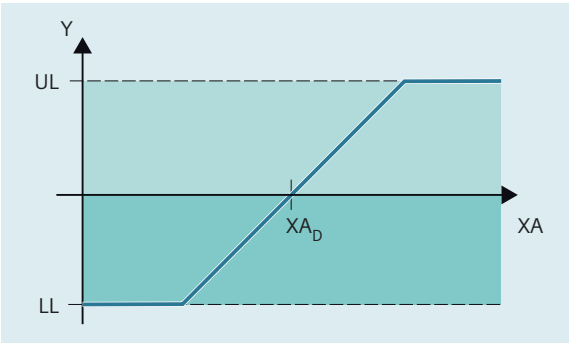
FCTDEF (...):	Defining a polynomial function for PUTFTOCF(...): $y = f(x) = a_0 + a_1 * x + a_2 * x^2 + a_3 * x^3$	
<Func>:	Function number	
	Data type:	INT
	Range of values:	1, 2, 3
<LLimit>:	Lower limit value	
	Data type:	REAL
<ULimit>:	Upper limit value	
	Data type:	REAL
<a0>, <a1>, <a2>, <a3>:	Coefficients of polynomial function	
	Data type:	REAL

Example

Definitions

- Function number: 1
- Lower and upper limit value: -100, 100
- Gradient of the characteristic: $a_1 = 1$
- The operating point should be located at the center of the characteristic. Based on the setpoint position of axis XA in the WCS at the instant that the function is defined in the NC program, the characteristic must be shifted in the negative Y direction: $a_0 = -a_1 * XA_D = -1 * \$AA_IW$
- $a_2 = a_3 = 0$

Characteristic



- UL Upper limit value
- LL Lower limit value
- XA_D Setpoint of axis XA at the time that the function is defined in the NC program

Programming

Program code	Comment
<code>FCTDEF(1,-100,100,-\$AA_IW[XA],1)</code>	; Function definition

5.2.3.2 Write online tool offset continuously (PUTFTOCF)

Using the predefined procedure PUTFTOCF(...), an online tool offset is executed based on a polynomial function previously defined with FCTDEF(...) (Page 333).

Note

The online tool offset can also be realized using a synchronized action.
For further information, see Function Manual Synchronized Actions.

Syntax

`PUTFTOCF(<Func>,<RefVal>,<ToolPar>,<Chan>,<Sp>)`

Meaning

PUTFTOCF (. . .):	Write online tool offset, continuously block-by-block using the polynomial function defined with FCTDEF(...)	
<Func>:	Function number, defined in the function definition with FCTDEF(...)	
	Data type:	INT
	Range of values:	1, 2, 3
<RefVal>:	Reference value, from which the offset is to be derived (e.g. setpoint of an axis).	
	Data type:	VAR REAL
<ToolPar>:	Number of the wear parameter (length 1, 2 or 3) in which the offset value is to be included.	
	Data type:	INT
<Chan>:	Number of the channel in which the online tool offset is to take effect.	
	Note: Only required if the offset is not to take effect in the active channel.	
	Data type:	INT
<Sp>:	Number of the spindle for which the online tool offset is to take effect.	
	Note: Only required if the offset is to be applied to a non-active grinding wheel rather than the active tool that is currently in use.	
	Data type:	INT

5.2.3.3 Write online tool offset, discrete (PUTFTOC)

Function

Using the predefined procedurePUTFTOC(...), an online tool offset is executed based on a fixed offset value.

Syntax

PUTFTOC (<CorrVal>, <ToolPar>, <Chan>, <Sp>)

Meaning

PUTFTOC (. . .):	Write online tool offset	
<CorrVal>:	Offset value, which is added to the wear parameter.	
	Data type:	REAL
<ToolPar>:	Number of the wear parameter (length 1, 2 or 3) in which the offset value is to be included.	
	Data type:	INT
<Chan>:	Number of the channel in which the online tool offset is to take effect.	
	Note: Only required if the offset is not to take effect in the active channel.	
	Data type:	INT

5.2 Online tool offset

<Sp>:	Number of the spindle for which the online tool offset is to take effect.	
	Note: Only required if the offset is to be applied to a non-active grinding wheel rather than the active tool that is currently in use.	
	Data type:	INT

5.2.3.4 Activate/deactivate online tool offset (FTOCON/FTOCOF)

The online tool offset is activated or deactivated using the G commands FTOCON and FTOCOF.

Syntax

```
FTOCON
...
FTOCOF
```

Meaning

FTOCON:	Activate online tool offset The command must be programmed in the channel in which the online tool offset is to be activated.
FTOCOF:	Deactivate online tool offset The command must be programmed in the channel in which the online tool offset is to be deactivated. Note: On FTOCOF, the axis does not move further out for the tool offset. However, the value calculated with PUTFTOC/PUTFTOCF remains in the cutting-specific offset data. To finally deactivate the online tool offset, the tool (T...) must again be selected/deselected after FTOCOF.

5.2.4 Supplementary conditions

DRF offset

The online tool offset is superimposed on the programmed axis motion, allowing for the defined limit values (e.g. velocity).

If a DRF offset and online offset are active simultaneously for an axis, the DRF offset is considered first.

Acceleration margin

The active online tool offset is traversed through at JOG velocity, allowing for the maximum acceleration.

In case of FTOCON the following channel-specific machine data is taken into account:

MD20610 \$MC_ADD_MOVE_ACCEL_RESERVE

An acceleration margin can thus be reserved for the movement which means that the overlaid movement can be executed immediately.

Reference point approach

The valid online offset is deleted on reference point approach with G74.

Tool change

In cases where FTOCON has been active since the last tool or cutting edge change, preprocessing stop with resynchronization is initiated in the control when a tool is changed.

Cutting edge changes can be implemented without preprocessing stop.

Note

Tool changes can be executed in conjunction with the online tool offset through the selection of T numbers.

Tool changes with M6 cannot be executed in conjunction with the online tool offset function.

Machining plane and transformation

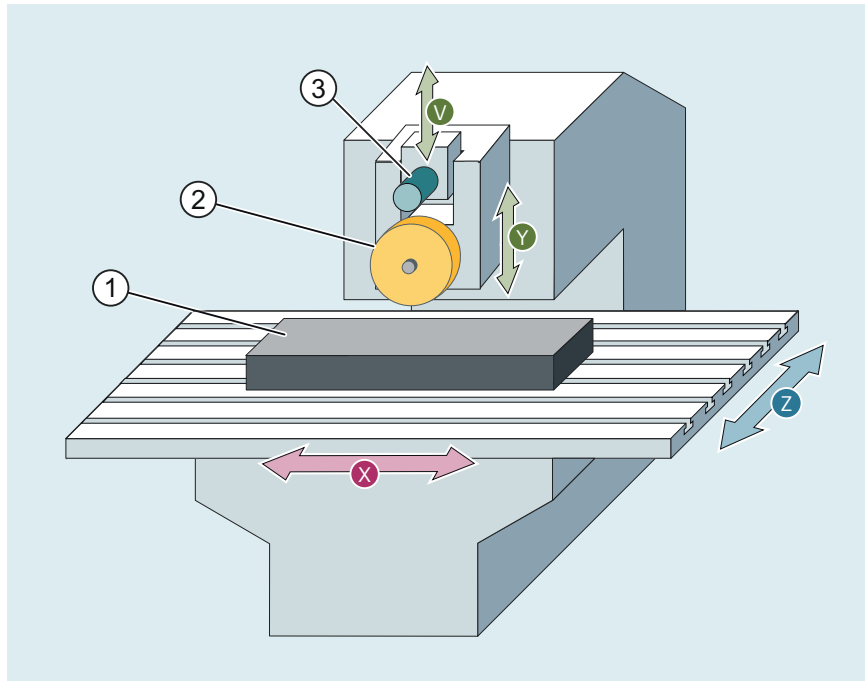
- FTOCON can be used only in conjunction with the "Inclined axis" transformation.
- It is not possible to change transformations or planes (e.g. G17 to G18) when FTOCON is active, except in the FTOCOF state.

Resets and operating mode changes

- When online offset is active, NC-Stop and program end with M2/M30 are delayed until the amount of compensation has been traversed.
- The online tool offset is immediately deselected in response to NC-Reset.
- Online tool offsets can be activated in AUTOMATIC mode and when the program is active.

5.2.5 Examples

5.2.5.1 Example: Write online tool offset continuously



- ① Workpiece
- ② Grinding disk
- ③ Dressing roll
- X: Oscillating axis
- Y: Infeed axis: Grinding disk
- Z: Table axis
- V: Infeed axis: Dressing roll

Figure 5-9 Surface grinding machine

Specifications

- Tool offset
 - Machining plane: Y/Z plane (G19)
 - Tool type: 401 (length 1 acts in Z, length 2 acts in Y)
- Channel 1: Machining channel, axes X, Y, Z
- Channel 2: Dressing channel, axis V

After the grinding operation has started at Y100, the grinding wheel must be dressed by 0.05 (in V direction). The dressing amount must be compensated continuously by means of an online tool offset.

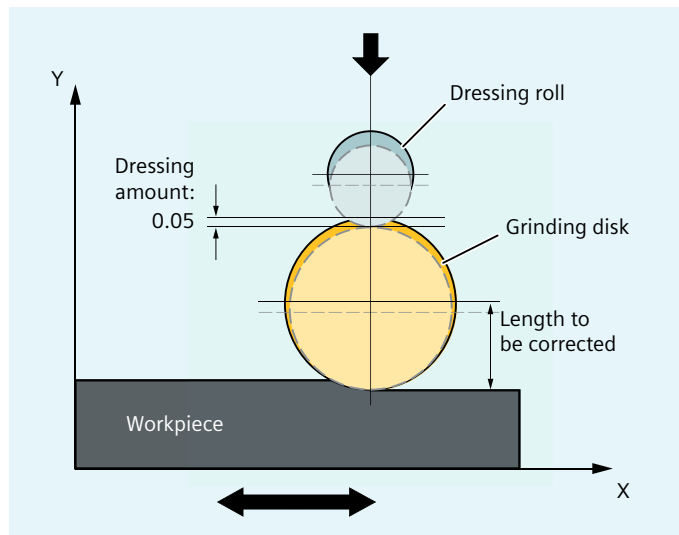


Figure 5-10 Tool offset

Program (section) for channel 1: Machining channel

Program code	Comment
...	
G1 G18 F10 G90	; initial setting
T1 D1	; select actual tool
S100 M3 X100	; switch-on spindle, X axis to initial position
INIT (2, "/_N_MPF_DIR/_N_ABRICHT_MPF", "S")	; select: program "DRESS" in channel 2:
START (2)	; start: Channel 2
X200	; X axis at target position
FTOCON	; activate online offset
...	

Program for channel 2: Dressing channel

The V axis traverses parallel to the Y axis, i.e. length 3 of the tool offset acts in the direction of the Y axis, and must therefore be compensated..

Program code	Comment
PROC ABRICHT	
FCTDEF(1, -1000, 1000, -\$AA_IW[V], 1)	; function definition
PUTFTOCF(1, \$AA_IW[V], 3, 1)	; continuously write online tool offset:
	; - Function number: 1
	; - Reference value: Setpoint position in the WCS of the V axis
	; - Tool parameters: Length 3 of the grinding disk
	; - Channel: 1
	; - Spindle: active grinding disk

5.3 Online tool radius compensation

Program code	Comment
V-0.05 G1 F0.01 G91 M30	; infeed motion of the V axis for dressing

5.3 Online tool radius compensation

Function

When the longitudinal axis of the tool and the contour are perpendicular to each other, the offset can be applied as a length compensation to one of the three geometry axes (online tool length compensation).

If this condition is not fulfilled, then the offset quantity can be entered as a real radius compensation value (online tool radius compensation).

Enable

Online tool radius compensation must be enabled using machine data:

MD20254 \$MC_ONLINE_CUTCOM_ENABLE = 1

Activation / deactivation

An online tool radius compensation is activated and deactivated using G commands FTOCON and FTOCOF (Page 336) (in the same way as an online tool length compensation).

Parameter assignment

Online tool offset is parameterized using PUTFTOCF (Page 334) or PUTFTOC (Page 335).

The parameter <ToolPar> (wear parameter in which the correction value is to be calculated) must be set for the online tool radius compensation as follows:

<ToolPar> = 4

Boundary conditions

- A tool radius compensation, and thus also an online tool radius compensation, can be activated only when the selected tool has a radius other than "0". This means that machining operations cannot be implemented solely with a tool radius compensation.
- The online correction values should be low in comparison to the original radius to prevent the permitted dynamic tolerance range from being exceeded when the offset is overlaid on the axis movement.
- For grinding and turning tools (types 400 - 599), the correction value is applied as a function of the cutting edge position, i.e. it acts as a radius compensation when tool radius compensation is active and as a length compensation when tool radius compensation is deactivated in the axes specified by the cutting edge position.
On all other tool types, the correction value is applied only when tool radius compensation has been activated with G41 or G42. The correction value is canceled when tool radius compensation is deactivated with G40.

5.4 Grinding-specific tool monitoring

5.4.1 Function

Grinding-specific tool monitoring is a combination of geometry and speed monitoring. It can be activated for every grinding tool (tool type: 40x to 49x).

Note

Monitoring of one tool is not deselected if the monitoring function is selected for another tool provided the two tools are referred to different spindles.

One tool and thus also **one** tool monitor can be active for every spindle at any point in time.

Activated monitors remain active after a reset.

Geometry monitoring

The actual wheel radius and the actual wheel width are monitored.

The compensation value of the first cutting edge (D1) of a grinding tool which is referenced in the parameter \$TC_TPG9 is used as the current wheel radius. If monitoring is active, this value is compared to the value in parameter \$TC_TPG3 (minimum wheel radius).

Note

The monitoring function for the wheel radius remains active if an online tool offset is selected:

- when the monitoring function is activated
 - when the current radius (online tool offset, wear parameter) or the current width (\$TC_TPG5) is altered
-

5.4 Grinding-specific tool monitoring

The current wheel width is usually determined through the dressing cycle and can be entered in parameter \$TC_TPG5 of a grinding tool. If monitoring is active, this entered value is compared to the value in parameter \$TC_TPG4 (minimum wheel width).

If the current wheel radius falls below the value specified in the parameter \$TC_TPG3 or the current wheel width (\$TC_TPG5) falls below the value defined in the parameter \$TC_TPG4, the control sets the following NC/PLC interface signal to "1":

```
<Axis>.grinding.in.geoMonitorError (geometry monitoring) == 1
```

Note

There is no controller-internal error response.

Speed monitoring

The speed monitor checks the grinding wheel peripheral speed in m/s (parameter \$TC_TPG7) as well as the maximum spindle speed in rpm (parameter \$TC_TPG6).

The speed setpoint is monitored against the speed limitation cyclically with allowance for the spindle override.

The speed limit is the smaller value resulting from the comparison of the maximum spindle speed with the speed calculated from the maximum wheel peripheral speed and the current wheel radius.

The speed limit value is recalculated:

- if monitoring is switched on
- if online correction values (wear parameters) are changed

When the speed limit value is reached, it is restricted to the speed limit value and the following NC/PLC interface signal is set to "1":

```
<Axis>.grinding.in.speedMonitorError (speed monitoring) == 1
```

Note

There is no controller-internal error response.

Activation/deactivation

Activation/deactivation of grinding-specific tool monitoring is done by programming of TMON(...)/TMOF(...) (Page 343) in the part program.

PLC signals

NC → PLC

Basic Program Plus	Basic Program	
<Axis>.grinding.in.geoMonitorError	LBP_Axis*.E_GeoMonit	DB31, ...DBX83.3
<Axis>.grinding.in.speedMonitorError	LBP_Axis*.E_SpeedMonit	DB31, ...DBX83.6

5.4.2 Commissioning

Automatic activation

If, when selecting the tool length compensation of a grinding tool with an odd tool type number, the grinding-specific tool monitoring is to be automatically activated, the following machine data must be set to "1":

MD20350 \$MC_TOOL_GRIND_AUTO_TMON (activation of tool monitoring) = 1

5.4.3 Programming

5.4.3.1 Activate/deactivate grinding-specific tool monitoring (TMON, TMOF)

With the predefined procedures TMON(...) and TMOF(...), the grinding-specific tool monitoring is activated or deactivated (geometry and speed monitoring).

Requirement

The tool-specific parameters \$TC_TPG1 to \$TC_TPG9 must be set.

Syntax

```
TMON (<TNo>)
...
TMOF (<TNo>)
```

Meaning

TMON (. . .) :	Activate grinding-specific tool monitoring The command must be programmed in the channel in which the grinding-specific tool monitoring is to be activated.
TMOF (. . .) :	Deactivate grinding-specific tool monitoring The command must be programmed in the channel in which the grinding-specific tool monitoring is to be deactivated.
<TNo>:	T number Note: Only required if the monitoring is to be switched on or off for an inactive grinding wheel rather than the active grinding wheel that is currently in use.
TMOF (0) :	Deactivate monitoring for all tools

5.5 Constant grinding wheel peripheral speed (GWPS)

5.5.1 Function

For grinding wheels, generally the wheel peripheral speed is used instead of the spindle speed. The value to be set is determined by technological process parameters (e.g. grinding wheel characteristic, material combination).

To be able to specify the wheel peripheral speed in the part program, the tool-specific parameters \$TC_TPG1 (Page 322), \$TC_TPG8 (Page 324) and \$TC_TPG9 (Page 325) must be set and the function "Constant grinding wheel peripheral speed (GWPS)" must be activated.

Note

Activation of GWPS does **not** cause the automatic activation of tool length compensation or grinding-specific tool monitoring!

As soon as the function is active, any subsequent S value for the spindle that is assigned to the grinding wheel via \$TC_TPG1 will be interpreted as grinding wheel peripheral speed. The grinding wheel speed is set such that, taking account of the current wheel radius, the grinding wheel peripheral speed remains constant.

The formula for calculating the speed is as follows:

$$n[\text{rpm}] = \frac{\text{GWPS} [\text{m} \cdot \text{s}^{\text{rpm}}] \cdot 60}{2\pi \cdot R[\text{m}]}$$

The correction value to be used for calculation is defined in the tool-specific parameter \$TC_TPG9. The associated wear and the base dimension as a function of the tool type are added to this correction value. The sum total is divided by "cos" (\$TC_TPG8) when the value of parameter \$TC_TPG8 (angle of inclined wheel) is positive, and is divided by "sin" (\$TC_TPG8) when the value is negative.

The speed is recalculated in response to the following events:

- Specification a new grinding wheel peripheral speed, e.g. by:
 - Programming in the part program/overstoring
 - Assignment to the address "S" in MDI
 - Spindle control via PLC (SINU_PositionSpindle / SINU_RotateSpindle (Basic Program FC18))
- Changing the online correction values (wear parameters)

Activation/deactivation

Activation/deactivation of GWPS is implemented by programming GWPSON(...)/GWPSOF(...) in the part program (see "Programming (Page 346)").

Status

The following interface signal can be used to determine whether or not the GWPS is active:

`<Axis>.grinding.in.gwpsActive` (GWPS active)

New tool

If GWPS is to be selected with a new tool for a spindle for which GWPS is already active, the active GWPS must first be deselected (otherwise an alarm is output).

Response to reset/part program end

In the basic setting, GWPS is deactivated for a reset or end of part program and the spindle is stopped. The machine data settings behind this behavior can be changed as needed (see "Commissioning (Page 345)").

Additional properties

- GWPS can be activated for every grinding tool (tool type: 400 to 499) in each channel.
- GWPS can be active simultaneously for several spindles, each with a different grinding tool, in the same channel.

PLC signals

NC → PLC

Basic Program Plus	Basic Program	
<code><Axis>.grinding.in.gwpsActive</code>	<code>LBP_Axis*.E_SUG</code>	<code>DB31, ...DBX84.1</code>

5.5.2 Commissioning

Automatic activation

If the constant grinding wheel peripheral speed (GWPS) is to be automatically selected for a spindle after Power On, and remain selected during mode switch, reset or end of the part program, the following machine data bit must be set to "1":

MD35032 \$MA_SPIND_FUNC_RESET_MODE Bit 0 = 1

Note

The prerequisite is that the tool be a grinding tool (tool type 400 to 499), the tool-specific parameters \$TC_TPG1, \$TC_TPG8 and \$TC_TPG9 are set, and \$TC_TPG1 refers to a valid spindle.

Behavior of spindle after reset/part program end

If the spindle is not to stop after a reset or end of part program, but is to continue rotating at the current speed, the following machine data must be set to "1":

MD35040 \$MA_SPIND_ACTIVE_AFTER_RESET = 1

Note

MD35040 only takes effect in the spindle mode open-loop control operation.

5.5.3 Programming

5.5.3.1 Switching constant grinding wheel peripheral speed (GWPSON, GWPSOF) on/off:

With the predefined procedures GWPSON(...) and GWPSOF(...), the constant grinding wheel peripheral speed (GWPS) for grinding tools (tool type: 400 to 499) is switched on and off.

Syntax

```
GWPSON (<TNo>)  
S<n>=... :  
...  
GWPSOF (<TNo>)
```

Meaning

GWPSON (. . .) :	Switch on the constant grinding wheel peripheral speed
GWPSOF (. . .) :	Switch off the constant grinding wheel peripheral speed
<TNo>:	T number Note: Only required if the constant grinding wheel peripheral speed is to be switched on or off for an inactive grinding wheel rather than the active grinding wheel that is currently in use.
S<n>=...:	Grinding wheel peripheral speed in m/s or ft/s for spindle <n>
S0=... or S . . . :	Grinding wheel peripheral speed for the master spindle

Query status

The following system variable can be used to query from the part program whether the constant grinding wheel peripheral speed is active for a specific spindle.

\$P_GWPS[<n>] ; where <n> = spindle number

Value	Meaning
0 (= FALSE)	GWPS is inactive .
1 (= TRUE)	GWPS is active .

5.5.4 Example

A constant grinding wheel peripheral speed is to be used for grinding tools T1 and T5. T1 is the active tool.

Data of tool T1 (peripheral grinding wheel)

\$TC_DP1[1,1] = 403 ;tool type
 \$TC_DP3[1,1] = 300 ;length1
 \$TC_DP4[1,1] = 50 ;length2
 \$TC_DP12[1,1] = 0 ;wear length 1
 \$TC_DP13[1,1] = 0 ;wear length 2
 \$TC_DP21[1,1] = 300 ;base length 1
 \$TC_DP22[1,1] = 400 ;base length 2
 \$TC_TPG1[1] = 1 ;spindle number
 \$TC_TPG8[1] = 0 ;angle of inclined wheel
 \$TC_TPG9[1] = 3 ;parameter no. for radius calculation

Data of tool T5 (inclined grinding wheel)

\$TC_DP1[5,1] = 401 ;tool type
 \$TC_DP3[5,1] = 120 ;length1
 \$TC_DP4[5,1] = 30 ;length2
 \$TC_DP12[5,1] = 0 ;wear length 1
 \$TC_DP13[5,1] = 0 ;wear length 2
 \$TC_DP21[5,1] = 100 ;base length 1
 \$TC_DP22[5,1] = 150 ;base length 2
 \$TC_TPG1[5] = 2 ;spindle number
 \$TC_TPG8[5] = 45 ;angle of inclined wheel
 \$TC_TPG9[5] = 3 ;parameter no. for radius calculation

Programming

Program code	Comment
N20 T1 D1	; select T1 and D1
N25 S1=1000 M1=3	; 1000 rpm for spindle 1
N30 S2=1500 M2=3	; 1500 rpm for spindle 2
...	
N40 GWPSON	; activate GWPS for active tool T1

5.5 Constant grinding wheel peripheral speed (GWPS)

Program code	Comment
N45 S[\$P_ATPG[1]]=60	; set GWPS for active tool to 60 m/s → n=1909.85 rpm
...	
N50 GWPSON(5)	; activate GWPS for tool 5 (spindle 2)
N55 S[\$TC_TPG1[5]]=40	; set GWPS for spindle 2 to 40 m/s → n=1909.85 rpm
...	
N60 GWPSOF	; deactivate GWPS for active tool
N65 GWPSOF(5)	; deactivate GWPS for tool 5 (spindle 2)
...	

Appendix

A.1 List of abbreviations

A	
A	Output
AFIS	Automatic Filter Switch: Automatic filter switch
ASCII	American Standard Code for Information Interchange: American coding standard for the exchange of information
ASIC	Application Specific Integrated Circuit: User switching circuit
ASUP	Asynchronous subprogram
AUTO	Operating mode "Automatic"
AUXFU	Auxiliary Function: Auxiliary function
AWL	Statement list

B	
BAG	Mode group
BCD	Binary Coded Decimals: Decimal numbers encoded in binary code
BICO	Binector Connector
BIN	Binary Files: Binary files
BKS	Basic coordinate system
BM	Operating alarm
BO	Binector Output
BTSS	Operator panel interface

C	
CLC	Clearance control
CNC	Computerized Numerical Control: Computerunterstützte numerische Steuerung
COM	Communication
CP	Communication Processor
CPU	Central Processing Unit: Central processing unit
CST	Configured Stop: Configured stop

D	
DDS	Drive Data Set: Drive data set
DIR	Directory: Directory
DO	Drive Object
DRF	Differential Resolver Function: Differential resolver function (handwheel)

A.1 List of abbreviations

D	
DRY	Dry Run: Dry run feedrate
DWORD	Double word (currently 32 bits)

E	
E	Input
EES	Execution from External Storage
E/A	Input/Output
ESR	Extended stop and retract
ETC	ETC key ">"; Softkey bar extension in the same menu

F	
FDD	Feed Disable: Feed disable
FdStop	Feed Stop: Feed stop
FIFO	First In First Out: Memory that works without address specification and whose data is read in the same order in which they was stored
FM	Error message
FUP	Control system flowchart (PLC programming method)
FW	Firmware

G	
GEO	Geometry, e.g. geometry axis
GP	Basic program (PLC)
GUD	Global User Data: Global user data

H	
HEX	Abbreviation for hexadecimal number
HiFu	Auxiliary function
HMI	Human Machine Interface: SINUMERIK user interface
HSA	Main spindle drive
HT	Handheld Terminal
HW	Hardware

I	
IBN	Commissioning
INC	Increment: Increment
INI	Initializing Data: Initializing data
IPO	Interpolator

J	
JOG	JOGging: Setup mode

K	
KOP	Ladder diagram (PLC programming method)

L	
LED	Light Emitting Diode: Light-emitting diode
LMS	Position measuring system
LR	Position controller

M	
Main	Main program: Main program (OB1, PLC)
MCP	Machine Control Panel: Machine control panel
MD	Machine Data
MDA	Manual Data Automatic: Manual input
MDS	Motor Data Set: Motor data set
MELDW	Message word
MKS	Machine coordinate system
MM	Motor Module
MPF	Main Program File: Main program (NC)
MPI	Multi Point Interface
MSTT	Machine control panel

N	
NC	Numerical Control: Numerical control with block preparation, traversing range, etc.
NCU	Numerical Control Unit: NC hardware unit
NCK	Numerical Control Kernel
NCSD	NC Start Disable
NST	Interface signal
NV	Work offset
NX	Numerical Extension: Axis expansion module

O	
OB	Organization block in the PLC
OEM	Original Equipment Manufacturer
OP	Operation Panel: Operating equipment

P	
PCU	PC Unit: PC box (computer unit)
PG	Programming device
PI	Program instance
PLC	Programmable Logic Control: Controller
PN	PROFINET
PO	Power On
POS	Position/positioning
PPO	Parameter process data object: Cyclic data telegram for PROFIBUS DP transmission and "Variable speed drives" profile
PPU	Panel Processing Unit (central hardware for a panel-based CNC, e.g. SINUMERIK 828D)
PROFIBUS	Process Field Bus: Serial data bus
PRT	Program test
PTP	Point to Point: Point-to-point
PZD	Process data: Process data part of a PPO

R	
REF	REference point approach function
REPOS	REPOSition function
RESU	Retrace support
RID	Read In Disable
RP	R Parameter, arithmetic parameter, predefined user variable

S	
SA	Synchronized action
SBL	Single Block: Single block
SBT	Safe Brake Test
SCC	Safety Control Channel
SCL	Structured Control Language
SD	Settingdatum or setting data
SDI	Safe Direction
SERUPRO	Search-Run by Program Test: Block search via program test
SIC	Safety Info Channel
SKP	Skip: Function for skipping a part program block
SLP	Safe Limited Position
SLS	Safely Limited Speed
SMI	Sensor Module Integrated
SOS	Safe Operating Stop
SPF	Sub Program File: Subprogram (NC)
SS1	Safe Stop 1
SS2	Safe Stop 2
STO	Safe Torque Off

S	
STW	Control word
SUG	Grinding wheel peripheral speed
SW	Software

T	
TCU	Thin Client Unit
TIA	Totally Integrated Automation
TM	Terminal Module (SINAMICS)
TO	Tool Offset: Tool offset
TOA	Tool Offset Active: Identifier (file type) for tool offsets
TOFF	Online tool length offset
TRANSMIT	Transform Milling Into Turning: Coordination transformation for milling operations on a turning machine

U	
UP	Subprogram
USB	Universal Serial Bus

V	
VDI	Internal communication interface between NC and PLC

W	
WKS	Workpiece coordinate system
WPD	Work Piece Directory: Workpiece directory
WZ	Tool
WZV	Tool management

Z	
ZSW	Status word (of drive)

Index

\$

\$AC_CUT_INV, 209
\$AC_CUTMOD, 209
\$AC_CUTMOD_ANG, 188, 209
\$AC_CUTMODK, 209
\$NT_ROT_AX_NAME, 206
\$P_AD, 208
\$P_CUT_INV, 209
\$P_CUTMOD, 209
\$P_CUTMOD_ANG, 188, 209
\$P_CUTMOD_ERR, 210
\$P_CUTMODK, 209
\$P_GWPS, 347
\$P_ORI_DIFF, 202
\$P_ORI_POS, 202
\$P_ORI_SOL, 203
\$P_ORI_STAT, 205
\$P_TCANG[<n>], 135
\$P_TOOLENV, 257
\$P_TOOLENVN, 257
\$TC_CARR_CORR_ELEM, 150
\$TC_CARR<n>[<m>], 132
\$TC_CARR13, 145
\$TC_CARR14, 145
\$TC_DP1, 183, 320
\$TC_DP10, 185
\$TC_DP11, 184
\$TC_DP19, 189
\$TC_DP2, 184, 194
\$TC_DP24, 185
\$TC_DPC1...10, 320
\$TC_DPCE[t,d], 28
\$TC_DPROT, 190
\$TC_TPC1...10, 326
\$TC_TPG_DRSPATH, 325
\$TC_TPG_DRSPROG, 325
\$TC_TPG3, 323
\$TC_TPG4, 323
\$TC_TPG5, 323
\$TC_TPG8, 324
\$TC_TPG9, 325

3

3D face milling, 48

A

Adapter dimension, 50
Alarm for preprocessing stop, 70
Angle
 Clearance, 185
 Holder, 185
Approach point/angle, 58
Assignment tool/tool carrier, 121

B

Behavior at inside corners, 294

C

CDOF, 89
CDOF2, 89
CDON, 89
Clamping angle, 190
Clearance angle, 185
Collision detection, 89
Collision monitoring, 88
Compensation
 Plane, 114
 Tool length, 19
 Tool radius, 20
Contour
 Approach/leave, 81
CORRTC, 147
Cut direction, 184
CUT2D, 111
CUT2DD, 111
CUT2DF, 111
CUT2DFD, 111
CUT3DC, 297
CUT3DCC, 308
CUT3DCCD, 308
CUT3DCD, 297
CUT3DF, 302
CUT3DFD, 302
CUT3DFF, 302
CUT3DFS, 302
CUTMOD, 206
CUTMODK, 206

Cutting edge
 Number of contour tools, 113
 position, 194
Cutting edge number, 28
Cutting edge position, 44
 relevant, 278
Cutting edges
 Center point, 188
 Position, 184
 -reference point, 188

D

D function, 23
D numbers
 Allocation of free ..., 28
DELTOOLENV, 255
Description of a rotation, 128
DISC, 62, 63, 70, 85
Dressing, 330

E

Evaluation of individual wear components, 251

F

Face machining, 54
Face milling, 288
FCTDEF, 333
Frame change, 122
FTOCOF, 336
FTOCON, 336

G

G40, 74
G41, 74
G42, 74
G450, 62, 85, 285
G451, 62, 85, 285
G460, 109
G461, 109
G462, 110
G91 extension, 213
 Zero point offset, 214
GETTCOR, 257
GETTENV, 256
Grinding Wheel Peripheral Speed, 344
Grooving saw, 43

GWPS, 344, 346
GWPSOF, 346
GWPSON, 346

H

Holder angle, 185

I

Inclined position, 189
Incrementally programmed offset values, 213
Insertion depth, 283, 300
ISD, 283

K

Kinematic type, 131, 133
KONT, 58, 81
KONTC, 58, 81
KONTT, 58, 81

L

LENTOAX, 275
Limited tool carrier orientation, 122
Longitudinal machining, 54

M

Machine kinematics, 131
Machine with rotary tool, 125
Machine with rotary workpiece, 126
Machines with extended kinematics, 126
Machining in direction of tool orientation, 214
MD10210, 147
MD11346, 36
MD11410, 89
MD11411, 89
MD18075, 332
MD18080, 235
MD18088, 132, 163
MD18096, 320
MD18100, 24, 321
MD18104, 254
MD18105, 28, 29, 30, 31
MD18106, 23, 28, 29
MD18108, 238
MD18112, 239
MD18114, 221

MD18116, 255
MD20096, 26, 27
MD20110, 168, 214, 243
MD20112, 168
MD20120, 168
MD20121, 23
MD20125, 187, 195
MD20126, 168, 196
MD20127, 196
MD20180, 160
MD20184, 162
MD20188, 128
MD20190, 128
MD20202, 104
MD20204, 98
MD20210, 64, 71, 285
MD20220, 63, 70
MD20230, 65, 71, 285
MD20240, 67, 71, 89
MD20250, 71
MD20252, 94
MD20254, 332, 340
MD20256, 72
MD20270, 23, 244
MD20272, 238, 244
MD20350, 343
MD20360, 36, 187, 196, 261
MD20390, 230, 232, 233
MD20392, 231
MD20610, 174, 336
MD20620, 174
MD20621, 174
MD20622, 175
MD20623, 175
MD20624, 175
MD21080, 296
MD21082, 297
MD21084, 291, 297
MD21190, 172
MD21194, 173
MD21196, 173
MD22530, 168
MD22550, 21, 33, 34
MD22560, 22
MD22562, 35, 36
MD24100, 232
MD24110, 232
MD24120, 232
MD24558, 262
MD24570, 232
MD24572, 232
MD24658, 262

MD25574, 233
MD28082, 162
MD28085, 24, 164
MD28291, 297
MD28620, 118
MD32084, 175
MD32750, 230, 232, 233
MD35032, 345
MD35040, 346
MD52210, 175
MD52211, 175
Milling tool machining point, 301
Milling tool reference point, 301
Milling tool tip, 301

N

NORM, 58, 81

O

OFFN, 74
Offset number, 28
Online tool length compensation, 170, 175
Online tool offset, 330
Online tool radius compensation:, 340
ORIC, 285, 294
ORID, 285, 294
ORISOLH, 198
Outside corners/inside corners, 284

P

Parameter set
 Tool carriers with orientation capability, 131, 132
Parameterization of the axes of rotation, 131
Predictive contour calculation, 67
PUTFTOC, 335
PUTFTOCF, 334

S

SD42440, 213
SD42442, 213
SD42480, 72
SD42494, 114
SD42496, 70, 72, 73
SD42900, 226, 246
SD42910, 227, 246, 251, 252
SD42920, 227, 246, 251, 252

SD42930, 228, 246, 251
SD42935, 156, 247, 252, 279
SD42940, 216, 246, 251
SD42950, 183, 217, 246, 251
SD42954, 218
SD42956, 218
SD42960, 231, 232
SD42970, 174
SD42972, 174
SD42974, 128
SD42977, 117
SD42984, 208
SD42998, 186, 197
SD42999, 198
SETTCOR, 263
Smooth approach and retraction
 Significance, 94
 Sub-movements, 95
Storing angles in the tool carrier data, 123

T

T function, 21
TCARR, 143, 161
TCOABS, 143, 161
TCOFR, 143, 161
TCOFRX, 143
TCOFRY, 143
TCOFRZ, 143
TMOF, 343
TMON, 343
TOA
 data, 40
 -unit, 24
TOFFOF, 175
TOFFON, 175
Tool
 Base dimension/adapter dimension, 50
 Cutting edge, 37
 DISC, 63, 70
 Group, 40
 Length, 45
 length compensation, 19
 offset data, 40
 parameters, 37
 radius compensation, 20
 shape, 46
 shape, active, 50
 size, active, 50
 Tool change, 21
 Tool cutting edge, 23
 Tool radius compensation 2D (TRC), 55

 Tool selection, 21
 Type, 40
 wear, 49, 50
Tool base dimension, 50
Tool carrier
 -orientable, 143
Tool carrier, with orientation capability
 Constraints, 167
 Control system response on reset, program start, repositioning, 167
 swiveling working table, 160
Tool carriers with orientation capability, 119
 Create new, 163
 Kinematic chain, 124
Tool carriers, with orientation capability, 119
 Create new, 163
 Inclined machining, 159
Tool change
 Offset memory, 24
Tool compensation
 Taken into account in the NC, 26
Tool length compensation
 calculate tool-specifically, example, 154
 Geometry, 45
 Wear, 49
 Workpiece-specific calculation, 245
Tool monitoring
 grinding-specific, 341
Tool offset
 Types, 122
Tool radius compensation
 At outside corners, 62, 85
 CUT2DF, 114
 keep constant, 91
 Wear, 50
Tool radius compensation, 2D
 Approach and retraction behavior, 58
 Geometry, 46
 Modified alarm response, 70
 Point of intersection G451, 62
 Smooth approach and retraction, 94
 Transition circle, 62
 variable compensation value, 68
Tool type, 320
TOOLENV, 253
Toolholder with orientation capability
 Calculation of active tool length, 160
 Examples, 151
TOWBCS, 250
TOWKCS, 251
TOWMCS, 249
TOWSTD, 249

TOWTCS, 251
TOWWCS, 250

U

Undercut angle, 53

Z

Zero vectors, 124

