SIEMENS

SINUMERIK 840D sl/ 840D/840Di/810D

Fundamentals

Programming Manual

Valid for

Programmable Controllers
SINUMERIK 840D sl/840DE sl
SINUMERIK 840D powerline/840DE powerline
SINUMERIK 840D powerline/840DE powerline
SINUMERIK 840D powerline/840DE powerline

Software	Version
NCU system software for 840D sl/840DE sl	1.3
NCU system software for 840D/840DE	7.3
NCU system software for 840Di/840DiE	2.3
NCU system software for 810D/810DE	7.3

Edition 08/2005

6FC5398-1BP10-0BA0

Safety Guidelines

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 to property damage only 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**

with a safety alert symbol, indicates that minor personal injury can result if proper precautions are not taken.

Caution

without a safety alert symbol, indicates that property damage can result if proper precautions are not taken.

Notice

indicates that an unintended result or situation can occur if the corresponding information is not taken into account.

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 device/system may only be set up and used in conjunction with this documentation. Commissioning and operation of a device/system may only be performed by **qualified personnel**. Within the context of the safety notes in this documentation qualified persons are defined as persons who are authorized to commission, ground and label devices, systems and circuits in accordance with established safety practices and standards.

Prescribed Usage

Note the following:

⚠ **Warning**

This device may only be used for the applications described in the catalog or the technical description and only in connection with devices or components from other manufacturers which have been approved or recommended by Siemens. Correct, reliable operation of the product requires proper transport, storage, positioning and assembly as well as careful operation and maintenance.

Trademarks

All names identified by ® are registered trademarks of the Siemens AG. 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.

Siemens AG
Automation and Drives
Postfach 4848, 90327 Nuremberg, Germany

Siemens AG 2005
Technical data subject to change

Preface

Preface

Structure of the Documentation

The SINUMERIK documentation is organized in 3 parts:

- General Documentation
- User Documentation
- Manufacturer/service documentation

Please contact your local Siemens office for more detailed information about other SINUMERIK 840D sl/840D/840Di/810D publications and publications that apply to all SINUMERIK controls (e.g. universal interface, measuring cycles, etc.).

An overview of publications (updated monthly) indicating the language versions available can be found on the Internet at:
http://siemens.com/motioncontrol
Select "Support" followed by "Technical Documentation" and then "Overview of Publications".

The Internet version of DOConCD (DOConWEB) is available at:
http://automation.siemens.com/doconweb

Target readership

This document is designed for machine tool users. This publication describes in detail all the facts the user needs to know for programming the SINUMERIK 840D sl/840D/840Di/810D controls.

Standard scope

This Programming Guide describes the functionality afforded by standard functions. Extensions or changes made by the machine tool manufacturer are documented by the machine tool manufacturer.

Other functions not described in this documentation might be executable in the control. This does not, however, represent an obligation to supply such functions with a new control or when servicing.

Hotline

If you have any questions, please get in touch with our hotline:

A&D Technical Support
Tel.: +49 (0)180 5050-222
Fax: +49 (0)180 5050-223
Internet: http://siemens.de/automation/support-request

If you have any comments, suggestions, or corrections regarding this documentation, please fax or e-mail them to:

Fax: +49 (0)9131 98-63315
E-mail: motioncontrol.docu@erlf.siemens.com

Fax form: See the reply form at the end of the document.

Internet address

http://www.siemens.com/motioncontrol

Export version

The following functions are not available in the export version:

Function	810DE	840DE sl	840DE	840DiE
Helical interpolation 2D+6 (Basic version, no options)	–	–	–	–
Milling machining package	–	–	–	–
Five axis machining package	–	–	–	–
Handling transformation package	–	–	–	–
Multi-axis interpolation (> 4 interpolating axes)	–	–	–	–
OA NCK compile cycles	–	–	–	–
Clearance control 1D/3D in position-control cycle 1)	–	–	–	–
Synchronized actions 1) (Basic version, no options)	#	#	#	#
Master-value coupling and curve-table interpolation	#	#	#	#
Sag compensation, multi-dimensional	#	#	#	#
Synchronized actions, stage 2 1)	–	–	#	#
Electronic gear 1)	–	–	#	#
Electronic transfer	–	–	#	#
	# Restricted functionality - Function not available			

1) The restricted functions for the SINUMERIK 810DE powerline SINUMERIK 840DE sl/SINUMERIK 840DE powerline/SINUMERIK 840DiE powerline export versions impose a limit of "max. 4 interpolating axes".

Description

Fundamentals

This Programming Guide "Fundamentals" is intended for use by skilled machine operators with the appropriate expertise in drilling, milling and turning operations. Simple programming examples are used to explain the commands and statements which are also defined according to DIN 66025.

Job planning

The Programming Guide "Job Planning" is intended for use by technicians with in-depth, comprehensive programming knowledge. By virtue of a special programming language, the SINUMERIK 840D sl/840D/840Di/810D control enables the user to program complex workpiece programs (e.g. for free-form surfaces, channel coordination, ...) and greatly facilitates the programming of complicated operations.

The commands and statements described in this Guide are not specific to one particular technology.

They can be used for a variety of tasks, such as:

- Turning, milling and grinding
- Cyclical machines (packaging, woodworking)
- Laser power controls.

Table of contents

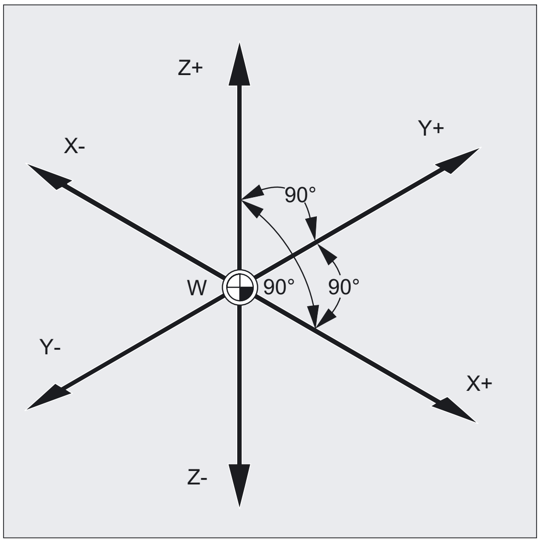
Fundamental Geometrical Principles 1

1.1 Description of workpiece points

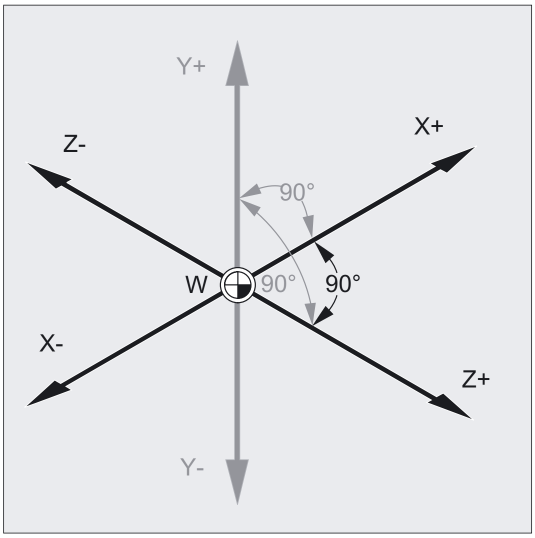
1.1.1 Workpiece coordinate systems

In order for the machine or control to operate with the specified positions, these data must be entered in a reference system that corresponds to the direction of motion of the axis slides. A coordinate system with the axes X, Y and Z is used for this purpose.

Milling:



Turning:



DIN 66217 stipulates that machine tools must use right-handed, rectangular (Cartesian) coordinate systems.

The workpiece zero (W) is the origin of the workpiece coordinate system. Sometimes it is advisable or even necessary to work with negative positional data. Positions to the left of the origin are prefixed by a negative sign (–).

1.1.2 Definition of workpiece positions

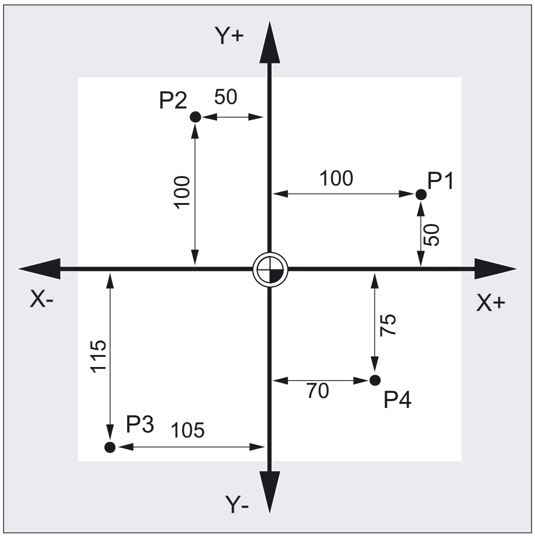
To specify a position, imagine that a ruler is placed along the coordinate axes. You can now describe every point in the coordinate system by specifying the direction (X, Y and Z) and three numerical values. The workpiece zero always has the coordinates X0, Y0, and Z0.

The infeed depth must also be described in milling operations.

One plane is sufficient to describe the contour on a lathe.

Workpiece positions in the working area

For the sake of simplicity, we will only use one plane of the coordinate system in this example, i.e., the X/Y plane. Points P1 to P4 then have the following coordinates:

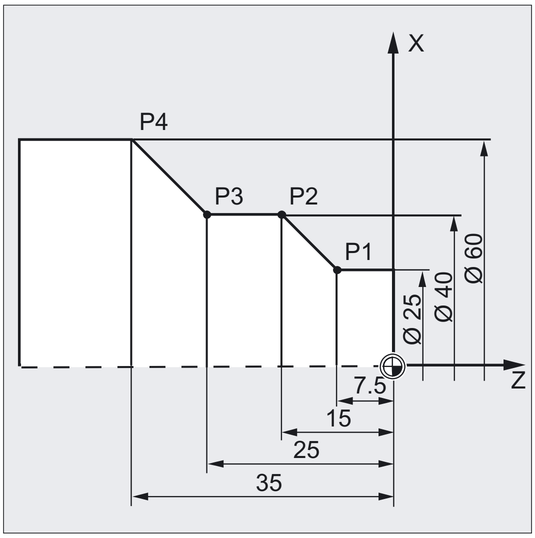


P1 corresponds to X100 Y50

P2 corresponds to X-50 Y100

P3 corresponds to X-105 Y-115

P4 corresponds to X70 Y-75



The workpiece positions are required only in one plane for turning.

Points P1 to P4 are defined by the following coordinates:

P1 corresponds to X25 Z-7.5

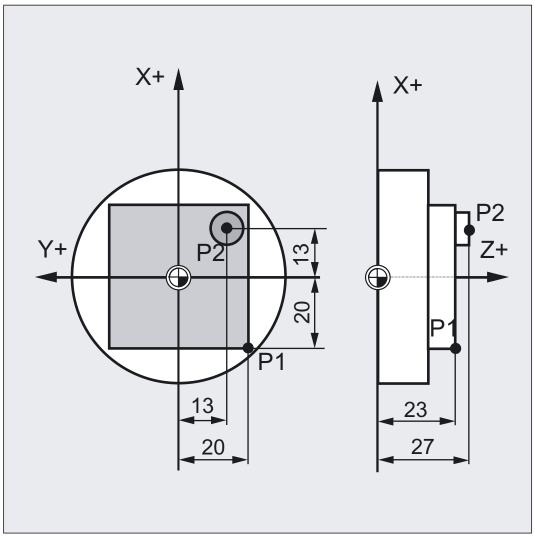
P2 corresponds to X40 Z-15

P3 corresponds to X40 Z-25

P4 corresponds to X60 Z-35

Example of turning positions

Points P1 and P2 are defined by the following coordinates:

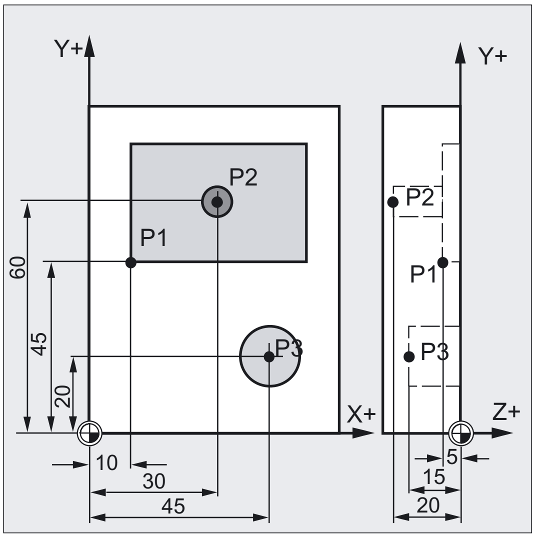


P1 corresponds to X-20 Y-20 Z23

P2 corresponds to X13 Y-13 Z27

Example: Positions for milling

To state the infeed depth, we need to specify a numerical value for the third coordinate (Z in this case).



Points P1 to P3 are defined by the following coordinates:

P1 corresponds to X10 Y45 Z-5

P2 corresponds to X30 Y60 Z-20

P3 corresponds to X45 Y20 Z-15

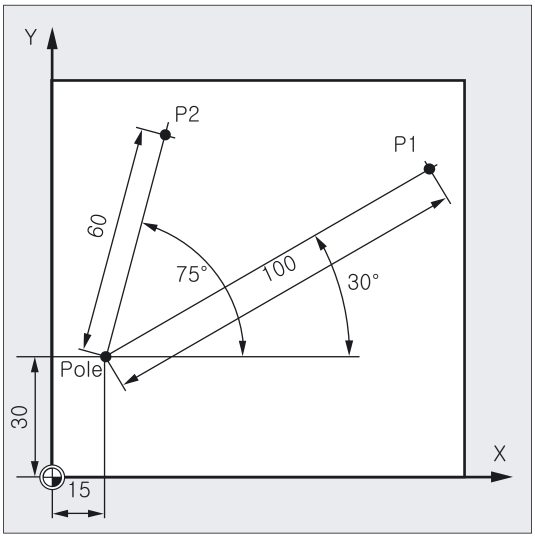
1.1.3 Polar coordinates

The method used to date to specify points in the coordinate system is known as the "Cartesian coordinate" method.

However, there is another way to specify coordinates, i.e., as so-called "polar coordinates". The polar coordinate method is useful only if a workpiece or part of a workpiece has radius and angle measurements. The point, on which the measurements are based, is called the "pole".

Example of polar data

The points P1 and P2 can then be described, with reference to the **pole**, as follows:



P1 corresponds to radius = 100 plus angle = 30°

P2 corresponds to radius = 60 plus angle = 75°

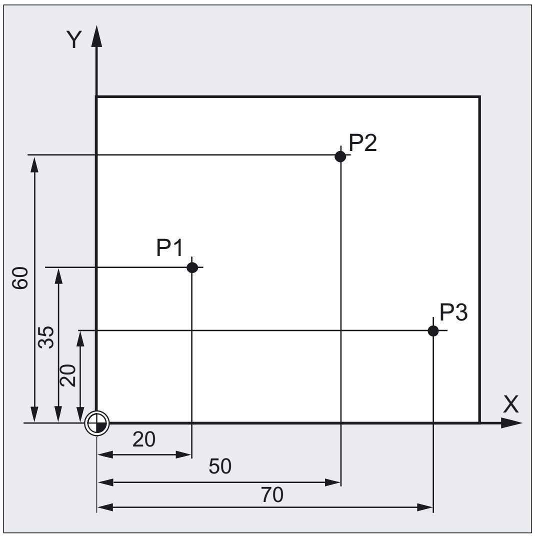
1.1.4 Absolute dimensions

With absolute dimensions, all the positional data refer to the currently valid zero point. Applied to tool movement this means:

the position, to which the tool is to travel.

Example of milling

The positional parameters for points P1 to P3 in absolute dimensions **referring to the zero point** are the following:



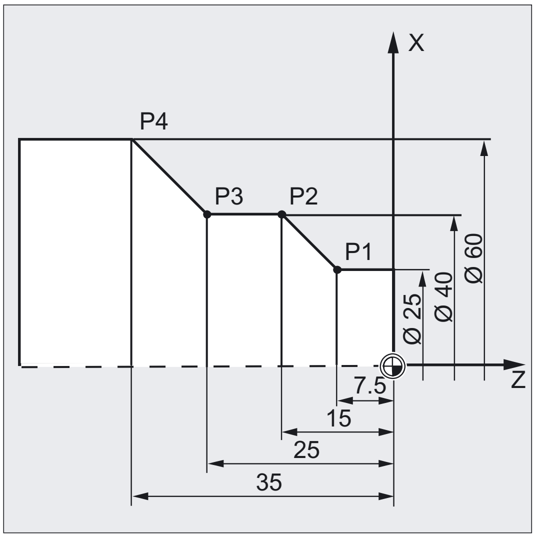
P1 corresponds to X20 Y35

P2 corresponds to X50 Y60

P3 corresponds to X70 Y20

Example of turning

The positions for points P1 to P4 in absolute dimensions are as follows **with reference to the zero point**:



P1 corresponds to X25 Z-7.5

P2 corresponds to X40 Z-15

P3 corresponds to X40 Z-25

P4 corresponds to X60 Z-35

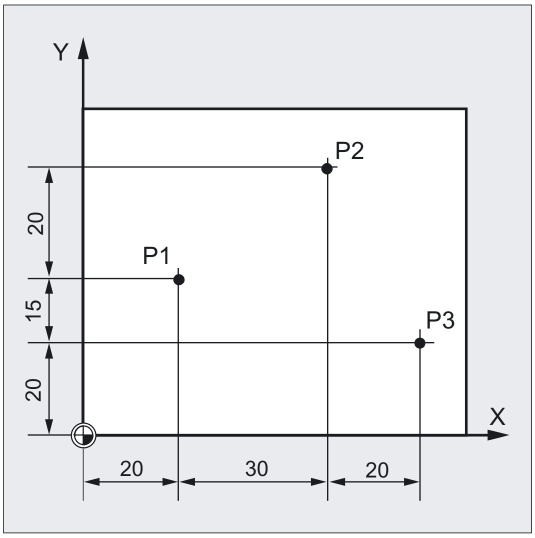
1.1.5 Incremental dimension

Production drawings are frequently encountered, however, where the dimensions refer not to the origin, but to another point on the workpiece. In order to avoid having to convert such dimensions, it is possible to specify them in incremental dimensions. Incremental dimensions refer to the positional data for the previous point. Applied to tool movement this means:

The incremental dimensions describe the distance the tool is to travel.

Example of milling

The positional data for points P1 to P3 in incremental dimensions are:



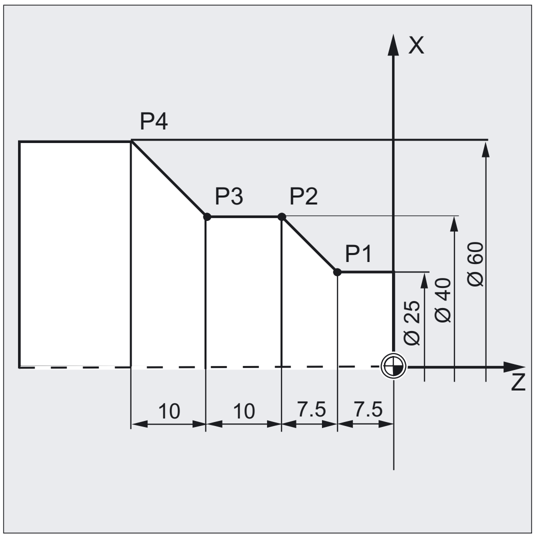
P1 corresponds to X20 Y35 ;(with reference to the zero point)

P2 corresponds to X30 Y20 ;(with reference to P1)

P3 corresponds to X20 Y-35 ;(with reference to P2)

Example of turning

The positions for points P1 to P4 in incremental dimensions are as follows:



G90 P1 corresponds to X25 Z-7.5 ;(with reference to the zero point)

G91 P2 corresponds to X15 Z-7.5 ;(with reference to P1)

G91 P3 corresponds to Z-10 ;(with reference to P2)

G91 P4 corresponds to X20 Z-10 ;(with reference to P3)

Note

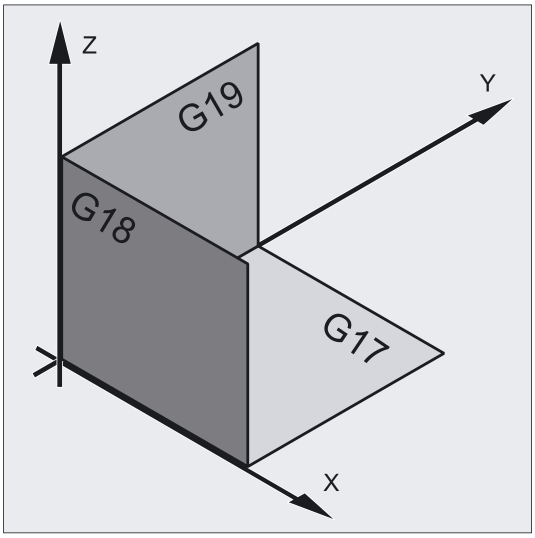
When DIAMOF or DIAM90 is active, the path setpoint is programmed as a radius dimension with G91.

1.1.6 Plane designations

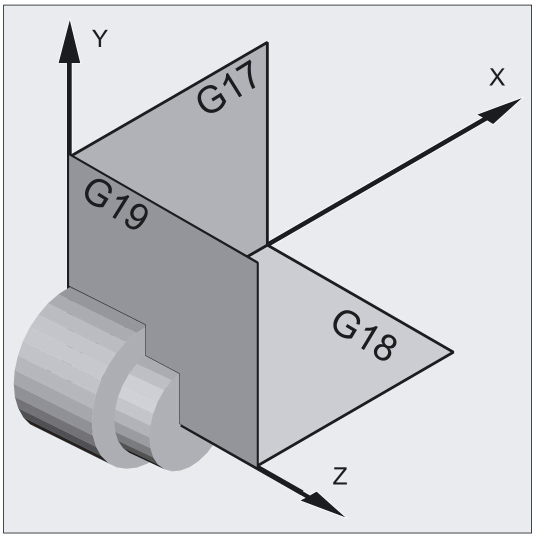
When programming, it is necessary to specify the working plane so that the control system can calculate the tool offset values correctly. The plane is also relevant to certain types of circular programming and polar coordinates.

A plane is defined by means of two coordinate axes.

Milling:



Turning:



The third coordinate axis is perpendicular to this plane and determines the infeed direction of the tool (e.g., for 2½ D machining).

Working planes

The working planes are specified as follows in the NC program with G17, G18 and G19:

Plane	*Designation*	*Infeed direction*
X/Y	G17	Z
Z/X	G18	Y
Y/Z	G19	X

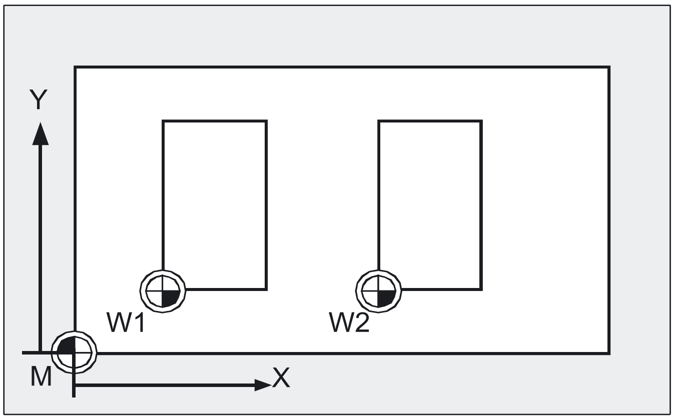
1.2 Position of zero points

The various origins (zero points) and reference positions are defined on the NC machine. They are reference points

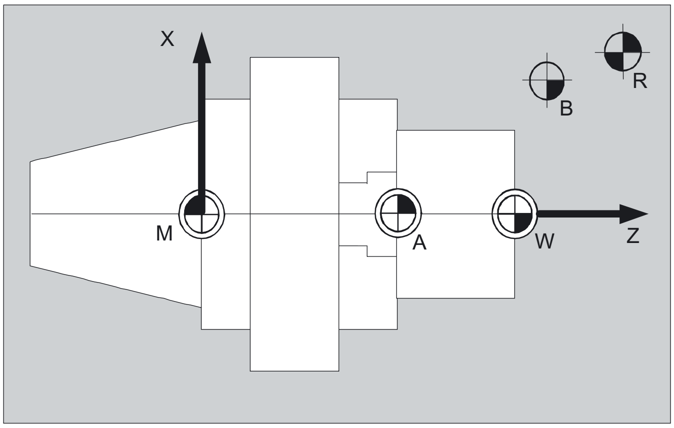
- for the machine to approach and
- for programming the workpiece dimensions.

The diagrams show the zero points and reference points for drilling/milling machines and turning machines.

Milling:



Turning:



Reference points

They are:

M	Machine zero
A	Blocking point. Can coincide with the workpiece zero point (only turning machines).
W	Workpiece zero = Program zero
B	Start point. Can be defined for each program. Start point of the first tool for machining.
R	Reference point. Position determined by cams and measuring system. The distance to the machine zero M must be known, so that the axis position can be set at this place exactly on this value.

1.3 Position of coordinate systems

1.3.1 Overview of various coordinate systems

We distinguish between the following coordinate systems:

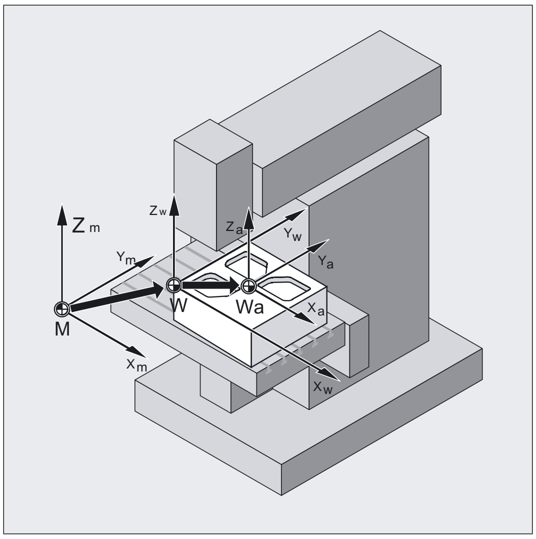
- The machine coordinate system with the machine zero M
- The basic coordinate system (this can also be the workpiece coordinate system W)
- The workpiece coordinate system with the workpiece zero W
- The current workpiece coordinate system with the current offset workpiece zero Wa

In cases where different machine coordinate systems are in use (e.g., 5-axis transformation), an internal transformation function mirrors the machine kinematics on the coordinate system currently selected for programming.

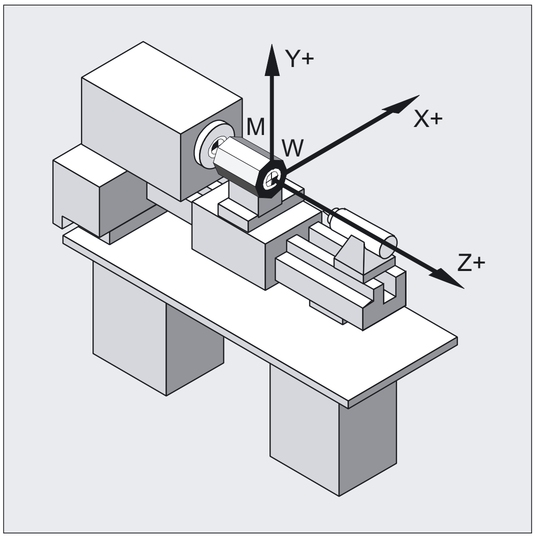
Note

The individual axis identifiers are explained in the section headed "Axis types".

Milling coordinate system:



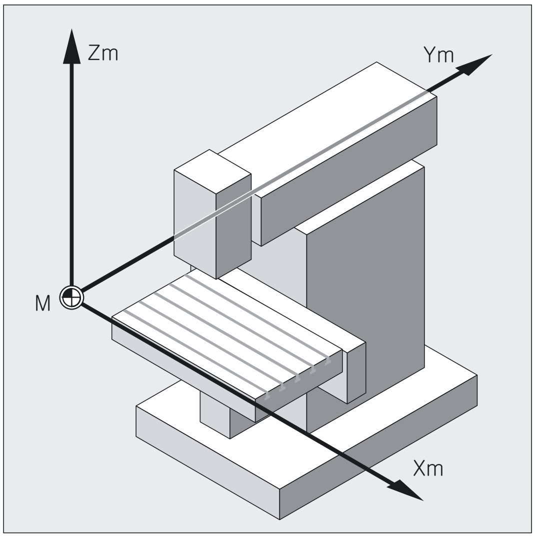
Turning coordinate system:



1.3.2 Machine coordinate system

The machine coordinate system comprises all the physically existing machine axes.

Reference points and tool and pallet changing points (fixed machine points) are defined in the machine coordinate system.



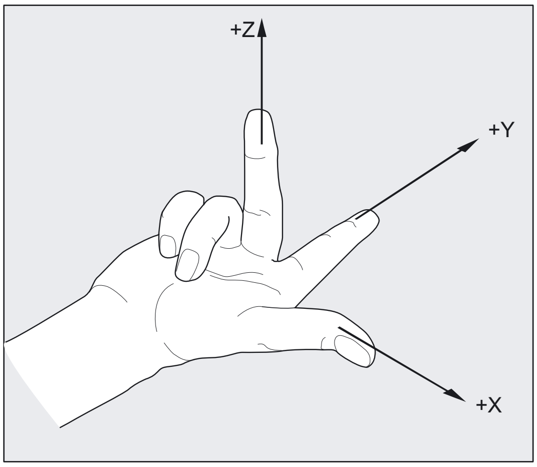
Where the machine coordinate system is used for programming (this is possible with some of the G functions), the physical axes of the machine are addressed directly. No allowance is made for workpiece clamping.

Right-hand rule

The orientation of the coordinate system relative to the machine depends on the machine type. The axis directions follow the so-called "three-finger rule" of the right hand (in accordance with DIN 66217).

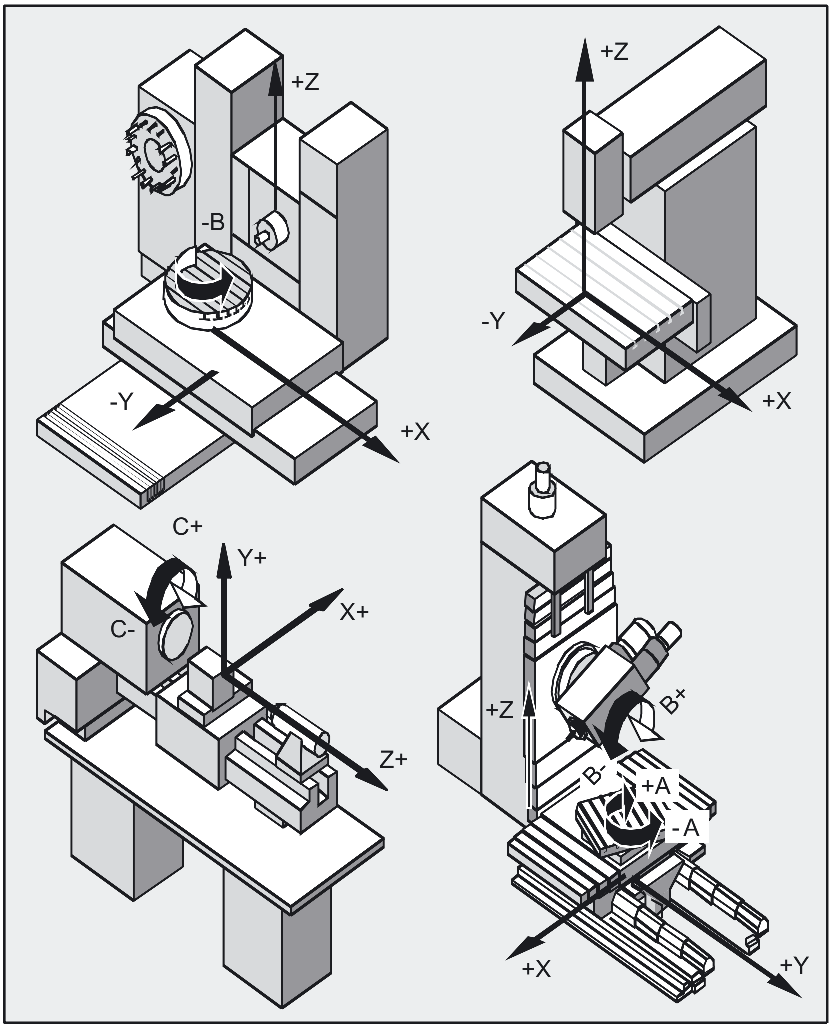
Seen from in front of the machine, the middle finger of the right hand points in the opposite direction to the infeed of the main spindle. Therefore:

- the thumb points in the +X direction
- the index finger points in the +Y direction
- the middle finger points in the +Z direction



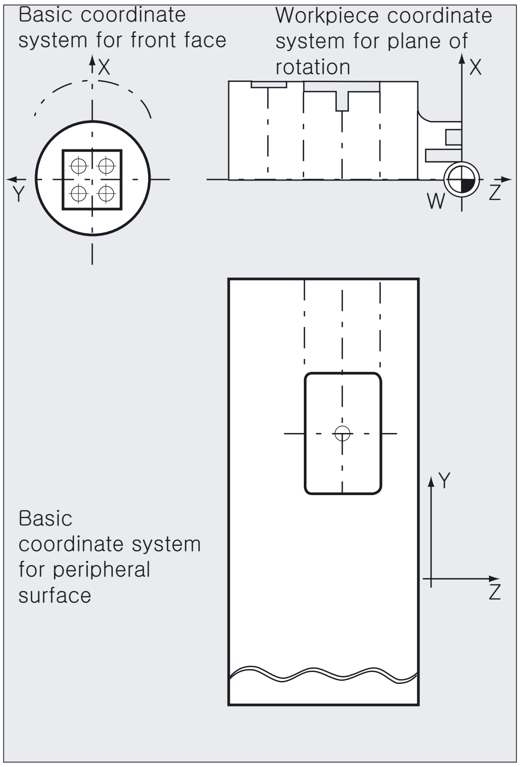
Determination from the right hand rule for different machine types

With different machine types the determination from the right hand rule can look different in each case. The following are examples of machine coordinate systems for various machines.



1.3.3 Basic coordinate system

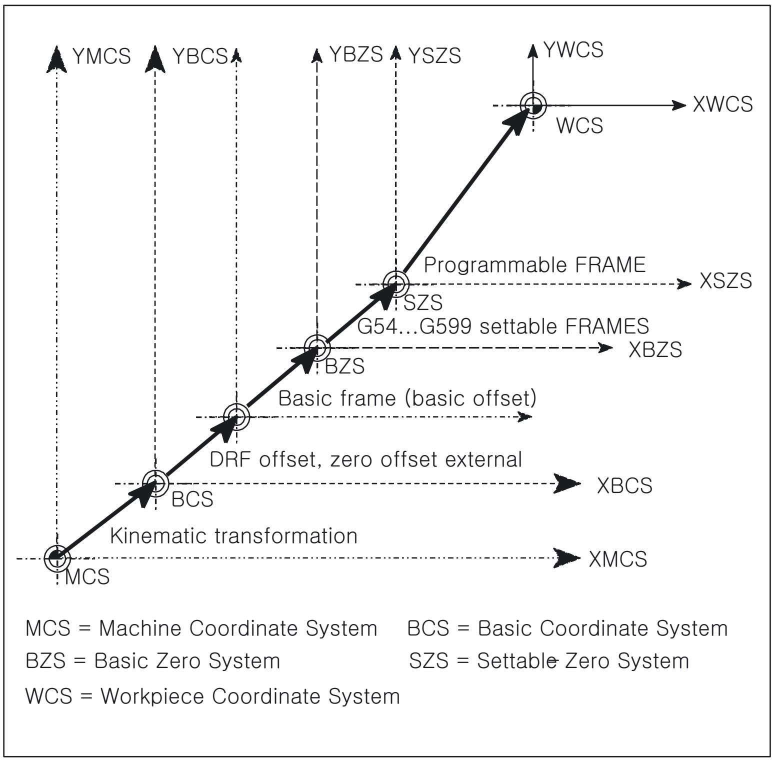
The basic coordinate system is a Cartesian coordinate system, which is mirrored by kinematic transformation (for example, 5-axis transformation or by using Transmit with peripheral surfaces) onto the machine coordinate system.


Basic coordinate system for front face
X
Y
Workpiece coordinate system for plane of rotation
X
W
Z
Y
Z
Basic coordinate system for peripheral surface

If there is no kinematic transformation, the basic coordinate system differs from the machine coordinate system only in terms of the axis designations.

The activation of a transformation can produce deviations in the parallel orientation of the axes. The coordinate system does not have to be at a right angle.

Further determinations

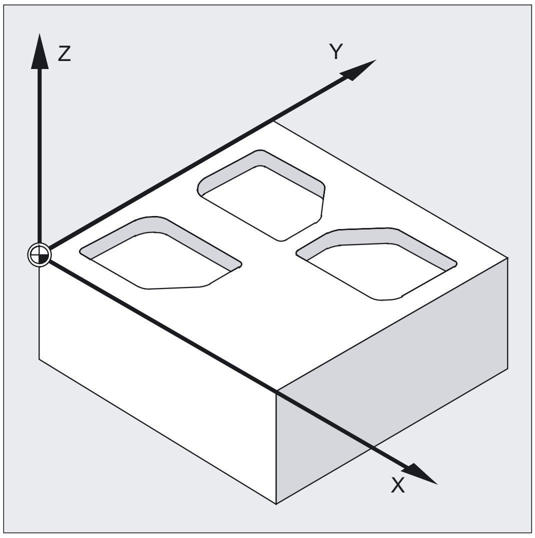

YMCS
YBCS
YBZS
YSZS
YWCS
XWCS
WCS
Programmable FRAME
XSZS
SZS
G54...G599 settable FRAMES
XBZS
BZS
Basic frame (basic offset)
DRF offset, zero offset external
XBCS
BCS
Kinematic transformation
XMCS
MCS
MCS = Machine Coordinate System
BCS = Basic Coordinate System
BZS = Basic Zero System
SZS = ~~Settable~~ Zero System
WCS = Workpiece Coordinate System

Zero offsets, scaling, etc., are always executed in the basic coordinate system.

The coordinates also refer to the basic coordinate system when specifying the working field limitation.

1.3.4 Workpiece coordinate system

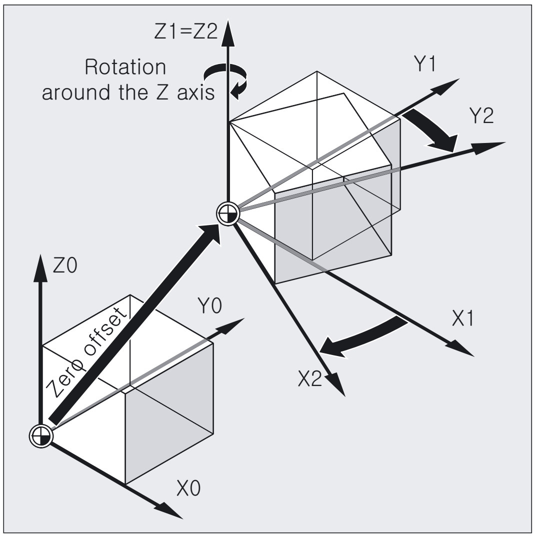
The geometry of a workpiece is described in the workpiece coordinate system. In other words, the data in the NC program refer to the workpiece coordinate system.



The workpiece coordinate system is always a Cartesian coordinate system and assigned to a specific workpiece.

1.3.5 Frame system

The frame is a self-contained arithmetic rule that transforms one Cartesian coordinate system into another Cartesian coordinate system.



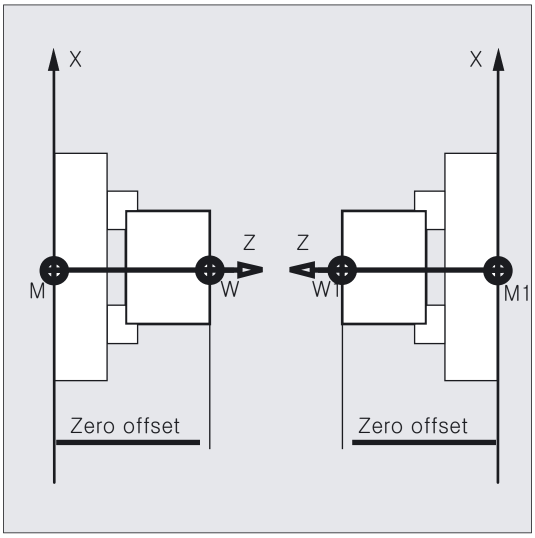
It is a spatial description of the workpiece coordinate system

The following components are available within a frame:

- Zero offset
- Rotate
- Mirroring
- Scaling

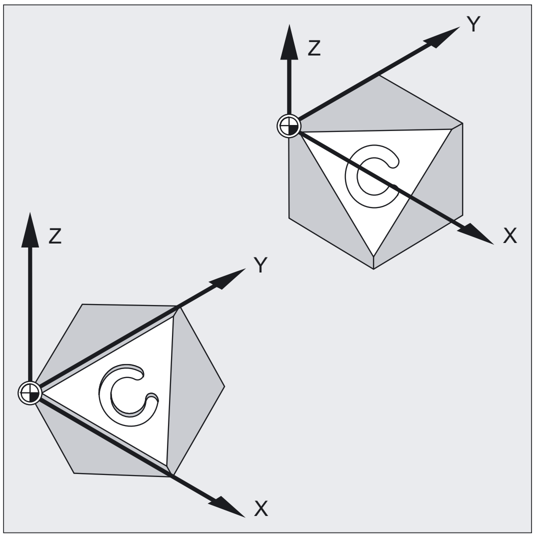
These components can be used individually or in any combination.

Mirroring of the Z axis



Shifting and turning the workpiece coordinate system

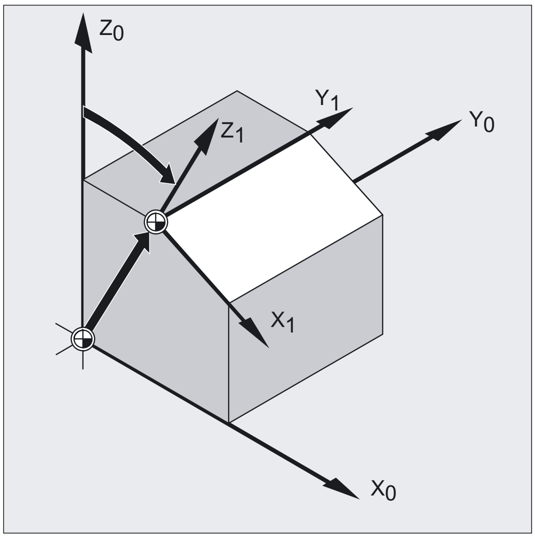
One way of machining inclined contours is to use appropriate fixtures to align the workpiece parallel to the machine axes.



... Another way is to generate a coordinate system, which is oriented to the workpiece. The coordinate system can be moved and/or rotated with programmable frames.

This enables you to

- move the zero point to any position on the workpiece.
- align the coordinate axes parallel to the desired working plane by rotation.
- and thus machine surfaces clamped in inclined positions, produce drill holes at different angles.
- Performing multi-side machining operations.

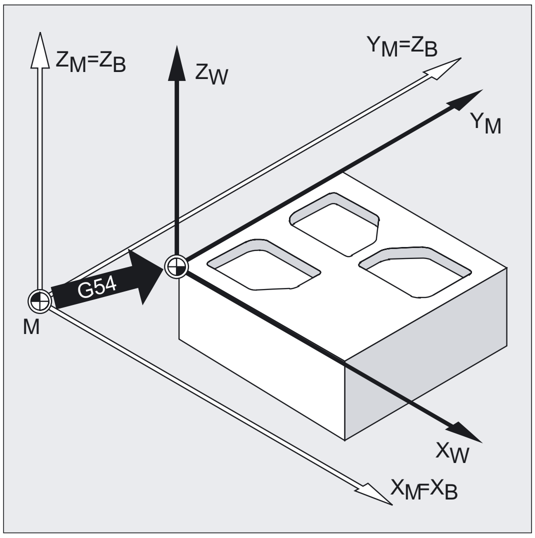


The conventions for the working plane and the tool offsets must be observed – in accordance with the machine kinematics – for machining operations in inclined working planes.

For further information, please see "Selection of working plane, G17 to G19".

1.3.6 Assignment of workpiece coordinate system to machine axes

The location of the workpiece coordinate system in relation to the basic coordinate system (or machine coordinate system) is determined by settable frames.

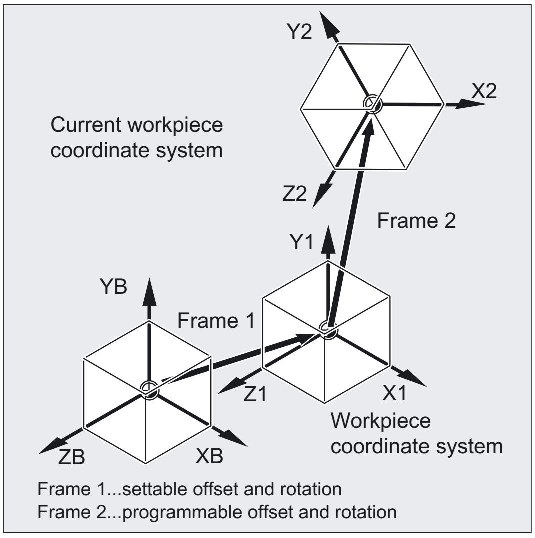


The settable frames are activated in the NC program by means of commands such as G54.

1.3.7 Current workpiece coordinate system

Sometimes it is advisable or necessary to reposition and to rotate, mirror and/or scale the originally selected workpiece coordinate system within a program.

The programmable frames can be used to reposition (rotate, mirror and/or scale) the current zero point at a suitable point in the workpiece coordinate system. You will thus obtain the current workpiece coordinate system.

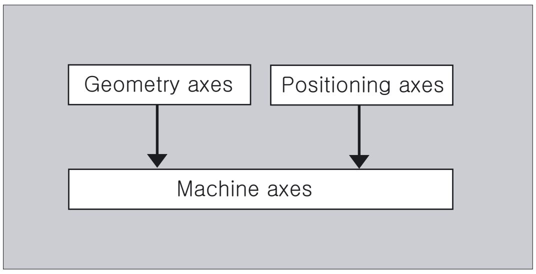


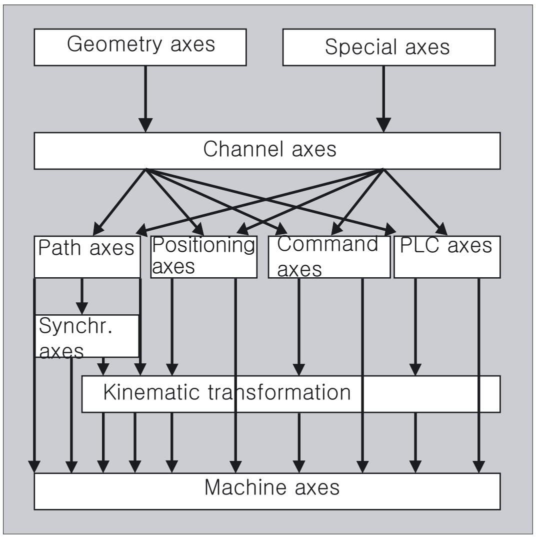
Several zero offsets are possible in the same program.

1.4 Axes

A distinction is made between the following types of axes when programming:

- Machine axes
- Channel axes
- Geometry axes
- Special axes
- Path axes
- Synchronized axes
- Positioning axes
- Command axes (motion-synchronous actions)
- PLC axes
- Link axes
- Lead link axes





Behavior of programmed axis types

Geometry, synchronized and positioning axes are programmed.

- Path axes traverse with feedrate F in accordance with the programmed travel commands.
- Synchronized axes traverse synchronously to path axes and take the same time to traverse as all path axes.
- Positioning axes traverse asynchronously to all other axes. These traversing movements take place independently of path and synchronized movements.
- Command axes traverse asynchronously to all other axes. These traversing movements take place independently of path and synchronized movements.
- PLC axes are controlled by the PLC and can traverse asynchronously to all other axes. The traversing movements take place independently of path and synchronized movements.

1.4.1 Main axes/Geometry axes

The main axes define a right-angled, right-handed coordinate system. Tool movements are programmed in this coordinate system.

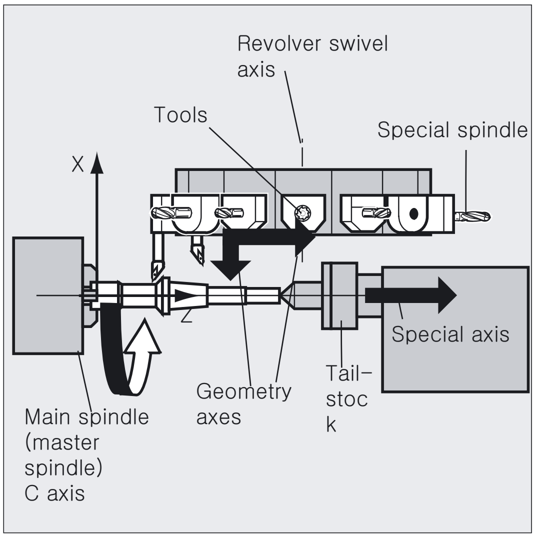
In NC technology, the main axes are called geometry axes. This term is also used in this Programming Guide.

The "Switchable geometry axes" function (see Advanced) can be used to alter the geometry axes grouping configured by machine data. Here any geometry axis can be replaced by a channel axis defined as a synchronous special axis.

Axis identifier

For turning machines:

Geometry axes X and Z are used, and sometimes Y.



For milling machines:

Geometry axes X, Y and Z are used.

A maximum of three geometry axes are used for programming frames and the workpiece geometry (contour).

The identifiers for geometry and channel axes may be the same, provided a reference is possible.

Geometry axis and channel axis names can be the same in any channel so that the same programs can be executed.

1.4.2 Special axes

In contrast to the geometry axes, no geometrical relationship is defined between the special axes.

Axis identifier

In a turning machine with revolver magazine, for example,

Turret position U, tailstock V

Application examples

Typical special axes are tool revolver axes, swivel table axes, swivel head axes, and loader axes.

```
N10 G1 X100 Y20 Z30 A40 F300                   ;Path axis movements
N20 POS[U]=10POS[X]=20 FA[U]=200 FA[X]=350     ;Positioning axis movements
N30 G1 X500 Y80 POS[U]=150FA[U]=300 F550       ;Path and positioning axis
N40 G74 X1=0 Z1=0                              ;Approaching a reference point
```

1.4.3 Main spindle, master spindle

The machine kinematics determine, which spindle is the main spindle. This spindle is declared the master spindle in the machine data. As a rule, the main spindle is declared the master spindle. This assignment can be changed with the program command SETMS (spindle number). By issuing SETMS without statement of the spindle number you can switch back to the master spindle defined in the machine data. Special functions such as thread cutting apply to the master spindle, see "Spindle speed S, spindle direction of rotation M3, M4, M5".

Spindle identifier

Identifiers: S or S0

1.4.4 Machine axes

Machine axes are the axes physically existing on a machine. The movements of axes can still be assigned by transformations (TRANSMIT, TRACYL, or TRAORI) to the machine axes. If transformations are intended for the machine, different axis names must be determined.

The machine axis names are programmed only in special cases, such as reference point or fixed point approaching.

Axis identifier

The axis identifiers can be set in the machine data.

Standard identifiers:

X1, Y1, Z1, A1, B1, C1, U1, V1

There are also standard axis identifiers that can always be used:

AX1, AX2, ..., AXn

1.4.5 Channel axes

Channel axes are all axes, which traverse in a channel.

Axis identifier

Identifiers: X, Y, Z, A, B, C, U, V

1.4.6 Path axes

Path axes define the path and therefore the movement of the tool in space.

The programmed feed is active for this path. The axes involved in this path reach their position at the same time. As a rule, these are the geometry axes.

However, default settings define, which axes are the path axes, and therefore determine the velocity.

Path axes can be specified in the NC program with FGROUP, see "Path behavior".

1.4.7 Positioning axes

Positioning axes are interpolated separately, i.e., each positioning axis has its own axis interpolator and its own feedrate. Positioning axes do not interpolate with the path axes.

Positioning axes are traversed by the NC program or the PLC. If an axis is to be traversed simultaneously by the NC program and the PLC, an error message appears.

Typical positioning axes are:

- Loaders for moving workpieces to machine
- Loaders for moving workpieces away from machine
- Tool magazine/turret

Programming

A distinction is made between positioning axes with synchronization at the block end or over several blocks.

Parameters

POS axes:

Block change occurs at the end of the block when all the path and positioning axes programmed in this block have reached their programmed end point.

POSA axes:

The movement of these positioning axes can extend over several blocks.

POSP axes:

The movement of these positioning axes for approaching the end position takes place in sections.

Note

Positioning axes become synchronized axes if they are traversed without the special POS/POSA identifier.

Continuous-path mode (G64) for path axes is only possible if the positioning axes (POS) reach their final position before the path axes.

Path axes that are programmed with POS/POSA are removed from the path axis grouping for the duration of this block.

You will find further information on POS, POSA, and POSP in the section on "Traversing positioning axes, POS, POSA, POSP".

1.4.8 Synchronized axes

Synchronized axes traverse synchronously to the path from the start position to the programmed end position.

The feedrate programmed in F applies to all the path axes programmed in the block, but does not apply to synchronized axes. Synchronized axes take the same time as the path axes to traverse.

A synchronized axis can be a rotary axis, which is traversed synchronously to the path interpolation.

1.4.9 Command axes

Command axes are started from synchronized actions in response to an event (command). They can be positioned, started, and stopped fully asynchronous to the parts program. An axis cannot be moved from the parts program and from synchronized actions simultaneously.

Command axes are interpolated separately, i.e., each command axis has its own axis interpolator and its own feedrate.

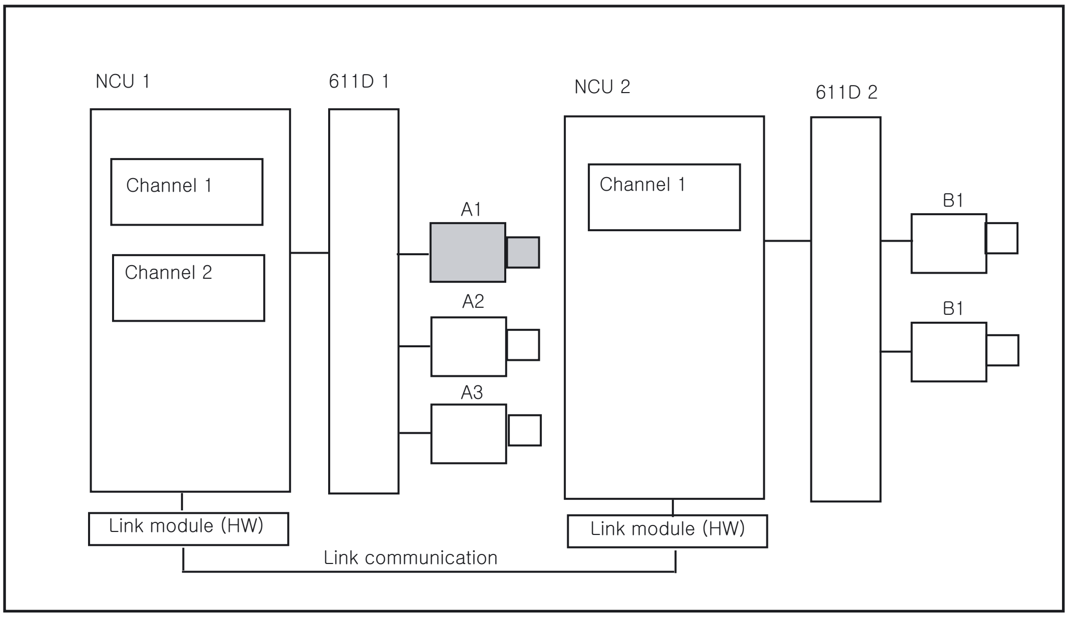
References: /FBSY/, Synchronized Actions

1.4.10 PLC axes

PLC axes are traversed by the PLC via special function blocks in the basic program; their movements can be asynchronous to all other axes. The traversing movements take place independently of path and synchronized movements.

1.4.11 Link axes

Link axes are axes, which are physically connected to another NCU and whose position is controlled from this NCU. Link axes can be assigned dynamically to channels of **another** NCU. Link axes are not local axes from the perspective of a particular NCU.



The **axis container** concept is used for the dynamic modification of the assignment to an NCU. Axis substitution with GET and RELEASE from the parts program is not available for link axes.

Precondition

The participating NCUs, NCU1 and NCU2, must be connected by means of high-speed communication via the link module.
References:
/PHD/, Configuring Manual NCU 571-573.2, Link Module

The axis must be configured appropriately by machine data.

The link axis option must be installed.

Description

The position control is implemented on the NCU on which the axis is physically connected to the drive. This NCU also contains the associated axis VDI interface. The position setpoints for link axes are generated on another NCU and communicated via the NCU link.

The link communication must provide the means of interaction between the interpolators and the position controller or PLC interface. The setpoints calculated by the interpolators must be transported to the position control loop on the home NCU and, vice versa, the actual values must be returned from there back to the interpolators.

For further information about link axes, please refer to

References: Function description /FB/ B3, Multiple Operator Panels and NCUs

Axis container

An axis container is a circular buffer data structure in which local axes and/or link axes are assigned to channels. The entries in the circular buffer can be **shifted cyclically**.

In addition to the direct reference to local axes or link axes, the link axis configuration in the logical machine axis image also allows references to axis containers. This type of reference consists of:

- a container number and
- a slot (circular buffer location within the container)

The entry in a circular buffer location contains:

- a local axis **or**
- a link axis

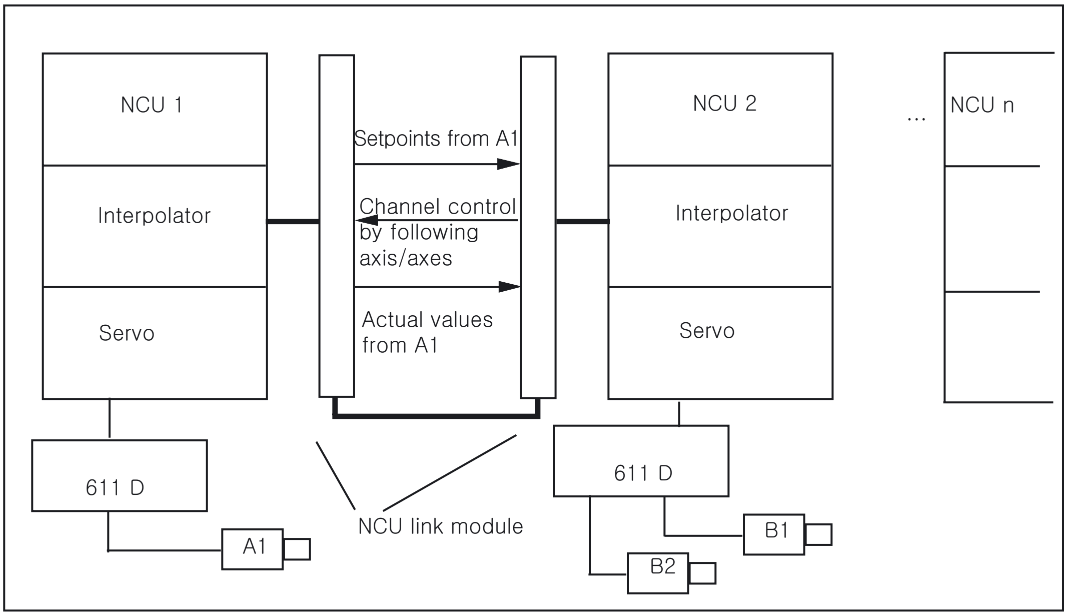
Axis container entries contain local machine axes or link axes from the perspective of an individual NCU. The entries in the logical machine axis image MN_AXCONF_LOGIC_MACHAX_TAB of an individual NCU are fixed.

The axis container function is described in

References: Function description /FB/ B3, Multiple Operator Panels and NCUs

1.4.12 Lead link axes

A leading link axis is one that is interpolated by one NCU and utilized by one or several other NCUs as the master axis for controlling slave axes.



An axial position controller alarm is sent to all other NCUs, which are connected to the affected axis via a leading link axis.

NCUs that are dependent on the leading link axis can utilize the following coupling relationships with it:

- Master value (setpoint, actual master value, simulated master value)
- Coupled motion
- Tangential correction
- Electronic gear (ELG)
- Synchronous spindle

Programming

Master NCU:

Only the NCU, which is physically assigned to the master value axis can program travel motions for this axis. The travel program must not contain any special functions or operations.

NCUs of slave axes:

The travel program on the NCUs of the slave axes must not contain any travel commands for the leading link axis (master value axis). Any violation of this rule triggers an alarm.

The leading link axis is addressed in the usual way via channel axis identifiers. The states of the leading link axis can be accessed by means of selected system variables.

Prerequisites

- The dependent NCUs, i.e., NCU1 to NCUn (n equals, max. of 8), must be interconnected via the link module for high-speed communication.
  References:
  /PHD/, Configuring Manual NCU 571-573.2, Link Module
- The axis must be configured appropriately by machine data.
- The link axis option must be installed.
- The same interpolation cycle must be configured for all NCUs connected to the leading link axis.

Restrictions

- A master axis, which is leading link axis cannot be a link axis, i.e., it cannot be operated by NCUs other than its home NCU.
- A master axis, which is leading link axis cannot be a container axis, i.e., it cannot be addressed alternately by different NCUs.
- A leading link axis cannot be the programmed leading axis in a gantry grouping.
- Couplings with leading link axes cannot be cascaded.
- Axis replacement can only be implemented within the home NCU of the leading link axis.

System variables:

The following system variables can be used in conjunction with the channel axis identifier of the leading link axis:

- $AA_LEAD_SP; Simulated master value position
- SAA_LEAD_SV; Simulated master value velocity

If these system variables are updated by the home NCU of the master axis, the new values are also transferred to any other NCUs, which wish to control slave axes as a function of this master axis.

References: /FB/ Function description B3, Multiple Operator Panels and NCUs

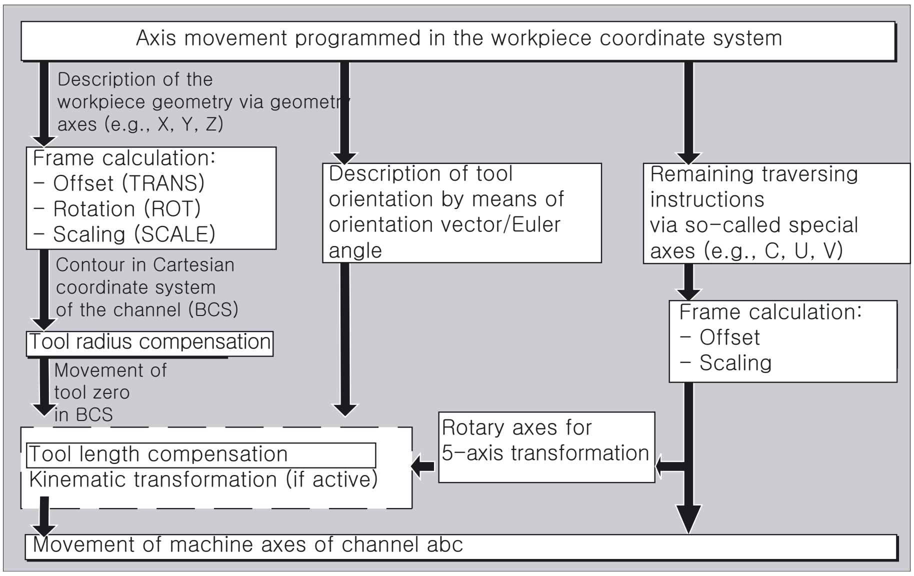
1.5 Coordinate systems and workpiece machining

The relationship between travel commands of the programmed axis movements from the workpiece coordinates and the resulting machine movement is displayed.

How you can determine the distance traveled taking into account all shifts and corrections is shown by reference to the path calculation.

Relationship between the travel commands from workpiece coordinates and the resulting machine movements

Axis movement programmed in the workpiece coordinate system

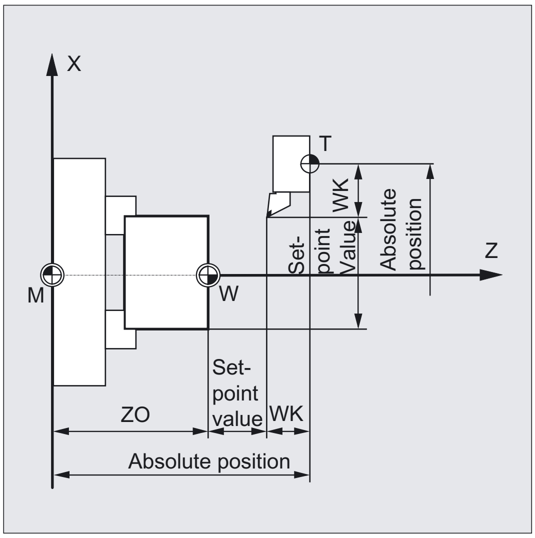

Axis movement programmed in the workpiece coordinate system
Description of the workpiece geometry via geometry axes (e.g., X, Y, Z)
Frame calculation:
– Offset (TRANS)
– Rotation (ROT)
– Scaling (SCALE)
Description of tool orientation by means of orientation vector/Euler angle
Remaining traversing instructions via so–called special axes (e.g., C, U, V)
Contour in Cartesian coordinate system of the channel (BCS)
Tool radius compensation
Frame calculation:
– Offset
– Scaling
Movement of tool zero in BCS
Rotary axes for 5–axis transformation
Tool length compensation
Kinematic transformation (if active)
Movement of machine axes of channel abc

Path calculation

The path calculation determines the distance to be traversed in a block, taking into account all offsets and compensations.

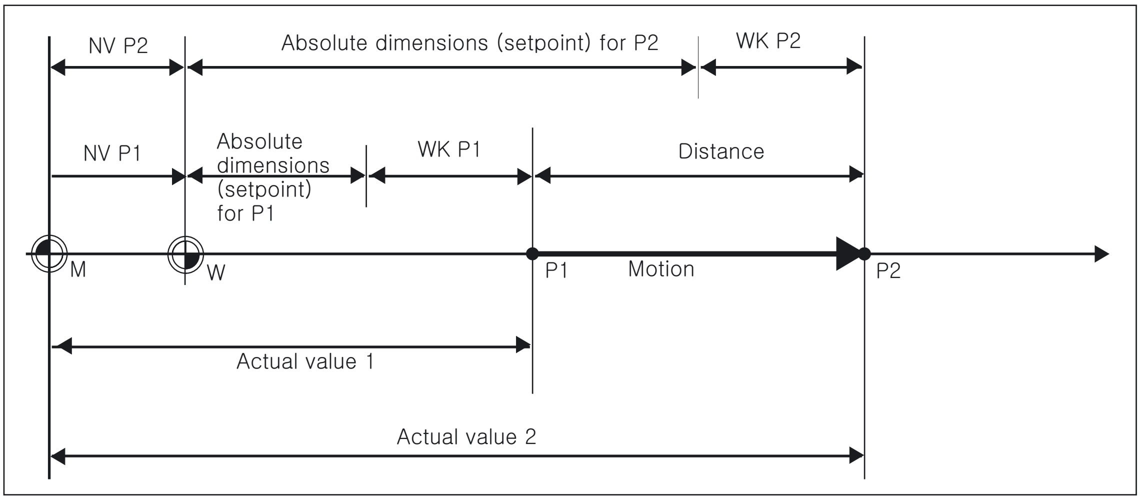
In general:

Distance = setpoint - actual value + zero offset (ZO) + tool offset (TO)



If a new zero offset and a new tool offset are programmed in a new program block, the following applies:

- With absolute dimensioning:
  Distance = (absolute dimension P2 - absolute dimension P1) + (ZO P2 - ZO P1) + (TO P2 - TO P1).
- With incremental dimensioning:
  Distance = incremental dimension + (ZO P2 - ZO P1) + (TO P2 - TO P1).


NV P2
Absolute dimensions (setpoint) for P2
WK P2
NV P1
Absolute dimensions (setpoint) for P1
WK P1
Distance
M
W
P1
Motion
P2
Actual value 1
Actual value 2

Fundamental Principles of NC Programming 2

2.1 Structure and contents of an NC program

Note

DIN 66025 is the guideline for designing a parts program.

An (NC/part) program consists of a sequence of NC **blocks** (see table below). Each data block represents one machining step. Instructions are written in the blocks in the form of **words.** The last block in the execution sequence contains a special word for the **end of program: M2**, **M17** or **M30.**

Set	Word	Word	Word	...	;Comment
Set	N10	G0	X20	...	;1. Set
Set	N20	G2	Z37	...	;2. Set
Set	N30	G91	...	...	;...
Set	N40	...	...	...	
Set	N50	M30	...	...	;End of program (last block)

Program names

Each program has a different name; the name can be chosen freely during program creation (except for punch tape format), taking the following conditions into account:

- The first two characters must be letters (or a letter with an underscore character)
- other letters, digits

Example:

_MPF100 or

SHAFT or

SHAFT_2

Only the first **24** characters of a program identifier are displayed on the NC.

Punch tape format

File names:

File names can contain the characters 0...9, A...Z, a...z or _ and must not exceed 24 characters in total.

File names must have a 3-character extension (_xxx).

Data in punch tape format can be generated externally or processed with an editor. A file name of a file that is filed internally in the NC memory starts with "_N_".
A file in punch tape format is introduced with %<name>, "%" must be in the first column of the first row.

Examples:

%_N_SHAFT123_MPF = part program SHAFT123

Or

%flange3_MPF = part program flange3

For further information on downloading, creating, and storing parts programs, please refer to:

/BAD/, /BEM/ Operating Instructions HMI Advanced, HMI Embedded section "Program operating area"/"Services operating area"

2.2 Language elements of the programming language

Overview

The language elements of the programming language are determined by

- Character set with uppercase and lowercase letters and digits
- Words with addresses and sequence of digits
- Blocks and block format
- Block length with maximum possible number of characters
- Order of the words in a block with table of the addresses and their meaning
- Main blocks and subblocks
- Block number
- Addresses with table for important addresses and explanations
- Addresses effective modally or non-modally
- Addresses with axial extension with table of extended address notations
- Fixed addresses with table and statement of the meaning for default setting
- Fixed addresses with axis expansion with table and statement of the meaning for default setting
- Adjustable addresses with statement of the adjustable address letters
- Predefined computing functions as well as arithmetic, comparative and logical operators with corresponding value assignments
- Identifiers such as variables, subroutines, keywords, DIN addresses and jump markers.

Character set

The following characters are available for writing NC programs:

Uppercase characters

A, B, C, D, E, F, G, H, I, J, K, L, M, N,(O), P, Q, R, S, T, U, V, W, X, Y, Z

Please note:

Take care to differentiate between the letter "O" and the digit "0".

Lowercase letters

a, b, c, d, e, f, g, h, i, j, k, l, m, n, o, p, q, r, s, t, u, v, w, x, y, z

Note

No distinction is made between upper and lower case letters.

Digits

1, 2, 3, 4, 5, 6, 7, 8, 9

Special characters

%	Program start character (used only for writing programs on an external PC)
(	For bracketing parameters or expressions
)	For bracketing parameters or expressions
[	For bracketing addresses or indexes
]	For bracketing addresses or indexes
<	Less than
>	Greater than
:	Main block, end of label, chain operator
=	Assignment, part of equation
/	Division, block suppression
*	Multiplication
+	Addition
-	Subtraction, minus sign
"	Double quotation marks, identifier for character string
'	Single quotation marks, identifier for special numerical values: hexadecimal, binary
$	System variable identifiers
_	Underscore, belonging to letters
?	Reserved
!	Reserved
.	Decimal point
,	Comma, parameter separator
;	Comment start
&	Format character, same effect as space character
LF	End of block
Tab character	Separator
space character	Separator (blank)

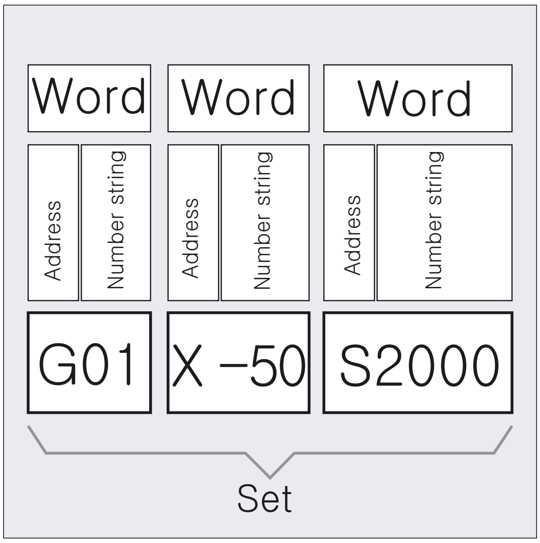
Note

Non-printable special characters are treated like blanks.

Words

In the same way as our language, NC programs are made up of blocks and each block is made up of words.

A word in the "NC language" consists of an address character and a digit or sequence of digits representing an arithmetic value.



The address character of the word is usually a letter. The sequence of digits can contain a leading sign and decimal point. The leading sign always appears between the address letter and the sequence of digits. The positive leading sign (+) does not have to be specified.

Blocks and block format

An NC program consists of individual blocks. A block generally consists of (several) words.

A block should contain all the data required for performing an operation step and is terminated with the character "L_F" (LINE FEED = new line).

Note

The "L_F" character does not have to be inserted manually, it is generated automatically when you change lines.

Block length

A block can contain a maximum of **512 characters** (including the comment and end-of-block character "L_F").

Note

Three blocks of up to 66 characters each are normally displayed in the current block display on the screen. Comments are also displayed. Messages are displayed in a separate message window.

Word sequence in blocks

In order to keep the block format as clear as possible, the words in a block should be arranged as follows:

Example:

N10 G… X… Y… Z… F… S… T… D… M… H…

Address	Meaning
N	Address of block number
10	Block number
G	Preparatory function
X,Y,Z	Positional data
F	Feed
S	Spindle speed
T	Tool
D	Tool offset number
M	Miscellaneous (i.e., special) function
H	Auxiliary function

Note

Certain addresses can be used repeatedly within a block
(e.g., G…, M…, H…)

Main block/subblock

There are two types of blocks:

- Main blocks and
- subblocks

The main block must contain all the words necessary to start the operation sequence in the program section beginning with the main block.

Note

Main blocks can be contained in both main programs and subprograms. The control does not check whether a main block contains all the necessary information.

Block number

Main blocks are identified by a main block number. A main block number comprises the character ":" and a positive whole number (block number). The block number always appears at the start of a block.

Note

Main block numbers must be unique within a program to achieve an unambiguous result when searching.

Example:

:10 D2 F200 S900 M3

Subblocks are identified by a subblock number. A subblock number comprises the character "N" and a positive whole number (block number). The block number always appears at the start of a block.

Example:

N20 G1 X14 Y35

N30 X20 Y40

Note

Subblock numbers must be unique within a program in order to achieve an unambiguous result when searching.

The order of the block numbers is arbitrary, however increasing block numbers are recommended. You can also program NC blocks without block numbers.

Addresses

Addresses are fixed or settable identifiers for axes (X, Y, etc.), spindle speed (S), feedrate (F), circle radius (CR), etc.

Example:

N10 X100

Important addresses

Address	Meaning (default setting)	Notes

A=DC(...)
A=ACP(...)
A=ACN(...)
	Rotary axis	variable

ADIS	Rounding clearance for path functions	fixed

B=DC(...)
B=ACP(...)
B=ACN(...)
	Rotary axis	variable

C=DC(...)
C=ACP(...)
C=ACN(...)
	Rotary axis	variable

CHR=...	Chamfer the contour corner	fixed
D...	Cutting edge number	fixed
F...	Feed	fixed

FA[axis]=... or
FA[spindle]=... or
[SPI(spindle)]=...
	Axial feed
	(only if spindle no. defined by variable)
		fixed

G...	Preparatory function	fixed
H... H=QU(...)	Auxiliary function Auxiliary function without read stop	fixed
I...	Interpolation parameters	variable
J...	Interpolation parameters	variable
K...	Interpolation parameters	variable
L...	Subprogram call	fixed
M... M=QU(...)	Miscellaneous (i.e., special) function Miscellaneous fct. w/o read stop	fixed
N...	Subblock	fixed
OVR=...	Path override	fixed
P...	Number of program passes	fixed
POS[Axis]=...	Position axis	fixed
POSA[Axis]=...	Positioning axis across block boundary	fixed

SPOS=...
SPOS[n]=...
	Spindle position	fixed

SPOSA=...
SPOSA[n]=...
	Spindle position across block boundary	fixed

Q...	Axis	variable
R0=... to Rn=... R...	- Arithmetic parameter, n can be set via MD (default 0-99) - Axis	fixed variable
RND	Round the contour corner	fixed
RNDM	Round contour corner (modally)	fixed
S...	Spindle speed	fixed
T...	Tool number	fixed
U...	Axis	variable
V...	Axis	variable
W...	Axis	variable

X...
X=AC(...)
X=IC(...)
	Axis
	" absolute
	" incremental
		variable

Y... Y=AC(...)	Axis	variable

Y=IC(...)		

Z...
Z=AC(...)
Z=IC(...)
	Axis	variable

AR+=...	Opening angle	variable
AP=...	Polar angle	variable
CR=...	Circle radius	variable
RP=...	Polar radius	variable
:...	Main block	fixed

"fixed"

These address names are available for a specific function.

Machine manufacturer

"variable"

The machine manufacturer may assign another name to these addresses via machine data.

Modal/non-modal addresses

Modal addresses remain valid with the programmed value (in all subsequent blocks) until a new value is programmed at the same address.

Non-modal addresses only apply in the block, in which they were programmed.
Example:

```
N10 G01 F500 X10
N20 X10              ;Feedrate remains operative until a new feed value is entered
```

Addresses with axial extension

In addresses with axial extension, an axis name is inserted in square brackets after the address. The axis name assigns the axis.

Example:

```
FA[U]=400            ;Axis-specific feed for U axis
```

Extended addresses

Extended address notation enables a larger number of axes and spindles to be organized in a system. An extended address is composed of a numeric extension or a variable identifier enclosed in square brackets and an arithmetic expression with an "=" sign.

Example:

```
X7                   ;No "=" required, 7 is a value, but the "=" character can
                     ;also be used here
X4=20                ;Axis X4 ("=" required)
CR=7.3               ;2 letters ("=" required)
S1=470               ;Speed for 1st spindle 470 rpm
M3=5                 ;Spindle stop for 3rd spindle
```

The extended address notation is only permitted for the following direct addresses:

Address	Meaning
X, Y, Z, ...	Axis addresses
I, J, K	Interpolation parameters
S	Spindle speed
SPOS, SPOSA	Spindle position
M	Miscellaneous functions
H	Auxiliary functions
T	Tool number
F	Feed

In the case of extended address notation, the number (index) can be substituted by a variable for addresses M, H and S and for SPOS and SPOSA. The variable identifier is enclosed in square brackets.

Example:

```
S[SPINU]=470          ;Speed for the spindle, whose number is stored in the
                      ;SPINU variables.
M[SPINU]=3            ;Clockwise rotation for the spindle, whose number is stored in
                      ;the SPINU variables.
T[SPINU]=7            ;Selection of the tool for the spindle, whose number is stored
                      ;in the SPINU variables.
```

Fixed addresses

The following addresses are set permanently:

Address	Meaning (default setting)
D	Cutting edge number
F	Feed
G	Preparatory function
H	Auxiliary function
L	Subprogram call
M	Miscellaneous (i.e., special) function
N	Subblock
P	Number of program runs
R	Arithmetic variables
S	Spindle speed
T	Tool number
:	Main block

Example for programming:

N10 G54 T9 D2

Fixed addresses with axis expansion

Address	Meaning (default setting)
AX	Axis value (variable axis programming)
ACC	Axial acceleration
FA	Axial feed
FDA	Axis feed for handwheel override
FL	Axial feed limit
IP	Interpolation parameter (variable axis programming)
OVRA	Axial override
PO	Polynomial coefficient
POS	Position axis
POSA	Positioning axis across block boundary

Example:

```
N10 POS[X]=100
```

Explanation:

When programming with the axis expansion, the axis to be traversed is enclosed in square brackets.

You will find a complete list of all fixed addresses in the Appendix.

Settable addresses

Addresses can be defined either as an address letter (with numerical extension if necessary) or as freely selected identifiers.

Note

Variable addresses must be unique within the control, i.e., the same identifier name may not be used for different address types.

A distinction is made between the following address types:

- Axis values and end points
- Interpolation parameters
- Feedrates
- Approximate positioning criteria
- Measurement
- Axis, spindle behavior
- …

Variable address letters are:

A, B, C, E, I, J, K, Q, U, V, W, X, Y, Z

Note

The user can change the names of the variable addresses in the machine data.

Example:

X1, Y30, U2, I25, E25, E1=90, …

The numeric extension has one or two digits and is always positive.

Address identifiers:

The address notation can be expanded by adding extra letters.

Example:

```
CR                    ;e.g., for circle radius
XPOS
```

Operators/mathematical functions

Operators and mathematical functions	Meaning
+	Addition
-	Subtraction
*	Multiplication
/	Division **Notice**: (type INT)/(type INT)=(type REAL); e.g., 3/4 = 0.75
DIV	Division, for variable types INT and REAL **Notice**: (type INT)DIV(type INT)=(typ INT); e.g., 3 DIV 4 = 0
MOD	Modulo division (INT type only) produces the remainder of INT division, e.g., 3 MOD 4=3
:	Chain operator (for FRAME variables)
Sin()	Sine
COS()	Cosine
TAN()	Tangent
ASIN()	Arcsine
ACOS()	Arccosine
ATAN2()	Arctangent2
SQRT()	Square root
ABS()	Absolute number
POT()	2nd power (square)
TRUNC()	Truncate to integer
ROUND()	Round to integer
LN()	Natural logarithm
EXP()	Exponential function

Comparison and logic operators

Comparison and logic operators	Meaning
==	Equal to
<>	Not equal to
>	Greater than
<	Less than
>=	Greater than or equal to
<=	Less than or equal to
AND	AND
OR	OR
NOT	Negation
XOR	Exclusive OR

In arithmetic expressions, the execution order of all the operators can be specified by parentheses, in order to override the normal priority rules.

Value assignments

Values can be assigned to the addresses. The method of value assignment depends on the type of address identifier.

An "=" -sign must be inserted between the address identifier and the value if

- The address identifier comprises more than one letter,
- The value includes more than one constant.

The "="-sign can be omitted if the address identifier is a single letter and the value consists of only one constant. Leading signs are allowed and separators are permitted after the address letter.

Example of value assignments

```
X10                  ;Value assignment (10) to address X, "=" not required
X1=10                ;Value assignment (10) to address (X) with
                     ;numeric extension (1), "=" required
FGROUP(X1, Y2)       ;Axis names from passed parameters
AXDATA[X1]           ;Axis name as an index when accessing axis data
AX[X1]=10            ;Indirect axis programming
X=10*(5+SIN(37.5)    ;Value assignment by means of a numeric expression
)                    ;"=" required
```

Note

A numeric extension must always be followed by one of the special characters "=", "(", "[", ")", "]", ",", or an operator, in order to distinguish an address name with numeric extension from an address letter with a value.

Names

Identifiers can also be used to describe words (in compliance with DIN 66025). The identifiers have the same meaning as the words within an NC block. Identifiers must be unique. The same identifier must not be used for different objects.

Identifiers can stand for:

- Variable
  - System variable
  - User variable
- Subprograms
- Vocabulary words
- DIN addresses with several letters
- Jump markers

Design

The identifiers are composed of up to 32 characters. The following characters may be used:

- Letters
- Underscore symbols
- Digits

The first two characters must be letters or underscores, separators must not be programmed between the individual characters (see the following pages).

Example:

CMIRROR, CDON

Note

Reserved vocabulary words must not be used as identifiers. Separators are not permitted between the individual characters.

Note

Number of characters for each identifier

- Program names: 24 characters
- Axis identifiers: 8 characters
- Variable identifiers: 31 characters

Rules for allocating identifiers

The following rules are provided in order to avoid identifier collisions:

- All identifiers beginning with "CYCLE" or "_" are reserved for SIEMENS cycles.
- All identifiers beginning with "CCS" are reserved for SIEMENS compile cycles.
- User compile cycles begin with "CC".
- We recommend that users select identifier names, which either begin with "U" (User) or contain the underscore symbol, because these are not used by the system, compile cycles or SIEMENS cycles.

Further reserved identifiers

- The identifier "RL" is reserved for conventional turning machines.
- All identifiers beginning with "E_ " are reserved for EASYSTEP programming.

Variable identifiers

In variables used by the system, the first letter is replaced by the "$" character. This character may not be used for user-defined variables.

Examples (see "List of system variables"):

$P_IFRAME, $P_F

Leading zeroes are ignored in variables with numeric extensions (i.e., R01 is interpreted as R1). Separators are allowed before a numeric extension.

Array identifiers

The rules for elementary variables also apply to array identifiers. It is possible to address arithmetic variables as arrays.

Example:

R[10]=…

Data types

A variable can contain a numeric value (or several) or a character (or several), e.g., an address letter.

The data type permitted for the variable is determined when the variable is defined. The data type for system variables and predefined variables is fixed.

Elementary variable types/data types are:

Type	Meaning	Range of values
INT	Integers with leading sign	±(231 - 1)
REAL	Real numbers (fractions with decimal point, LONG REAL to IEEE)	±(10-300 … 10+300)
BOOL	Boolean values: TRUE (1) and FALSE (0)	1, 0
CHAR	ASCII character specified by the code	0 … 255
STRING	Character string, number of characters in [...], maximum of 200 characters	Sequence of values with 0 ... 255
AXIS	Axis names (axis addresses) only	Any axis identifiers in the channel
FRAME	Geometrical parameters for translation, rotation, scaling, and mirroring	

Identical elementary types can be combined in arrays. Up to two-dimensional arrays are possible.

Constants

Integer constants

Integer with or without leading sign, e.g., for assigning a value to an address

Examples:

```
X10.25              ;Assignment of the value +10.25 to address X
X -10.25            ;Assignment of the value -.25 to address X
X0.25               ;Assignment of the value +0.25 to address X
X.25                ;Assignment of the value +0.25 to address X without leading "0"
X=-.1EX-3           ;Assignment of the value -.1*10-3 to address X
```

Note

If, in an address, which permits decimal point input, more decimal places are specified than actually provided for the address, then they are rounded to fit the number of places provided.

X0 cannot be replaced with X.

Example:

Do not replace G01 X0 with G01 X!

Hexadecimal constants

Constants can also be interpreted in hexadecimal format. The letters "A" to "F" stand for the digits 10 to 15.

Hexadecimal constants are enclosed in single quotation marks and start with the letter "H", followed by the value in hexadecimal notation. Separators are allowed between the letters and digits.

Example for machine data (see also "Programming Guide Advanced"):

```
$MC_TOOL_MANAGEMENT_MASK='H3C7F'          ;Assignment of hexadecimal values to
                                          ;machine data
```

The maximum number of characters is limited by the value range of the integer data type.

Binary constants

Constants can also be interpreted in binary format. In this case, only the digits "0" and "1" are used.

Binary constants are enclosed in single quotation marks and start with the letter "B", followed by the binary value. Separators are allowed between the digits.

Example for machine data (see also "Programming Guide Advanced"):

```
$MN_AUXFU_GROUP_SPEC='B10000001'          ;Assignment of binary constants to
                                          ;machine data bit 0 and 7 are set
```

The maximum number of characters is limited by the value range of the integer data type.

Program section

A program section consists of a main block and several subblocks.

Examples:

:10 D2 F200 S900 M3

N20 G1 X14 Y35

N30 X20 Y40

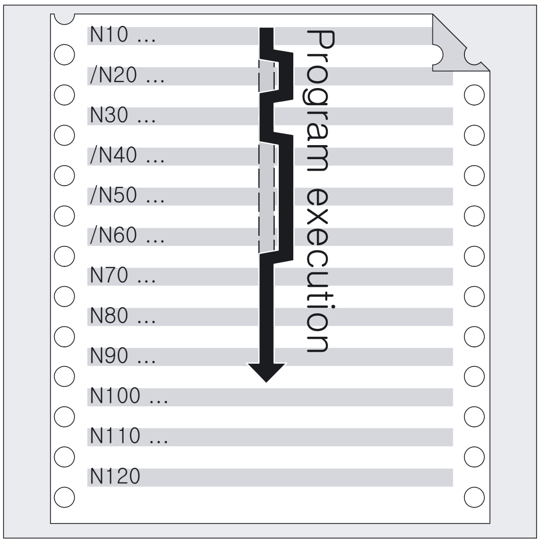
N40 Y-10

...

N100 M30

Skipping blocks

Blocks, which are not to be executed in every program pass (e.g., execute a trial program run), can be skipped.



Blocks, which are to be skipped are marked with an oblique "/" in front of the block number. Several consecutive blocks can also be skipped. The instructions in the skipped blocks are not executed; the program continues with the next block, which is not skipped.

Example of skipping blocks

```
N10                               ;Is executed
/N20 …                            ;Skipped
N30 …                             ;Is executed
/N40 …                            ;Skipped
N70 …                             ;Is executed
```

Up to 10 skip levels can be programmed. Only one skip level can be specified per NC block:

```
/ ...                             ;Block is skipped (1st skip level)
/0 ...                            ;Block is skipped (1st skip level)
/1 N010...                        ;Block is skipped (2nd skip level)
/2 N020...                        ;Block is skipped (3rd skip level)
...
/7 N100...                        ;Block is skipped (8th skip level)
/8 N080...                        ;Block is skipped (9th skip level)
/9 N090...                        ;Block is skipped (10th skip level)
```

Machine manufacturer

The number of skip levels that can be used depends on a display machine datum.

Block skipping of levels /0 to /9 is activated by an operator action (see /BA/ Operator's Guide HMI Advanced Embedded, program control menu in Machine operating area) or by the programmable controller.

Note

System and user variables can also be used in conditional jumps in order to control program execution.

Jump destinations (labels)

Labels can be defined to jump within a program.

Label names are allocated with at least two and up to 32 characters (letters, digits, underscore). The first two characters must be letters or underscores. The label name is followed by a colon (":").

For further information refer to

References: /PGA/, Programming Guide Advanced.

Note

Labels must be unique within a program.

Labels always appear at the start of a block. If a program number exists, the label appears immediately after the block number.

Comments

To make NC programs easier to understand for other users and programmers, it is advisable to insert meaningful comments in the program.

Comments are appended to the end of a block and are separated from the program section of the NC block by a semicolon (";").

Examples of comments

```
N10 G1 F100 X10 Y20           ;Comments to explain the NC block
Or
N10                           ;Company G&S, order no. 12A71
N20                           ;Program written by H. Müller, Dept. TV 4
                              ;on November 21, 1994
N50                           ;Section no. 12, housing for submersible pump
                              ;type TP23A
```

Note

Comments are stored and appear in the current block display when the program is running.

Programming messages

Messages can be programmed to provide the user with information about the current machining situation during program execution.

A message in an NC program is generated when the message text is typed after vocabulary word "MSG" in round parentheses "()" and double quotation marks.
A message can be deleted using "MSG ()".

Example of activating/deleting messages

```
N10 MSG ("Roughing the contour")  ;Activate message
N20 X… Y…
N …
N90 MSG ()                        ;Clear message from N10
```

Note

A message text can be up to 124 characters long and is displayed in two lines (2*62 characters). Contents of variables can also be displayed in message texts.

Example of message texts

```
N10 R12=$AA_IW [X]                ;Current position of the X axis in R12
N20 MSG (″Check position of X axis″<<R12<<)
N …
N90 MSG ()                        ;Clear message from N20
Or
N20 MSG (″Check position of X axis″<<$AA_IW[X]<<)
```

Setting alarms

You can also set alarms in addition to messages in an NC program. Alarms are displayed in a separate field on the screen display. An alarm always goes hand in hand with a response from the controller according to the alarm category.

Alarms are programmed by writing the vocabulary word "SETAL" followed by the alarm number enclosed in round brackets.

The valid range for alarm numbers is between 60,000 and 69,999, whereby 60,000 to 64,999 are reserved for SIEMENS cycles and 65,000 to 69,999 are available to the user.

Note

Alarms are always programmed in a separate block.

Example:

```
N100 SETAL (65000)                ;Set alarm no. 65000
```

You will find a list of reactions associated with specific alarms in the Installation and Start-up Guide.

The alarm text must be configured in the HMI.

Programmable cycle alarms

A character string containing up to 4 parameters can be specified in addition to the alarm number for the predefined subprogram SETAL.

Programming

SETAL(<alarmnumber>, <string>)

Parameters

Variable user texts can be defined in these parameters. Predefined parameters with the following meaning are also provided:

%1 =	Channel number
%2 =	Block number, label
%3 =	Text index for cycle alarms
%4 =	additional alarm parameters

2.3 Programming a sample workpiece

The programming of the individual operation steps in the NC language generally represents only a small proportion of the work in the development of an NC program.

Programming of the actual instructions should be preceded by the planning and preparation of the operation steps. The more accurately you plan in advance how the NC program is to be structured and organized, the faster and easier it will be to produce a complete program, which is clear and free of errors.

Programming

Clearly structured programs are a particular advantage if you need to make changes at a later date.

Since workpieces differ in shape and form, it is not advisable to create every program using exactly the same method. There are certain methods, which have proven to be successful in most instances. A sort of "checklist" can be found below.

Procedures

- Prepare the workpiece drawing
  - define the workpiece zero
  - Draw in the coordinate system
  - Calculate any missing coordinates
- Define machining sequence
  - Which tools are used when and to machine which type of contour?
  - In what order are the individual elements of the workpiece machined?
  - Which individual elements repeat (possibly rotated) and should therefore be included in a subprogram?
  - Can you use part contours or similar elements, which already exist in other subprograms or subroutines?
    Where is it advisable or necessary to perform zero offset, rotation, mirroring or scaling (frame concept)?
- Create a machining plan

  Define all the machining processes in steps, e.g.:
  - Rapid traverse motions for positioning
  - Tool change
  - Retract to tool change point
  - Activate/deactivate spindle, coolant
  - Call tool data
  - Infeed
  - Path override
  - Approach contour
  - Retraction from the contour
  - etc.
- Translate the work steps into the programming language
  - Enter each individual step in an NC block or blocks.
- Combine all the individual steps in a program

2.4 First programming example for milling application

Testing first programming steps on the NC

Please proceed on the NC as described below to verify the following programming example:

- Create a new parts program (name)
- Edit the parts program
- Select the parts program
- Activate single block
- Start the parts program

References: See Operator's Guide

Note

Alarms can occur during program verification. These alarms have to be reset first.

Machine manufacturer

The machine data settings must be defined correctly before the program can run on the machine.

References: /FB1/ Functional description, K2, "Axes, Coordinate Systems,..."

Programming example

_MILL1_MPF

```
N10 MSG("THIS IS MY NC PROGRAM")  ;MSG = Message output in an alarm line
:10 F200 S900 T1 D2 M3            ;Feed, spindle, tool,
                                  ;tool offset, spindle clockwise
N20 G0 X100 Y100                  ;Rapid traverse to position
N30 G1 X150                       ;Rectangle with feed, straight line in X
N40 Y120                          ;Straight line in Y
N50 X100                          ;Straight line in X
N60 Y100                          ;Straight line in Y
N70 G0 X0 Y0                      ;Return rapid traverse movement
N100 M30                          ;End of block
```

2.5 Second programming example for milling application

Programming a sample workpiece

This programming example contains surface and side milling, as well as drilling.

- The workpiece is intended for machining on a **vertical milling machine**.
- The dimensions are in inches.

Machine manufacturer

The machine data settings must be defined correctly before the program can run on the machine.

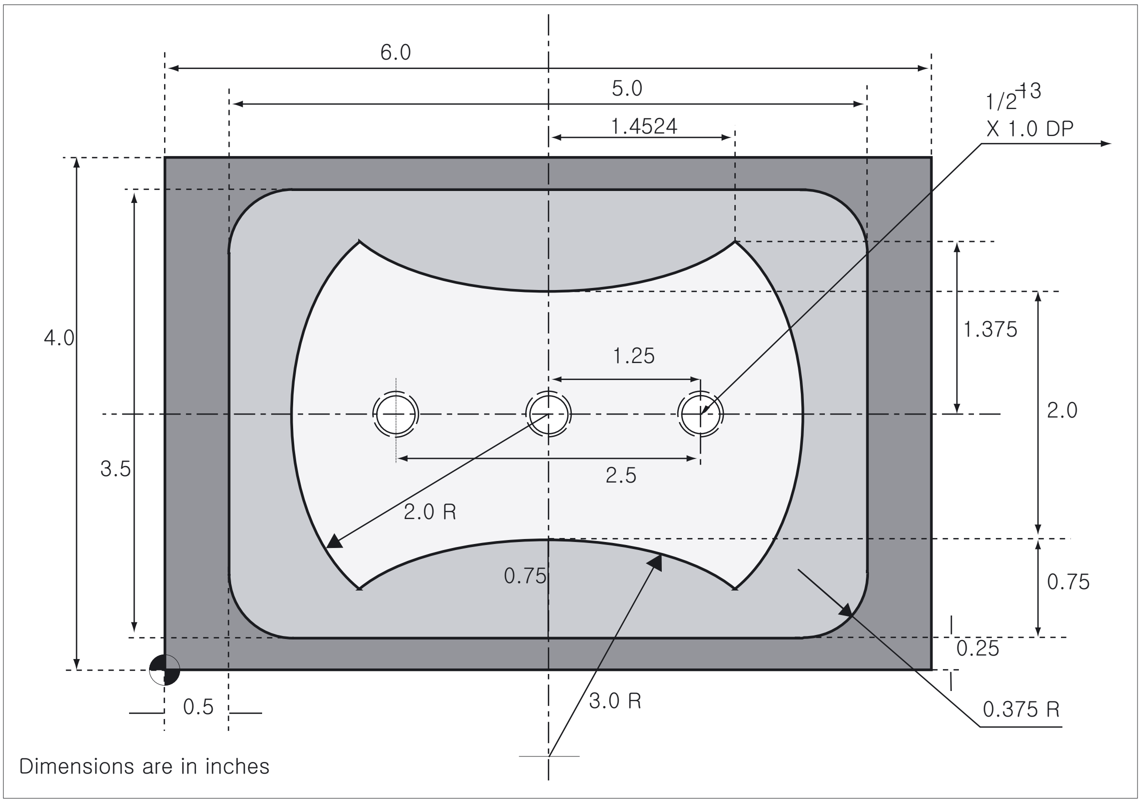
References: /FB1/Functional description, K2, "Axes, Coordinate Systems,..."

Example

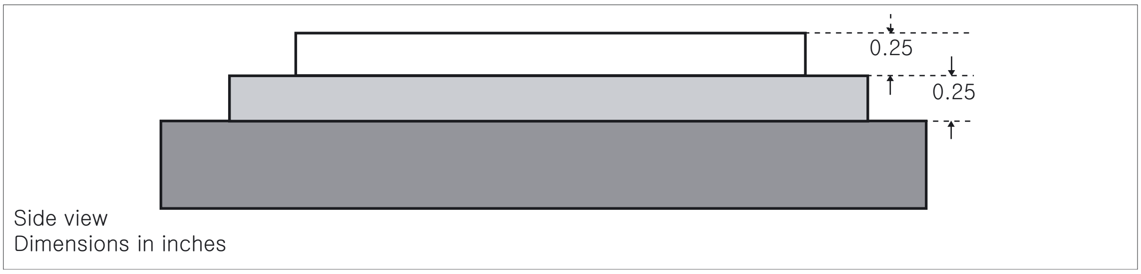
%_N_RAISED_BOSS_MPF

```
N005 MSG ("Traverse axes to tool change location")
N010 START01: SUPA G0 G70 Z0 D0
N015 SUPA X0 Y0
;********************Tool change********************
N020 MSG ("Tool change active")
N025 T1 M6                        ;d = 3 inch face cutter
N030 MSG ()                       ;Clears the message from block N020
N035 MSG ("Face milling Z=0 workpiece surface")
N040 G0 G54 X-2 Y.6 S800 M3 M8
N045 Z1 D1
N050 G1 Z0 F50
N055 X8 F25
N060 G0 Y3.5
N065 G1 X-2
N070 SUPA G0 Z0 D0 M5 M9
;********************Tool change********************
N075 T2 M6                        ;d = 1 inch facing tool
MSG ("Side machining")
N080 G0 X-1 Y.25 S1200 M3 M8
N085 Z1 D1
N090 G1 Z-.5 F50
N095 G42 X.5 F30
N100 X5.5 RNDM=-.375              ;Modal rounding. Radius=0.375
N105 Y3.625
N110 X.5
N115 Y.25
N120 X=IC(.375) RNDM=0            ;Needed for edge rounding
N125 G40 G0 Y-1 M5 M9             ;Rapid traverse to initial setting
N130 Z1
N135 X-1 Y0
N140 Z-.25
,********************Continue to use 1-inch mill****************
MSG ("Side Cut Top Boss")
N145 G01 G41 X1 Y2
N150 G2 X1.5476 Y3.375 CR=2
N155 G3 X4.4524 CR=3
N160 G2 Y.625 CR=2
N165 G3 X1.5476 CR=3
N170 G2 X1 Y2 CR=2
N175 G0 G40 X0
```

```
N180 SUPA G0 Z0 D0 M5 M9          ;Z approaches tool change position
N185 SUPA X0 Y0                   ;X and Y to tool change position
;*******************Tool change*******************
N190 T3 M6                        ;27/64 drill
MSG ("Drill 3 holes")
N195 G0 X1.75 Y2 S1500 M3 M8      ;Approach first drill hole
N200 Z1 D1
N205 MCALL CYCLE81 (1,0,.1,-.5,)
N207 X1.75                        ;Drill first hole
N210 X3                           ;Drill second hole
N215 X4.25                        ;Drill third hole
N220 MCALL
N221 SUPA Z0 D0 M5 M9             ;Delete modal call. Z axis traverses to machine
                                  ;zero
N225 SUPA X0 Y0
MSG ()
N230 M30                          ;End of program
```



Dimension drawing of workpiece "The Raised Boss" (not to scale).



2.6 Programming example for turning application

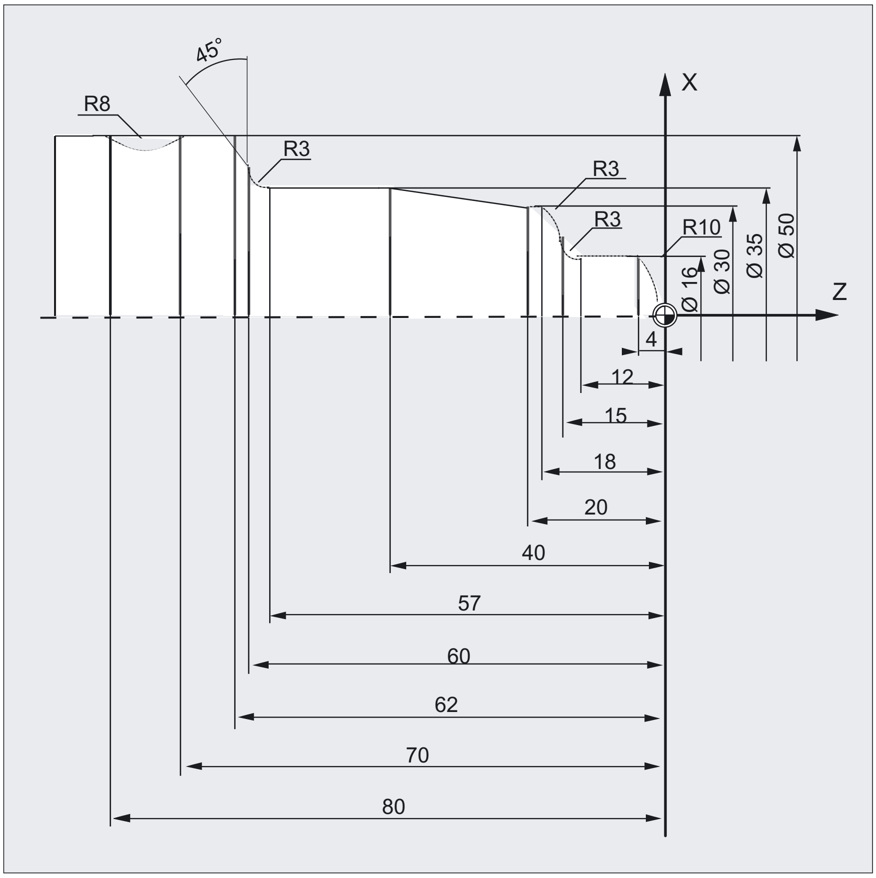
Radius programming and tool radius compensation

The sample program contains radius programming and tool radius compensation.

Programming example

%_N_1001_MPF

```
N5 G0 G53 X280 Z380 D0            ;Start point
N10 TRANS X0 Z250                 ;Zero offset
N15 LIMS=4000                     ;Speed limitation (G96)
N20 G96 S250 M3                   ;Select constant cutting speed
N25 G90 T1 D1 M8                  ;Select tool and offset
N30 G0 G42 X-1.5 Z1               ;Activate tool with tool radius compensation
N35 G1 X0 Z0 F0.25
N40 G3 X16 Z-4 I0 K-10            ;Rotate radius 10
N45 G1 Z-12
N50 G2 X22 Z-15 CR=3              ;Rotate radius 3
N55 G1 X24
N60 G3 X30 Z-18 I0 K-3            ;Rotate radius 3
N65 G1 Z-20
N70 X35 Z-40
N75 Z-57
N80 G2 X41 Z-60 CR=3              ;Rotate 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 up tool and select offset
N105 G96 S210 M3                  ;Select constant cutting speed
N110 G0 G42 X50 Z-60 M8           ;Activate tool with tool radius compensation
N115 G1 Z-70 F0.12                ;Rotate diameter 50
N120 G2 X50 Z-80 I6.245 K-5       ;Rotate radius 8
N125 G0 G40 X100 Z50 M9           ;Retract tool and deselect tool radius
                                  ;compensation
N130 G0 G53 X280 Z380 D0 M5       ;Move to tool change location
N135 M30                          ;Program end
```



Machine manufacturer

The MD settings must be defined correctly before the program can run on the machine.

References: /FB/ Functional description, K2, "Axes, Coordinate Systems,..."

Positional Data 3

3.1 General notes

3.1.1 Program dimensions

In this section you will find descriptions of the commands, with which you can directly program dimensions taken from a drawing. This has the advantage that no extensive calculations have to be made for NC programming.

Note

The commands described in this section stand in most cases at the start of a NC program.

The way, in which these functions are combined, is not intended to be a patent remedy. For example, the choice of working plane may be made at another point in the NC program.

The real purpose of this and all the following sections is to illustrate the conventional structure of an NC program.

Overview of typical dimensions

The basis of most NC programs is a drawing with concrete dimensions.

When implementing in a NC program, it is helpful to take over exactly the dimensions of a workpiece drawing into the machining program. These can be:

- Absolute dimension, G90 modally effective applies for all axes in the block, up to revocation by G91 in a following block.
- Absolute dimension, X=AC(value) only this value applies only for the stated axis and is not influenced by G90/G91. This is possible for all axes and also for SPOS, SPOSA spindle positionings, and interpolation parameters I, J, K.
- Absolute dimension, X=CC(value) directly approaching the position by the shortest route, only this value applies only for the stated rotary axis and is not influenced by G90/G91. Is also possible for SPOS, SPOSA spindle positionings.
- Absolute dimension, X=ACP(value) approaching the position in positive direction, only this value is set for the rotary axis, the range of which is set in the machine datum to 0...< 360°.

- Absolute dimension, X=ACN(value) approaching the position in negative direction, only this value is set for the rotary axis, the range of which is set in the machine datum to 0...< 360°.
- Incremental dimension, G91 modally effective applies for all axes in the block, until it is revoked by G90 in a following block.
- Incremental dimension, X=IC(value) only this value applies exclusively for the stated axis and is not influenced by G90/G91. This is possible for all axes and also for SPOS, SPOSA spindle positionings, and interpolation parameters I, J, K.
- Inch dimension, G70 applies for all linear axes in the block, until revoked by G71 in a following block.
- Metric dimension, G71 applies for all linear axes in the block, until revoked by G70 in a following block.
- Inch dimension as for G70, but applies also for feedrate and length-related setting data.
- Metric dimension as for G71, but applies also for feedrate and length-related setting data.
- Diametral programming, DIAMON on
- Diametral programming, DIAMOF off

Diametral programming, DIAM90 for traversing blocks with G90. Radius programming for traversing blocks with G91.

3.2 Absolute/relative dimensions

3.2.1 Absolute dimension (G90, X=AC)

Function

With the G90 command or the non-modal statement AC you determine the descriptive system for approaching individual axes from setpoints in absolute dimensions.

You program where the tool should travel.

Programming

```
G90
```

Or

```
X=AC(...) Y=AC(...) Z=AC(...)
```

Parameters

G90	Absolute reference dimension
X Y Z	Axis identifiers of the axes to be traversed
=AC	Absolute dimensions non-modally effective

Note

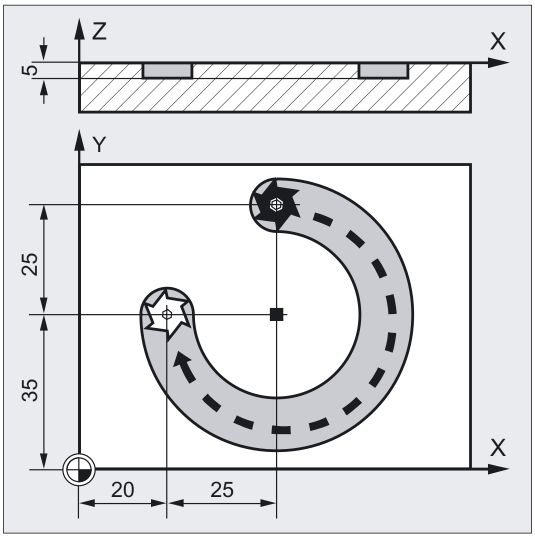
The command G90 is modal.

Generally G90 applies to all axes programmed in subsequent NC blocks.

Example of milling

The traverse paths are entered in absolute coordinates with reference to the workpiece zero.

For entering the circle center point coordinates I and J see circle interpolation G2/G3.

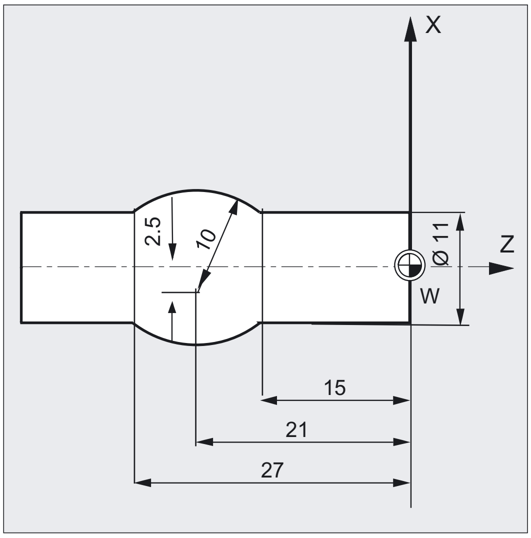


```
N10 G90 G0 X45 Y60 Z2 T1 S2000 M3        ;Absolute dimensioning, rapid traverse to
                                         ;XYZ, tool, spindle on
                                         ;clockwise
N20 G1 Z-5 F500                          ;Tool infeed at feedrate
N30 G2 X20 Y35 I=AC(45) J=AC(35)         ;Circle center point in absolute
                                         ;dimensions
N40 G0 Z2                                ;Retracting
N50 M30                                  ;End of block
```

Example of turning

The traverse paths are entered in absolute coordinates with reference to the workpiece zero.

For entering the circle center point coordinates I and J see circle interpolation G2/G3.



```
N5 T1 D1 S2000 M3                           ;Tool, spindle on clockwise
N10 G0 G90 X11 Z1                           ;Absolute dimensioning, rapid traverse
                                            ;to position XYZ
N20 G1 Z-15 F0.2                            ;Tool infeed at feedrate
N30 G3 X11 Z-27 I=AC(-5) K=AC(-21)          ;Circle center point in absolute
                                            ;dimensions
N40 G1 Z-40                                 ;Retracting
```

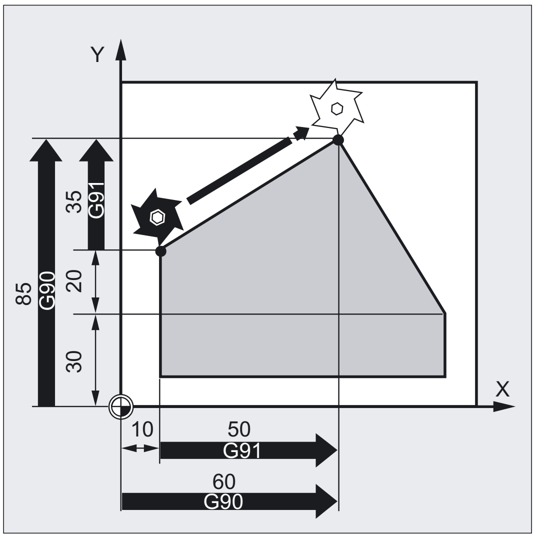
Description

The dimensions refer to the origin of the active coordinate system. You program the point to which the tool is to travel, e.g., in the workpiece coordinate system.

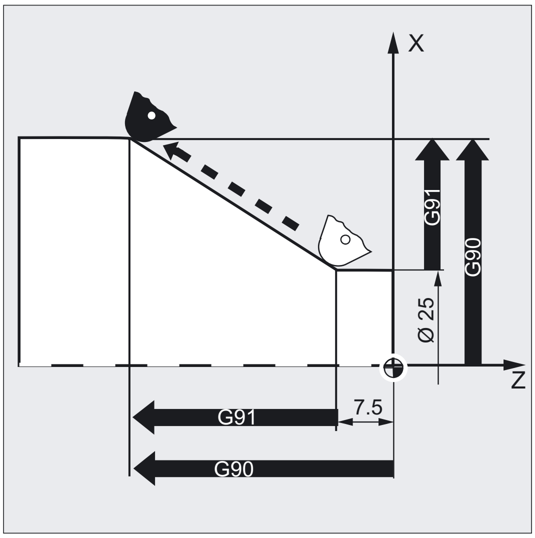
Non-modal absolute dimensioning AC

When incremental dimension G91 is active, AC can be used to allow entry of absolute dimensions for individual axes on a block-by-block basis.

Milling:



Turning:



Note

On conventional turning machines, it is standard practice to interpret incremental traversing blocks in the transverse axis as radius values, while diameter dimensions are valid for absolute coordinates. This conversion for G90 is performed using the commands DIAMON, DIAMOF or DIAM90.

For dimensioning for diameter or radius, see circular interpolation G2/G3.

3.2.2 Incremental dimensions (G91, X=IC)

Function

With the G91 command or the non-modal statement IC, you determine the descriptive system for approaching individual axes from setpoints in incremental dimensions.

You program how far the tool is to travel.

Programming

```
G91
```

Or

```
X=IC(...) Y=IC(...) Z=IC(...)
```

Parameters

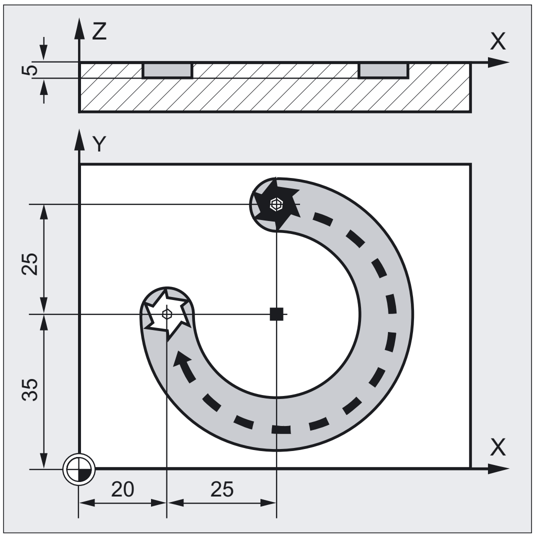
G91	Relative incremental dimensioning
X Y Z	Axis identifiers of the axes to be traversed
=IC	Incremental dimensions non-modally effective

Example of milling

The dimensions refer to the last point approached.

The circle center point coordinates of the circle interpolation are stated non-modally in absolute coordinates, since as default the circle center point is independent of G91.

For entering the circle center point coordinates I and J see circle interpolation G2/G3.

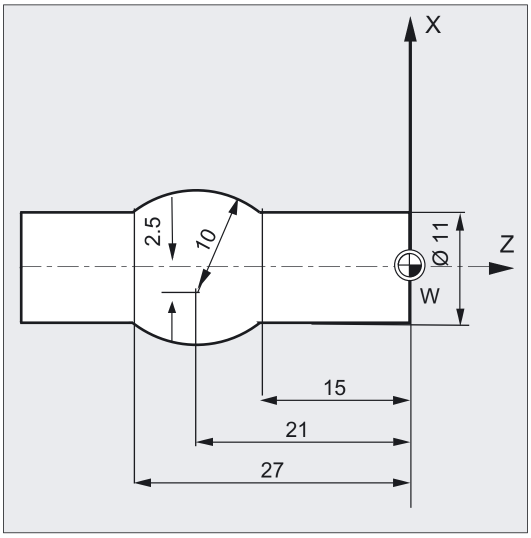


```
N10 G90 G0 X45 Y60 Z2 T1 S2000 M3      ;Absolute dimensioning, rapid traverse to
                                       ;XYZ, tool, spindle on clockwise
N20 G1 Z-5 F500                        ;Tool infeed at feedrate
N30 G2 X20 Y35 I0 J-25)                ;Circle center point in incremental
                                       ;dimensions
N40 G0 Z2                              ;Retracting
N50 M30                                ;End of block
```

Example of turning

The dimensions refer to the last point approached.

For entering the circle center point coordinates I and J see circle interpolation G2/G3.



```
N5 T1 D1 S2000 M3                        ;Tool, spindle on clockwise
N10 G0 G90 X11 Z1                        ;Absolute dimensioning, rapid traverse to
                                         ;position XYZ
N20 G1 Z-15 F0.2                         ;Tool infeed at feedrate
N30 G3 X11 Z-27 I-8 K-6                  ;Circle center point in incremental
                                         ;dimensions
N40 G1 Z-40                              ;Retracting
N50 M30                                  ;End of block
```

Example without traversing through the active zero offset

- G54 contains an offset of 25 in X
- SD 42440: FRAME_OFFSET_INCR_PROG = 0

```
N10 G90 G0 G54 X100
N20 G1 G91 X10                          ;Traverse X by 10 mm, the offset is
                                        ;not traversed
N30 G90 X50                             ;Traverse to position X75, the offset
                                        ;is traversed
```

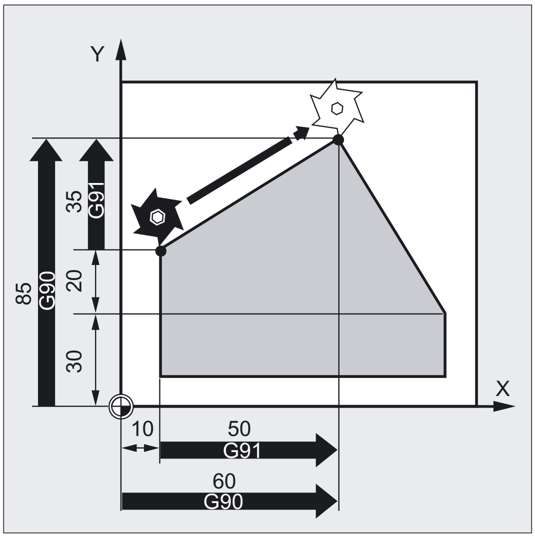
Description

The dimensions refer to the last point approached. You program how far the tool is to travel.

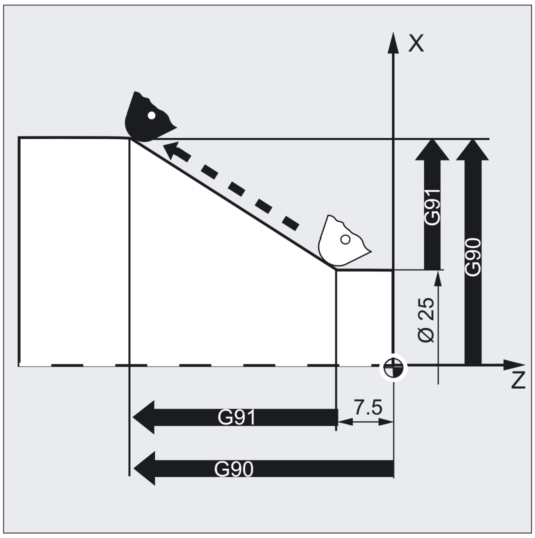
Non-modally effective incremental dimensioning IC

Using IC and with a predefined absolute G90 dimension the incremental dimensioning can be set non-modally for individual axes.

Milling:



Turning:



Note

On conventional turning machines it is standard practice to interpret incremental NC blocks in the transverse axis as radius values, while diameter dimensions are valid for absolute coordinates. This conversion for G91 is performed using the commands DIAMON, DIAMOF or DIAM90.

For dimensioning for diameter or radius see circular interpolation G2/G3.

G91 extension

For applications such as scratching, it is necessary only to traverse the path programmed in the incremental coordinates. The active zero offset or tool offset is not traversed. This can be set separately using setting data.

Incremental dimensioning without traversing through the active tool offset

The active tool offset is not traversed if the setting datum
SD 42442: TOOL_OFFSET_INCR_PROG = 0.

Incremental dimensioning without traversing through the active zero offset

The active zero offset is not traversed if the setting datum
SD 42440: FRAME_OFFSET_INCR_PROG = 0.

3.3 Absolute dimension for rotary axes (DC, ACP, ACN)

With the above parameters you can define the desired approach strategy for positioning rotary axes.

Programming

```
A=DC(…) B=DC(…) C=DC(…)
```

Or

```
A=ACP(…) B=ACP(…) C=ACP(…)
```

Or

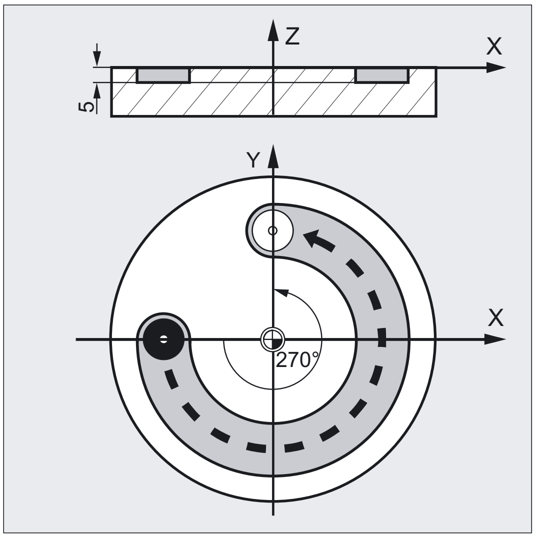
```
A=ACP(…) B=ACP(…) C=ACP(…)
```

Parameters

A B C	Axis identifier for rotary axis to be traversed
DC	Absolute dimensions, approach position directly
ACP	Absolute dimensions, approach position in positive direction
ACN	Absolute dimensions, approach position in negative direction

Example of milling

Machining on a rotary table: The tool is stationary, the table rotates through 270° **in clockwise direction.** to produce a circular groove.



```
N10 SPOS=0                              ;Spindle in position control
N20 G90 G0 X-20 Y0 Z2 T1                ;Absolute, infeed in rapid traverse
N30 G1 Z-5 F500                         ;Lower at feedrate
N40 C=ACP(270)                          ;The table rotates through 270° in
                                        ;clockwise direction (positive), the tool
                                        ;mills a circular groove
N50 G0 Z2 M30                           ;Lift, end of program
```

Absolute dimensioning with DC

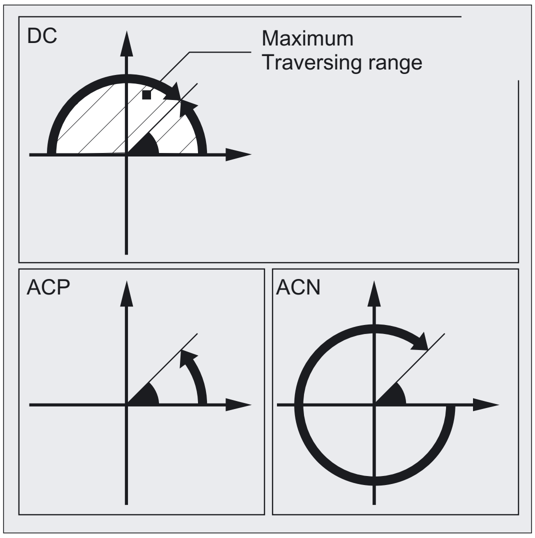
The rotary axis travels to the position programmed in absolute coordinates along the shortest direct path. The rotary axis traverses across an area of up to 180°.

Absolute dimensioning with ACP

The rotary axis travels to the positions programmed in absolute coordinates in the positive direction of axis rotation (counterclockwise).

Absolute dimensioning with ACN

The rotary axis travels to the positions programmed in absolute coordinates in the negative direction of axis rotation (clockwise).



Note

The traversing range must be set to between 0° and 360° in the machine data (modulo method) for positioning with directional data (ACP, ACN). To traverse modulo rotary axes by more than 360° in a block, G91 or IC must be programmed.

The positive direction of rotation (clockwise or counterclockwise) is set in the machine data.

All of the commands are non-modal.

You can also use DC, ACP and ACN for spindle positioning from zero speed.

Example: SPOS=DC(45)

3.4 Dimensions inch/metric, (G70/G700, G71/G710)

Function

Depending on the dimensions in the production drawing, you can program workpiece geometries alternately in metric measurements and inches.

Programming

Call-up

`G70` or `G71`

`G700` or `G710`

Parameters

G70	Imperial measure (length [inches])
G71	Metric dimensions (length [mm])
G700	Imperial measure (length [inch]; feedrate [inch/min])
G710	Metric dimensions (length [mm]; feedrate F [mm/min])

G700/G710

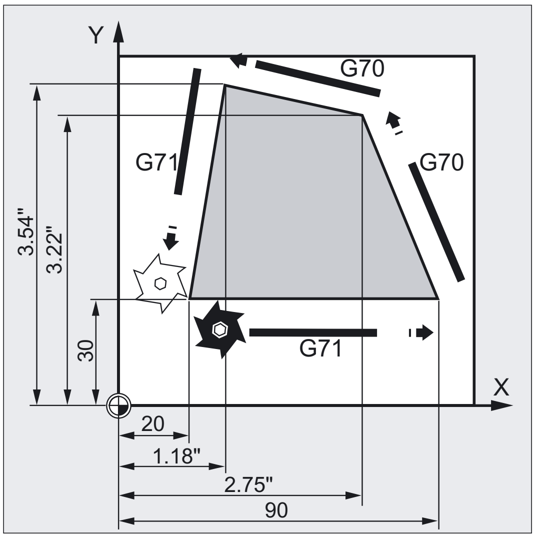
The functionality of G70/G71 has been extended with G700/G710. In addition to the geometrical parameters, the technological parameters, such as feed F, are interpreted during parts program execution in the measuring system set in G700/G710.

The controller interprets all feedrates used with G700/G710 in the programmed measuring system, unlike G70/G71.

The programmed feedrate value is modal and thus does not change automatically on subsequent G70/G71/G700/G710 selections.

Example of milling

Change between metric and imperial input with basic setting metric (G70/G71).



```
N10 G0 G90 X20 Y30 Z2 S2000 M3 T1        ;Basic setting metric
N20 G1 Z-5 F500                          ;At feedrate in Z [mm/min]
N30 X90
N40 G70 X2.75 Y3.22                      ;Enter destination positions in inches,
                                         ;G70
                                         ;is active until deselected by G71 or
                                         ;end of program
N50 X1.18 Y3.54
N60 G71 X 20 Y30                         ;Enter positions in mm
N70 G0 Z2 M30                            ;Retract in rapid traverse, end of
                                         ;program
```

Description

G70 or G71

You can instruct the control to convert the following geometrical dimensions (with necessary deviations) into the measuring system not set and enter them directly:

Examples

- Positional data X, Y, Z, ...
- Interpolation point coordinates I1, J1, K1
  Interpolation parameters I, J, K and circle radius
  CR with circular-path programming
- Thread pitch (G34, G35)
- Programmable zero offset (TRANS)
- Polar radius RP

Note

All other parameters such as feedrates, tool offsets or settable zero offsets are interpreted (when using G70/G71) in the default measuring system (MD 10240: SCALING_SYSTEM_IS_METRIC).

The representation of system variables and machine data is also independent of the G70/G71 context.

If the feedrate in the G70/G71/G700/G710 context is to be activated, a new F value must be programmed explicitly.

All length-related NC data, machine data and setting data for G700/G710 are always read and written in the programmed context of G700/G710.

References: /FB1/, Description of Functions Fundamentals, G2, Chapter "Metric/Inch measuring system".

Synchronized actions

If positioning tasks are performed in synchronized actions and no G70/G71/G700/G710 command is programmed in the synchronized action itself, the G70/G71/G700/G710 context active at the time of execution determines which measuring system is used.

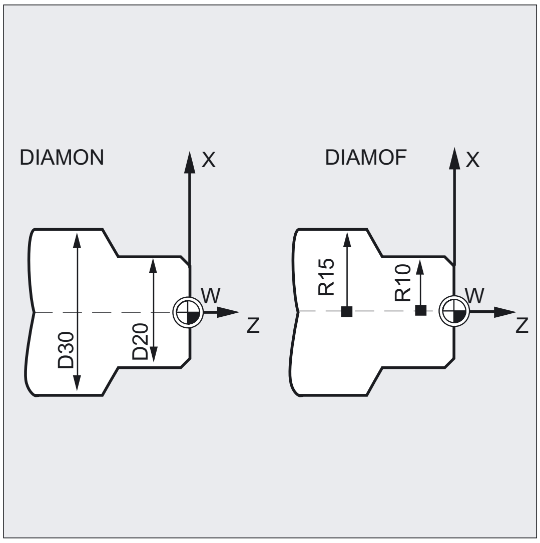
References: /PGA/, Programming Guide Advanced, Section "Motion-Synchronous Actions"
/FBSY/, Description of Functions, Synchronized Actions.

3.5 Special turning functions

3.5.1 Dimensions for radius, diameter, (DIAMON, DIAMOF, DIAM90)

Function

The free choice of diameter or radius dimensions allows you to program the dimensions straight from the engineering drawing without conversion.



After power up of

- DIAMON/DIAM90, diameter dimensions are defined for the specified transverse axis. as diameter.
- DIAM90 is displayed as a diameter, irrespective of the traversing method (G90/G91) the actual value of the transverse axis. This also applies to reading of actual values in the workpiece coordinate system with MEAS, MEAW, $P_EP[x] and $AA_IW[x].

Programming

```
DIAMON
```

Or

```
DIAMOF
```

Or

```
DIAM90
```

Parameters

	Absolute dimensioning (G90)	**Incremental dimensioning (G91)**
DIAMON	Diameter	Diameter
DIAMOF	Radius (for default, see machine manufacturer)	Radius
DIAM90	Diameter	Radius
	The diameter programming is set with DIAM90 for G90 and the radius programming for G91.	

Further information

The commands for diameter and radius data are modal.

By programming DIAMOF you can switch at any time to radius as dimension.

Diameter values (DIAMON/DIAM90)

Diameter values apply to the following data:

- Actual-value display of transverse axis in the workpiece coordinate system
- JOG mode: Increments for incremental dimension and travel with handwheel
- Programming:
  End positions, independent of G90/G91 interpolation parameters inG2/G3, if these are programmed with AC absolutely
- Actual values read in the workpiece coordinate system in
  MEAS, MEAW, $P_EP[X], $AA_IW[X] (see PGA, Programming Guide "Advanced")

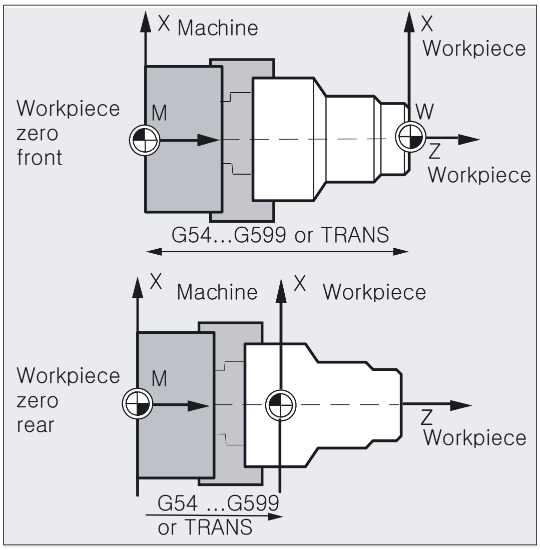
Example

```
N10 G0 X0 Z0                            ;Approach starting point
N20 DIAMOF                              ;Diameter input off
N30 G1 X30 S2000 M03 F0.7               ;X axis = transverse axis
                                        ;radius dimensions active
                                        ;Traverse to radius position X30
N40 DIAMON                              ;Diameter dimensions active
N50 G1 X70 Z-20                         ;Traverse to diameter position X70 and
                                        ;Z-20
N60 Z-30
N70 DIAM90                              ;Diameter programming for absolute
                                        ;dimensions and radius programming for
                                        ;incremental dimensions
N80 G91 X10 Z-20                        ;Incremental dimension
N90 G90 X10                             ;Absolute dimensions
N100 M30                                ;End of program
```

3.5.2 Position of workpiece

Function

While the machine zero is fixed, you can choose the position for the workpiece zero on the longitudinal axis. The workpiece zero is generally located on the front or rear side of the workpiece.



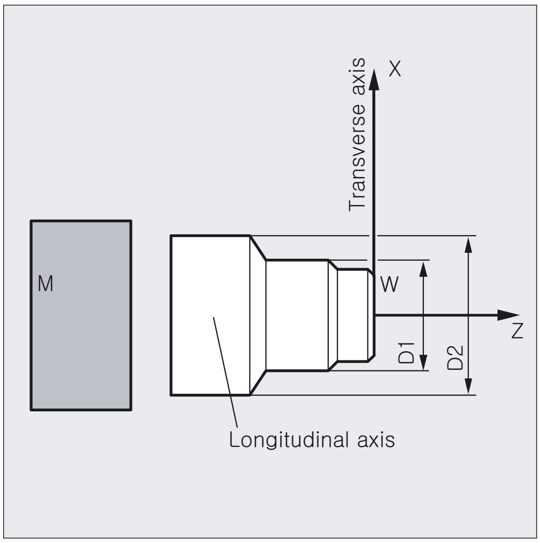
Zero points

Both the machine zero and the workpiece zero are positioned on the center of rotation. The settable offset on the X axis is thus zero.

coordinate system

The dimensions for the transverse axis are generally specified as diameter measurements (double path dimension as compared to other axes).

The geometry axis to be used as a transverse axis is defined in machine data.



Parameters

`G54 to G599 or TRANS`	`Call for the position of the workpiece zero`
`M`	`Machine zero`
`W`	`Tool zero point`
`Z axis`	`Longitudinal axis`
`X axis`	`Transverse axis`

The two mutually perpendicular geometry axes are usually designated as follows:

- Longitudinal axis = Z axis (abscissa)
- Transverse axis = X axis (ordinate)

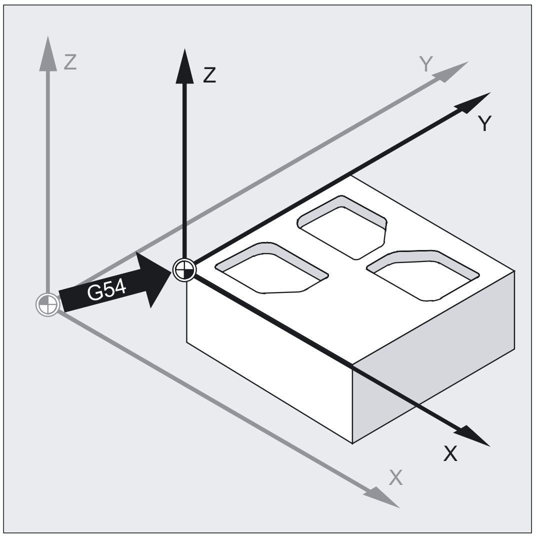
3.6 Zero offset (frame), G54 to G57, G505 to G599, G53, G500/SUPA

Function

The settable zero offset relates the workpiece zero on all axes to the origin of the basic coordinate system.

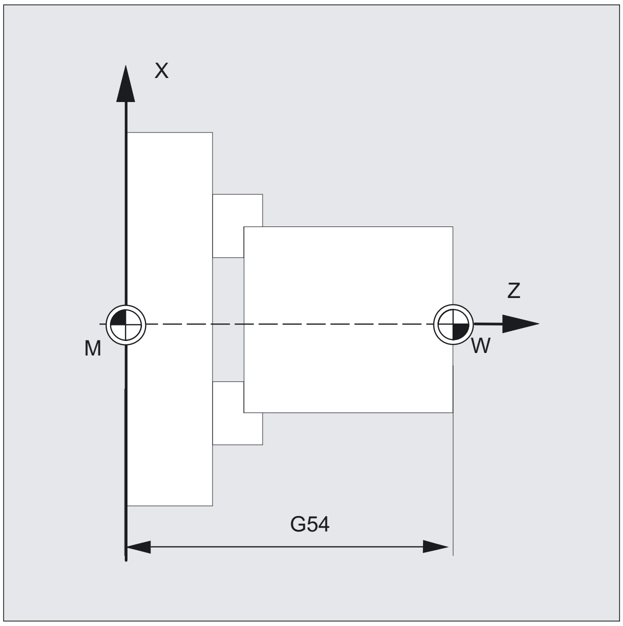
It is, therefore, possible to call up cross-program zero points for different fixtures with a G command.

Milling:



For turning, for example, the offset value for tightening the chuck is entered in G54.

Turning:



Programming

Call-up

`G54`

Or

`G55`

Or

`G56`

Or

`G57`

Or

`G505 … G599`

Switching off

`G53`

Or

`G500`

Or

`SUPA`

Or

`G153`

Parameters

G54 to G57	Call the second to fifth settable zero offset/frame
G505 ...G599	Call the 6th to the 99th settable zero offset
G53	Non-modal deactivation of current settable zero offset and programmable zero offset
G500	G500=zero frame, default setting, (contains no offset, rotation, mirroring or scaling)
	Deactivation of settable zero offsets/frames (G54 to G599) until the next call.
	Activation of the total basic frame ($P_ACTBFRAME).
	G500 is not 0
	Activation of first settable zero offset/frames ($P_UIFR[0]) and
	Activation of total basic frame ($P_ACTBFRAME), or a modified basic frame is activated.
SUPA	Non-modal deactivation, including programmed offsets, handwheel offsets (DRF), external zero offset and PRESET offset.
G153	Non-modal suppression of settable, programmable and total basic frame

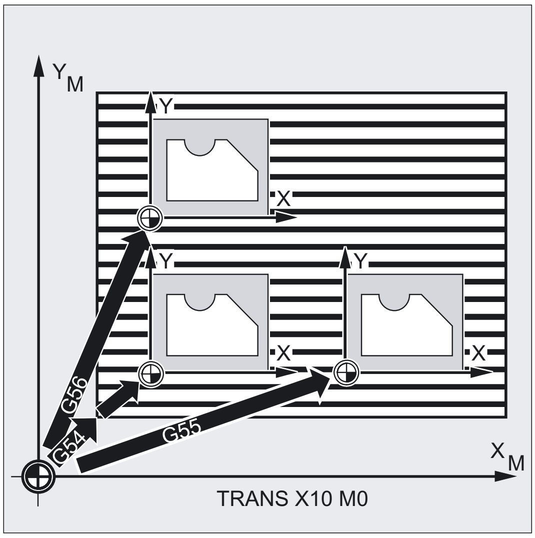
For more information please refer to Frame section.

Offset of the zero in the Cartesian coordinate system by frames such as

- Programmable zero offset, e.g., TRANS, ATRANS
- Programmable rotations, e.g., ROT, AROT
- Programmable scalings, e.g., SCALE, ASCALE
- Programmable mirrorings, e.g., MIRROR, AMIRROR

Example

In this example, three workpieces, arranged on a pallet according to the zero offset values G54 to G56, are machined successively. The machining sequence is programmed in subprogram L47.



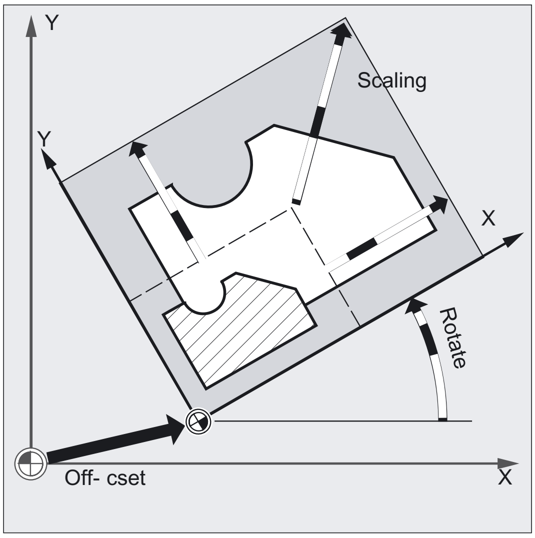
```
N10 G0 G90 X10 Y10 F500 T1          ;Approach
N20 G54 S1000 M3                    ;Call the first zero offset,
                                    ;spindle clockwise
N30 L47                             ;Run program, in this case as a
                                    ;subprogram
N40 G55 G0 Z200                     ;Call the second zero offset
                                    ;Z via obstacle
N50 L47                             ;Run program as subprogram
N60 G56                             ;Call third zero offset
N70 L47                             ;Run program as subprogram
N80 G53 X200 Y300 M30               ;Suppress zero offset,
                                    ;end of program
```

Description

Setting the offset values

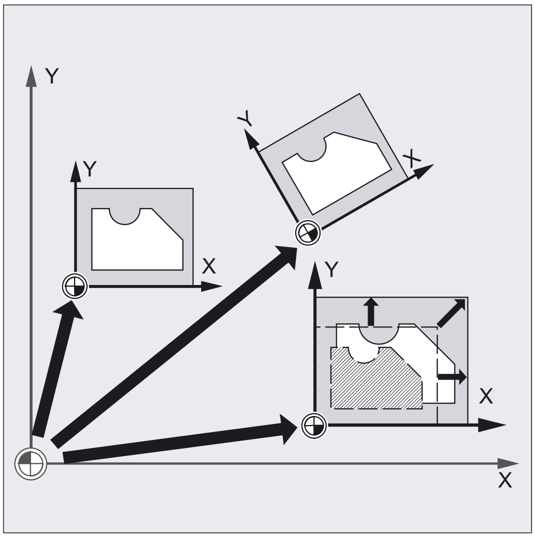
On the operator panel or universal interface, enter the following values in the internal control zero offset table:

- Coordinates for the offset
- Angle for rotated clamping and
- Scale factors if necessary



Switching on zero offset, G54 to G57

In the NC program, the zero offset is moved from the machine coordinate system to the workpiece coordinate system by executing one of the four commands G54 to G57.



In the next NC block with a programmed movement, all of the positional parameters and thus the tool movements refer to the workpiece zero, which is now valid.

Note

The 4 available zero offsets can be used, for example, for multiple machining operations, to describe 4 workpiece clamping positions simultaneously and execute them in the program.

Further settable zero offsets, G505 to G599

Command numbers G505 to G599 are available for this purpose. This enables you to create up to 100 settable zero offsets in total, in addition to the 4 default zero offsets G54 to G57, by using the machine data. They are stored in the zero point memory.

Deactivating a zero offset

Command G500 activates the first settable zero offset including basic offset, i.e., when zero frame is selected as the default, the current settable zero offset is deactivated.

G53 suppresses the programmable and settable offset modally.

G153 has the same effect as G53 and also suppresses the total basic frame.

SUPA has the same effect as G153 and also suppresses the DRF offset, overlaid motions, and external ZOs.

Note

The basic setting at program start, e.g., G54 or G500, can be set with machine data.

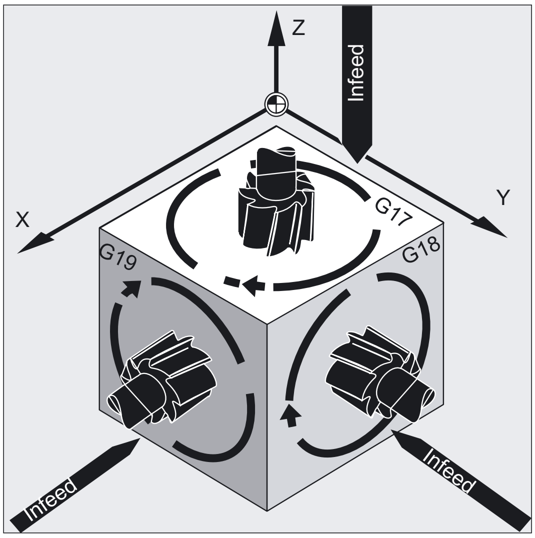
You will find more information on programmable zero offsets in the Frames section "Programmable zero offset".

3.7 Selection of working plane (G17 to G19)

Function

The specification of the working plane, in which the desired contour is to be machined also defines the following functions:

- The plane for tool radius compensation.
- The infeed direction for tool length compensation depending on the tool type.
- The plane for circular interpolation.



Programming

Call-up

```
G17
```

or

```
G18
```

or

```
G19
```

Parameter

G17	Working plane X/Y Infeed direction Z Plane selection 1st - 2nd geometry axis
G18	Working plane Z/X Infeed direction Y Plane selection 3rd - 1st geometry axis
G19	Working plane Y/Z Infeed direction X Plane selection 2nd - 3rd geometry axis

Note

In the default setting, G17 (X/Y plane) is defined for milling and G18 (Z/X plane) is defined for turning.

When calling the tool path correction G41/G42 (see Section "Tool offsets"), the working plane must be defined so that the controller can correct the tool length and radius.

Example of milling

The "conventional" approach with milling tool:

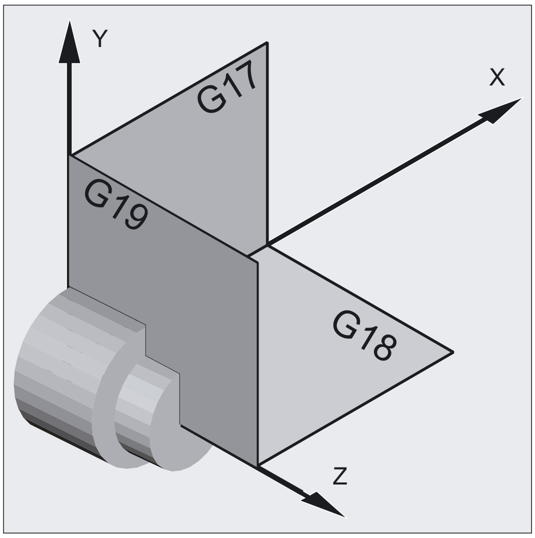
- Define working plane (G17 default setting for milling).
- Select tool type (T) and tool offset values (D).
- Switch on path correction (G41).
- Program traversing movements.

```
N10 G17 T5 D8                             ;G17 Call the working plane, in this
                                          ;case: X/Y T,
                                          ;D tool call. Tool length compensation is
                                          ;performed in the Z direction.
N20 G1 G41 X10 Y30 Z-5 F500               ;Radius compensation is performed in the
                                          ;X/Y plane.
N30 G2 X22.5 Y40 I50 J40                  ;Circular interpolation/tool radius
                                          ;compensation in
                                          ;the X/Y plane.
```

Description

It is advisable to define the working plane G17 to G19 at the beginning of the program. In the default setting, the Z/X plane is preset for turning G18.

Turning:



For calculating the direction of rotation, the controller requires the specification of the working plane, refer to circular interpolation G2/G3.

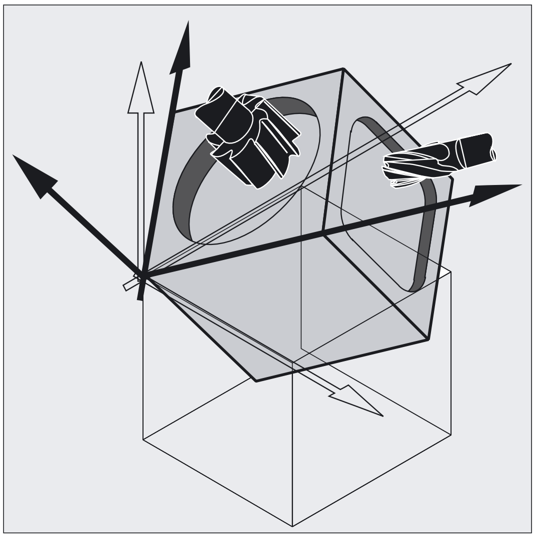
Machining on inclined planes

Rotate the coordinate system with ROT (see Section "Coordinate system offset") to position the coordinate axes on the inclined surface. The working planes rotate accordingly.

Tool length compensation on inclined planes

As a general rule, the tool length compensation always refers to the fixed, non-rotated working plane.

Milling:



Note

The tool length components can be calculated according to the rotated working planes with the functions for "Tool length compensation for orientable tools".

The offset plane is selected with CUT2D, CUT2DF. For further information on this and for the description of the available calculation methods, refer to Section "Tool offsets"

The control provides convenient coordinate transformation functions for the spatial definition of the working plane.

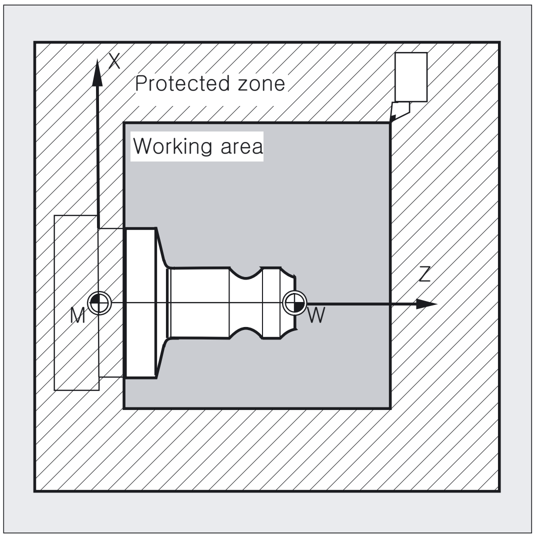
For further information, please refer to Section "Coordinate system offset".

3.8 Working area limitation (G25/G26, WALIMON, WALIMOF)

Function

G25/G26 limits the working area, in which the tool can traverse in all channel axes. These commands allow you to set up protection zones in the working area, which are out of bounds for tool movements.

The working area limitation for all validated axes must be programmed with the WALMON command. The WALIMOF command deactivates the working area limitation. WALIMON is the default setting. Therefore, it only has to be programmed if working area limitation has been disabled.



Programming

`G25 X…Y…Z…` Programming in a separate NC block

Or

`G26 X…Y…Z…` Programming in a separate NC block

Or

`WALIMON`

Or

`WALIMOF`

Parameters

`G25, X Y Z`	Lower working area limitation, value assignment in the channel axes in the basic coordinate system
`G26, X Y Z`	Upper working area limitation, value assignment in the channel axes in the basic coordinate system
`WALIMON`	Switch on working area limitation for all axes
`WALIMOF`	Working area limitation switch-off for all axes

Note

G25/G26 can also be used to program limits for spindle speeds at the address S. For further information, please refer to "Feed control and spindle motion".

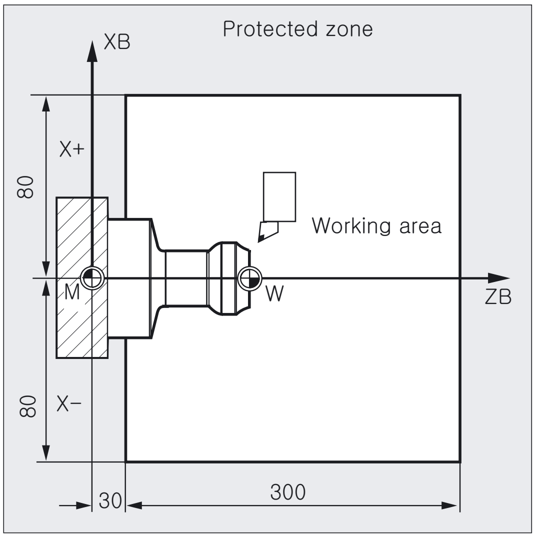
In addition to programming values using G25/G26, you can also parameterize them in setting data.

The axial setting data define the axes, for which the working area limitation is valid.

Example of turning

A protection zone is defined in the working area of a turning machine. This protects the surrounding equipment such as turrets, measuring stations, etc., against damage.

Default setting: WALIMON



```
N10 G0 G90 F0.5 T1
N20 G25 X-80 Z30                          ;Define the lower limit for
                                          ;the individual coordinate axes
N30 G26 X80 Z330                          ;Define the upper limit
N40 L22                                   ;Cutting program
N50 G0 G90 Z102 T2                        ;To tool change location
N60 X0
N70 WALIMOF                               ;Deactivate working area limitation
N80 G1 Z-2 F0.5                           ;Drill
N90 G0 Z200                               ;Back
N100 WALIMON                              ;Activate working area limitation
N110 X70 M30                              ;End of program
```

Description

Reference points on the tool

When tool length compensation is active, the reference point is the tip of the tool, otherwise it is the toolholder reference point. If the tool is positioned outside the specified area or leaves this area, the program stops executing.

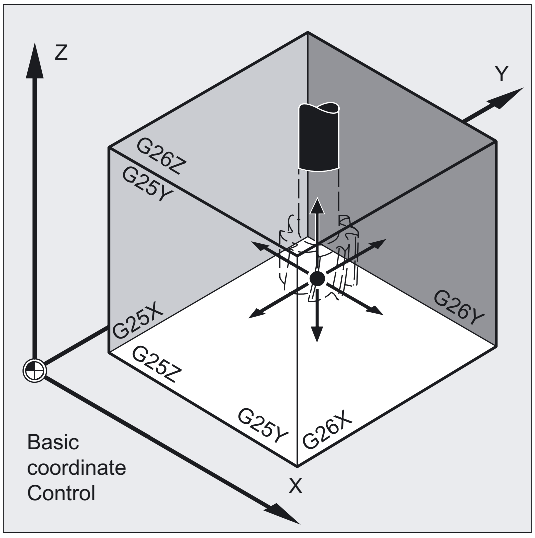
Programmable working area limitation, G25/G26

An upper (G26) and lower (G25) working area limit can be defined for each axis. These values apply immediately and are not lost on Reset and when the control is switched on again.

The tool (milling tool) radius can be changed in the channel-specific machine data $MC_WORKAREA_WITH_TOOL_RADIUS (see "Advanced").

Note

The coordinates for the individual axes apply in the basic coordinate system!



Further information

You will find the CALCPOSI subroutine in the Advanced Programming Guide This routine can be used to check in advance whether traversal of the planned path will take into account working area limitations or protection zones.

3.9 Reference point approach (G74)

Function

When the machine has been powered up (where incremental position measuring systems are used), all of the axis slides must approach their reference mark. Only then can traversing movements be programmed.

The reference point can be approached in the NC program with G74.

Programming

`G74 X1=0 Y1=0 Z1=0 A1=0` … Programmed in a separate NC block

Parameter

G74	Reference point approach
X1=0 Y1=0 Y1=0…	The stated machine axis address
A1=0 B1=0 C1=0…	X1, Y1, Z1… Reference point approach for linear axes. A1, B1, C1… Reference point approach for rotary axes.

Note

A transformation must not be programmed for an axis which is to approach the reference point with G74.

The transformation is deactivated with command TRAFOOF.

Example

When the measurement system is changed, the reference point is approached and the workpiece zero point is initialized.

```
N10 SPOS=0                           ;Spindle in position control
N20 G74 X1=0 Y1=0 Z1=0 C1=0          ;Reference point approach for linear axes
                                     ;and rotary axes
N30 G54                              ;Zero offset
N40 L47                              ;Cutting program
N50 M30                              ;End of program
```

Programming Motion Commands 4

4.1 General notes

In this section you will find a description of all the travel commands you can use to machine workpiece contours.

These travel commands with the associated parameters enable you to program quite different workpiece contours for milling and also for turning.

Travel commands for programmable workpiece contours

The programmed workpiece contours are composed of straight lines and circular arcs. A helix can be produced by combining these two elements.

Executed in succession, these contour elements produce the workpiece contour.

A programming example is given for each travel command.

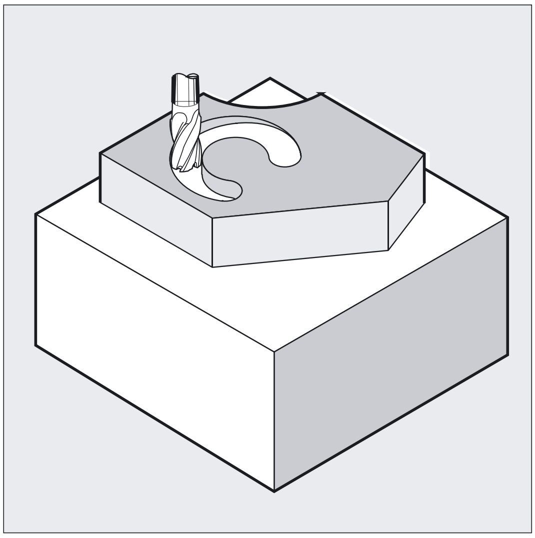
Programming more complex sequences of motions is also discussed. These are also described with the possible variants or special cases.

The positional data contain all necessary geometric data serving for an unequivocal representation of the positions in the corresponding coordinate systems. These are:

- Travel commands with specification of coordinates
- Rapid traverse movements to the end point
- Linear interpolation 3D surface processing
- Circular interpolation for full circles or circular arcs
- Helical interpolation
- Involute interpolation
- Thread cutting and tapping
- Interruptions both of approaching certain positions and traveling to these
- Special turning functions
- Chamfering or rounding off contour corners

Tool prepositioning

Before a machining process is started, you need to position the tool in such a way as to avoid any damage to the tool or workpiece.



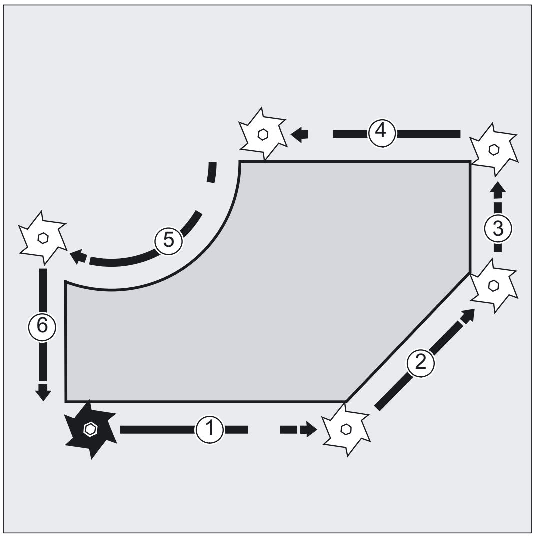
Start point - destination point

The traversing movement always runs from the last approached position to the programmed destination position. This destination position is also the start position for the next travel command.

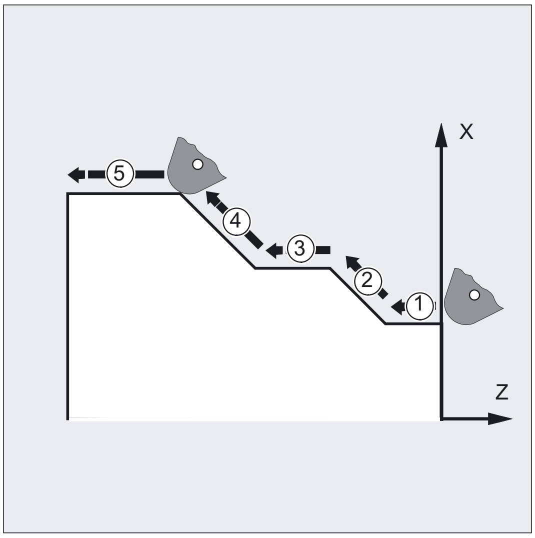
Number of axis values

Depending on the control configuration, you can program motions for up to 8 axes in each motion block. These may include path axes, synchronized axes, positioning axes, and spindle oscillation mode.

Number of motion blocks in milling:



Number of motion blocks in turning:



Caution

An axis address can only be programmed once in each block.

These commands can be programmed in Cartesian or polar coordinates. Synchronized axes, positioning axes and oscillation mode.

4.2 Travel commands with polar coordinates, polar angle, polar radius

4.2.1 Defining the pole (G110, G111, G112)

Function

The dimensioning starting point is called a pole. The pole can be specified in either Cartesian or polar coordinates (polar radius RP=... and polar angle AP=...). The programming commands G110 to G112 are used to provide a unique definition of the reference point for dimensions. Therefore, neither absolute nor incremental dimensioning affects the system defined in the programming command.

Programming

`G110 X… Y… Z…`	Pole parameter, with reference to the last approached position in Cartesian coordinates
Or	
`G110 AP=… RP=…`	Pole parameter, with reference to the last approached pole in polar coordinates
Or	
`G111 X… Y… Z…`	Pole parameter, absolute in workpiece coordinate system with Cartesian coordinates
Or	
`G111 AP=… RP=…`	Pole parameter, absolute in workpiece coordinate system with polar coordinates
Or	
`G112 X… Y… Z…`	Pole parameter, with reference to the last valid pole with Cartesian coordinates
Or	
`G112 AP=… RP=…`	Pole parameter, with reference to the last valid pole with polar coordinates

Parameters

G110	Polar programming relative to the last programmed setpoint position
G111	Polar programming relative to origin of current workpiece coordinate system
G112	Polar programming relative to the last valid pole
X Y Z	Coordinate identifiers of the axes to be traversed
AP=	Polar angle, value range ±0…360°, angle refers to horizontal axis of the working plane
RP=	Polar radius in mm or inches always in absolute positive values.

Note

In the NC program you can switch non-modally between polar and Cartesian coordinates.

You come directly back into the Cartesian system by using the Cartesian coordinate identifiers (X, Y, Z...). The defined pole is moreover retained up to program end.

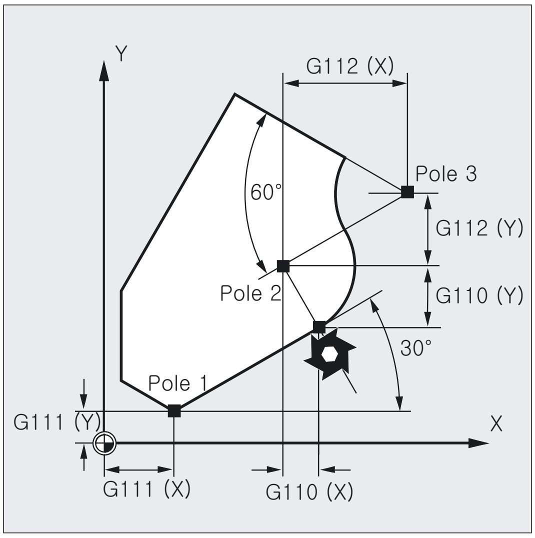
Note

All the commands relating to pole input must be programmed in a separate NC block.

If no pole is specified, the origin of the current coordinate system applies.

Example of defining a pole with G110, G111, G112

The statement of the pole in Cartesian G110(X,Y), G111(X,Y) G112(X,Y) or polar coordinates by stating G110, G111, G112 with polar angle AP= and polar radius RP=.

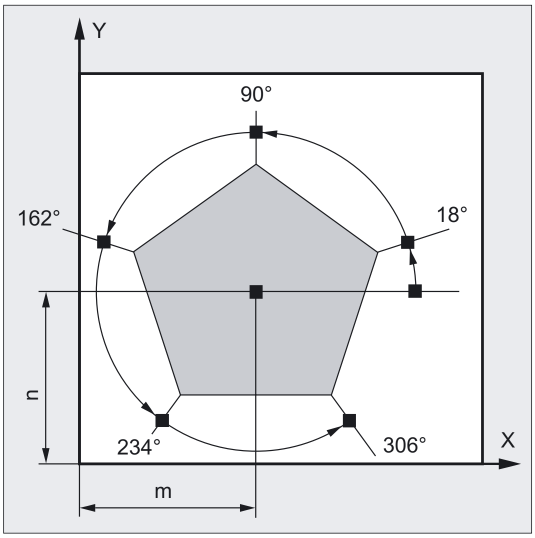


4.2.2 Traversing commands with polar coordinates, (G0, G1, G2, G3 AP=..., RP=...)

Function

The polar coordinate method is useful only if a workpiece or part of a workpiece has radius and angle measurements. Polar coordinates can be used to program these dimensions directly in accordance with the drawing.

If the dimensions of a workpiece, such as those in hole patterns, proceed from a central point, then the dimensions are given in terms of angles and radii.



Programming

```
G0 AP=… RP=…
```

Or

```
G1 AP=… RP=…
```

Or

```
G2 AP=… RP=…
```

Or

```
G3 AP=… RP=…
```

The new end position is defined relative to a pole, see defining the pole G110, G111, G112.

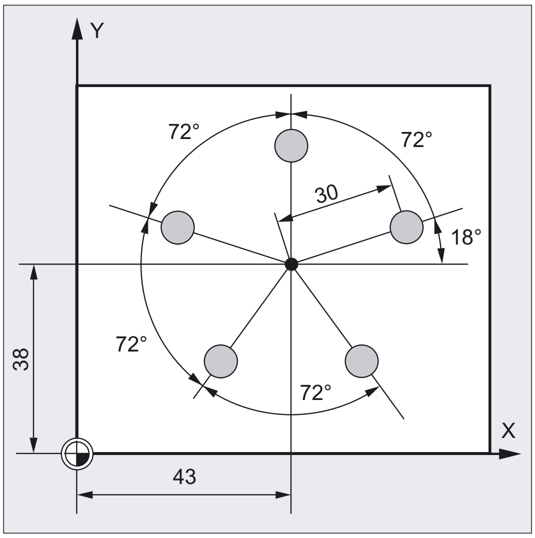
Parameters

G0	Rapid traverse movement
G1	Linear interpolation
G2	Circular interpolation clockwise
G3	Circular interpolation counter-clockwise
AP=	Polar angle, value range ±0…360°, the polar angle can be defined both absolutely and incrementally.
RP=	Polar radius in mm or inches always in absolute positive values.
=AC(...)	Absolute dimensioning
=IC(...)	Incremental dimensioning

Example of making a hole pattern

The positions of the holes are specified in polar coordinates.

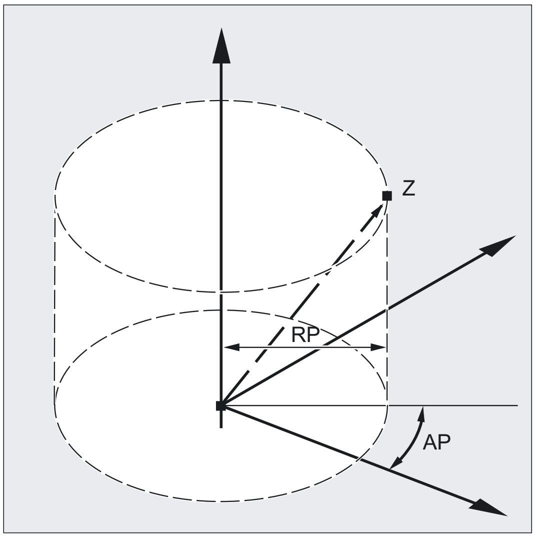
Each hole is machined with the same production sequence: Predrilling, drilling to size, reaming, etc. The machining sequence is filed in the subroutine.



```
N10 G17 G54                           ;Working plane X/Y, workpiece zero
N20 G111 X43 Y38                      ;Define pole
N30 G0 RP=30 AP=18 Z5                 ;Approach starting point, position in
                                      ;cylindrical coordinates
N40 L10                               ;Subprogram call
N50 G91 AP=72                         ;Approach next position in rapid
                                      ;traverse, polar angle in incremental
                                      ;dimensions, polar radius
                                      ;from block N30 is still stored and does
                                      ;not need to be specified
N60 L10                               ;Subprogram call
N70 AP=IC(72)                         ;…
N80 L10                               ;…
N90 AP=IC(72)
N100 L10                              ;…
N110 AP=IC(72)
N120 L10                              ;…
N130 G0 X300 Y200 Z100 M30            ;Retract tool, program end
N90 AP=IC(72)
N100 L10                              ;…
```

Example of cylinder coordinates

The 3rd geometry axis, which lies perpendicular to the working plane, can also be specified in Cartesian coordinates.



This enables spatial parameters to be programmed in cylindrical coordinates.

Example: G17 G0 AP… RP… Z…

Traversing commands

The positions stated with polar coordinates can be approached with rapid traverse G0, linear interpolation G1, circular interpolation clockwise G2 or counterclockwise G3.

Working plane

The polar coordinates are valid in the working plane selected with G17 to G19.

You must not program Cartesian coordinates, such as interpolation parameters or axis addresses, for the selected working plane in NC blocks with polar end position coordinates.

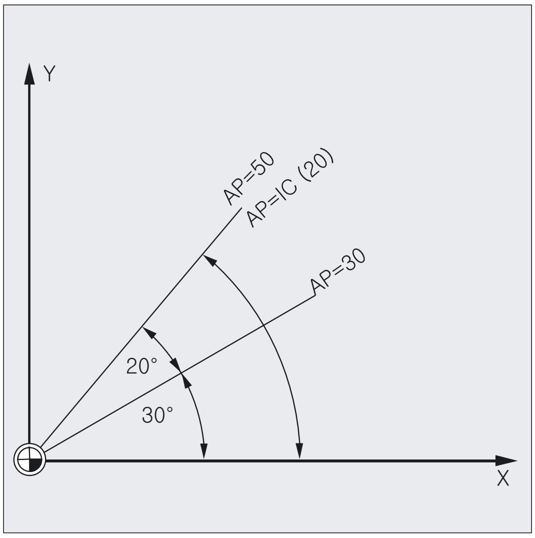
Polar angle AP

When absolute dimensions are specified, the angular reference is based on the horizontal axis of the working plane, e.g., X axis with G17. The positive direction of rotation is counter-clockwise.

The polar angle can be defined both absolutely and incrementally.

When incremental coordinates are entered (AP=IC…), the last angle programmed is taken as the reference. The polar angle is stored until a new pole is defined or the working plane is changed.

If no pole is defined, then the origin of the current workpiece coordinate system is considered as pole.



Polar radius RP

The polar radius remains stored until a new value is input.

If the modally active polar radius is RP = 0

The polar radius is calculated from the distance between the starting point vector in the polar plane and the active pole vector. The calculated polar radius is stored modally afterwards.

This applies irrespective of a selected pole definition such as G110, G111, G112. If both points are programmed identically, this radius becomes 0 and alarm 14095 is generated.

If a pole angle AP is programmed with RP = 0

If the current block contains a polar angle AP rather than a polar radius RP and if there is a difference between the current position and pole in workpiece coordinates, then this difference is applied as the polar radius and stored modally. If the difference = 0, the pole coordinates are specified again and the modal polar radius remains zero.

4.3 Rapid traverse movement (G0, RTLION, RTLIOF)

Function

You can use the rapid traverse movements to position the tool rapidly, to travel round the workpiece or to approach tool change locations.

Non-linear interpolation is activated using RTLIOF parts program commands; linear interpolation is activated using RTLION.

Note

This function is not suitable for workpiece machining!

Programming

```
G0 X… Y… Z…
```

Or

```
G0 AP=…
```

Or

```
G0 RP=…
```

Or

```
RTLIOF
```

Or

```
RTLION
```

Parameters

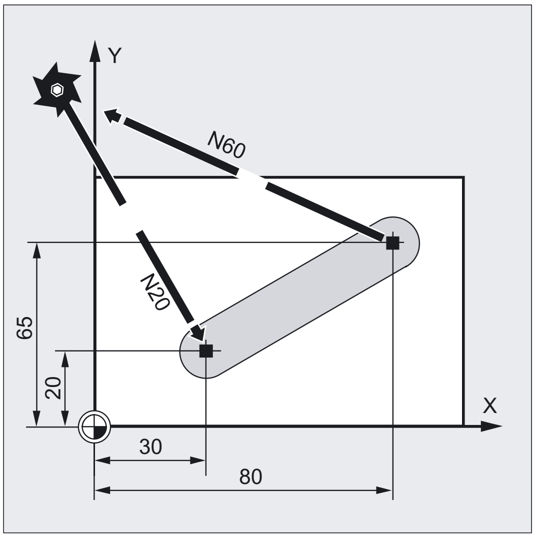
G0	Rapid traverse movement
X Y Z	End point in Cartesian coordinates
AP=	End point in polar coordinates, in this case the polar angle
RP=	End point in polar coordinates, in this case the polar radius
RTLIOF with G0	Nonlinear interpolation (each path axis interpolates as a single axis)
RTLION with G0	Linear interpolation (path axes are interpolated together)

Note

G0 cannot be replaced with G.

G0 is modal.

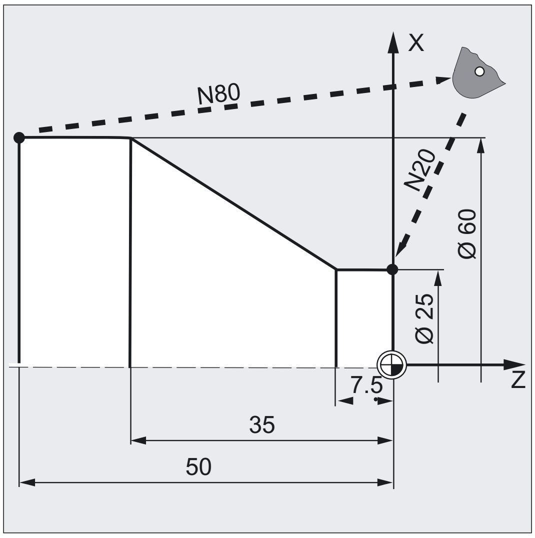
Example of milling



Start positions or tool change points, retracting the tool, etc., are approached with G0.

```
N10 G90 S400 M3                          ;Absolute dimensioning, spindle clockwise
N20 G0 X30 Y20 Z2                        ;Approach start position
N30 G1 Z-5 F1000                         ;Tool infeed
N40 X80 Y65                              ;Travel on straight line
N50 G0 Z2
N60 G0 X-20 Y100 Z100 M30                ;Retract tool, program end
```

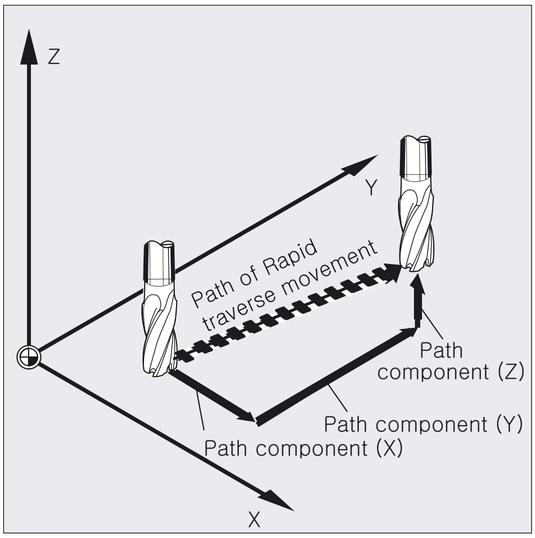
Example of turning



```
N10 G90 S400 M3                          ;Absolute dimensioning, spindle clockwise
N20 G0 X25 Z5                            ;Approach start position
N30 G1 G94 Z0 F1000                      ;Tool infeed
N40 G95 Z-7.5 F0.2
N50 X60 Z-35                             ;Travel on straight line
N60 Z-50
N70 G0 X62
N80 G0 X80 Z20 M30                       ;Retract tool, program end
```

Description

The tool movement programmed with G0 is executed at the highest possible speed (rapid traverse). The rapid traverse speed is defined separately for each axis in machine data. If the rapid traverse movement is executed simultaneously on several axes, the rapid traverse speed is determined by the axis, which requires the most time for its section of the path.



Traversing path axes as positioning axes with G0

Path axes can travel in one of two different modes to execute movements in rapid traverse:

- **Linear interpolation** (behavior in earlier SW version):
  The path axes are interpolated together.
- **Non-linear interpolation**:
  Each pat axis is interpolated as an individual (positioning) axis independently of the other axes involved in the rapid traverse movement.

With non-linear interpolation, the setting for the appropriate positioning axis BRISKA, SOFTA or DRIVEA applies, with reference to the axial jerk.

Notice

Since a different contour can be traversed in nonlinear interpolation mode, synchronized actions that refer to coordinates of the original path are not operative in some cases!

Linear interpolation applies in the following cases:

- With a G code combination including G0, which does not permit positioning movements (e.g., G40/41/42).
- With a combination of G0 and G64
- When the compressor is active
- When a transformation is active

Example

```
G0 X0 Y10
G0 G40 X20 Y20
G0 G95 X100 Z100 m3 s100
```

Path POS[X]=0 POS[Y]=10 is traversed in path mode. No revolutional feedrate is active if path POS[X]=100 POS[Z]=100 is traversed.

Settable block change time with G0

For single axis interpolations, a new movement end criterion

`FINEA`

Or

`COARSEA`

Or

`IPOENDA`

can already be set for block change inside the braking ramp.

Consecutive axes are handled in G0 like positioning axes.

With the combination of

- "Block change settable in the braking ramp of the single axis interpolation" and
- "Traversing path axes in rapid traverse movement as positioning axes with G0"

all axes can travel to their end point independently of one another. In this way, two sequentially programmed X and Z axes are treated like positioning axes in conjunction with G0.

The block change to axis Z can be initiated by axis X as a function of the braking ramp time setting (100-0%). Axis Z starts to move while axis X is still in motion. Both axes approach their end point independently of one another.

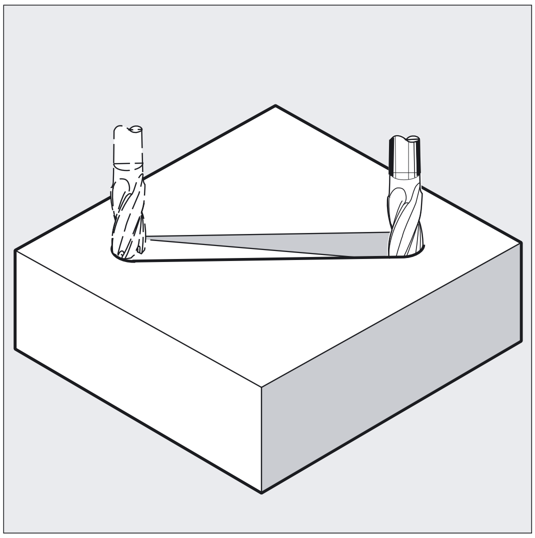
For further information, please refer to "Feed control and spindle motion".

4.4 Linear interpolation (G1)

Function

With G1, the tool travels along straight lines that are parallel to the axis, inclined or in any orientation in space. Linear interpolation permits machining of 3D surfaces, grooves, etc.

Milling:



Programming

```
G1 X… Y… Z … F…
```

Or

```
G1 AP=… RP=… F…
```

Parameters

G1	Linear interpolation with feedrate (linear interpolation)
X Y Z	End point in Cartesian coordinates
AP=	End point in polar coordinates, in this case the polar angle
RP=	End point in polar coordinates, in this case the polar radius
F	Feedrate in mm/min. The tool travels at feedrate F along a straight line from the current starting point to the programmed destination point. You can enter the destination point in Cartesian or polar coordinates. The workpiece is machined along this path. Example: G1 G94 X100 Y20 Z30 A40 F100 The end point on X, Y, Z is approached at a feedrate of 100 mm/min; the rotary axis A is traversed as a synchronized axis, ensuring that all four movements are completed at the same time.

Note

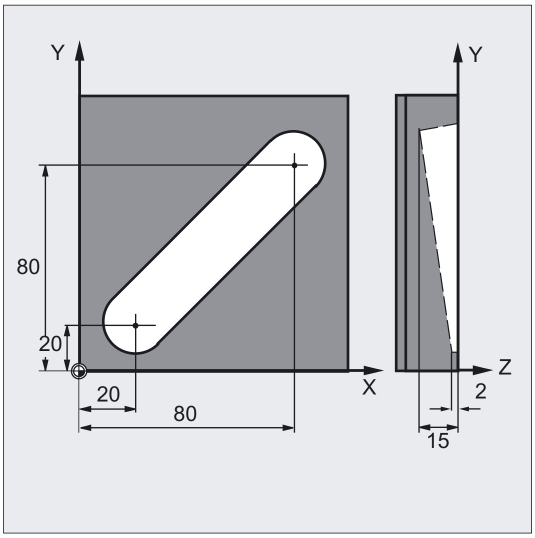
G1 is modal.

The spindle speed S and the direction of spindle rotation M3/M4 must be specified for machining.

FGROUP can be used to define groups of axes, to which the path feed F applies. You will find more information in the "Path behavior" section.

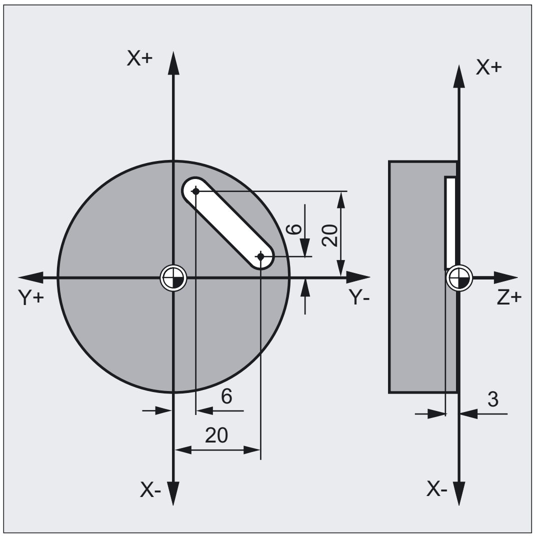
Example of milling

Machining of a groove: The tool travels from the starting point to the end point in the X/Y direction. Infeed takes place simultaneously in the Z direction.



```
N10 G17 S400 M3                      ;Select working plane, spindle clockwise
N20 G0 X20 Y20 Z2                    ;Approach start position
N30 G1 Z-2 F40                       ;Tool infeed
N40 X80 Y80 Z-15                     ;Travel along inclined
                                     ;straight line
N50 G0 Z100 M30                      ;Retract to tool change point
```

Example of turning



```
N10 G17 S400 M3                          ;Select working plane, spindle clockwise
N20 G0 X40 Y-6 Z2                        ;Approach start position
N30 G1 Z-3 F40                           ;Tool infeed
N40 X12 Y-20                             ;Travel along inclined
                                         ;straight line
N50 G0 Z100 M30                          ;Retract to tool change point
```

4.5 Circular interpolation types, (G2/G3, CIP, CT)

Possibilities of programming circular movements

The control provides a range of different ways to program circular movements. This allows you to implement almost any type of drawing dimension directly. The circular movement is described by the:

- center point and end point in the absolute or incremental dimension (default)
- Radius and end point in Cartesian coordinates
- Opening angle and end point in Cartesian coordinates or center point under the addresses
- Polar coordinates with the polar angle AP= and the polar radius RP=
- Intermediate and end point
- End point and tangent direction at the start point.

Programming

`G2/G3 X… Y… Z…`
`I=AC(…) J=AC(…) K=AC(…)` Absolute center point and end point with reference to the workpiece zero

Or

`G2/G3 X… Y… Z… I… J… K…` Center point in incremental dimensions with reference to the circle starting point

Or

`G2/G3 X… Y… Z… CR=…` Circle radius CR= and circle end position in Cartesian coordinates X..., Y..., Z...

Or

`G2/G3 X… Y… Z… AR=…` Opening angle AR= end point in Cartesian coordinates X..., Y..., Z...

Or

`G2/G3 I… J… K… AR=…` Opening angle AR= center point at addresses I..., J..., K...

Or

`G2/G3 AP=… RP=…` Polar coordinates with the polar angle AP= and the polar radius RP=

Or

`CIP X… Y… Z… I1=AC(…) J1=AC(…) K1=(AC…)` The intermediate point at addresses I1=, J1=, K1=

Or

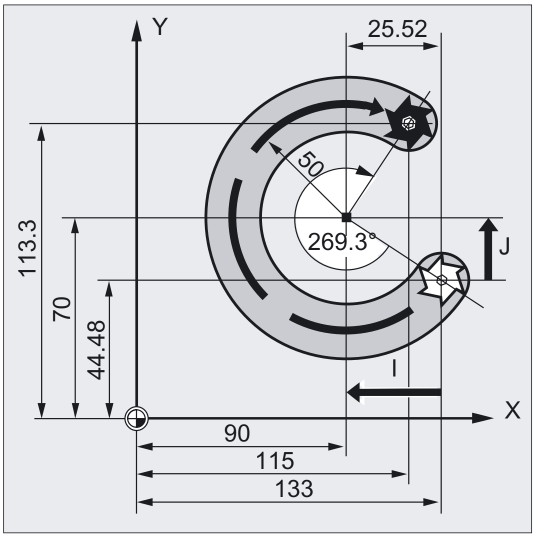
`CT X… Y… Z…` Circle through starting and end point and tangent direction at starting point

Parameters

G2	Circular interpolation, CW
G3	Circular interpolation counterclockwise
CIP	Circular interpolation through intermediate point
CT	Circle with tangential transition defines the circle
X Y Z	End point in Cartesian coordinates
I J K	Circle center point in Cartesian coordinates in X, Y, Z direction
CR=	Circle radius
AR=	Opening angle
AP=	End point in polar coordinates, in this case the polar angle
RP=	End point in polar coordinates, in this case polar radius corresponding to circle radius
I1= J1= K1=	Intermediate points in Cartesian coordinates in X, Y, Z direction

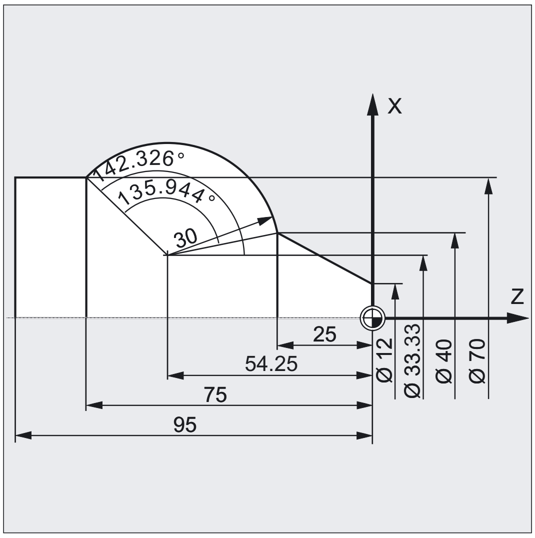
Example of milling

The following program lines contain an example for each circular programming possibility. The necessary dimensions are shown in the production drawing on the right.



```
N10 G0 G90 X133 Y44.48 S800 M3                ;Approach starting point
N20 G17 G1 Z-5 F1000                          ;Tool infeed
N30 G2 X115 Y113.3 I-43 J25.52                ;Circle end point, center point in
                                              ;incremental dimensions
Or
N30 G2 X115 Y113.3 I=AC(90) J=AC(70)          ;Circle end point, center point in
                                              ;absolute dimensions
Or
N30 G2 X115 Y113.3 CR=-50                     ;Circle end point, circle radius
Or
N30 G2 AR=269.31 I-43 J25.52                  ;Opening angle, center point in
                                              ;incremental dimensions
Or
N30 G2 AR=269.31 X115 Y113.3                  ;Opening angle, circle end point
Or
N30 N30 CIP X80 Y120 Z-10                     ;Circle end point and intermediate
                                              ;point:
 I1= IC(-85.35)J1=IC(-35.35) K1=-6            ;Coordinates for all
                                              ;three geometry axes
N40 M30                                       ;End of program
```

Example of turning

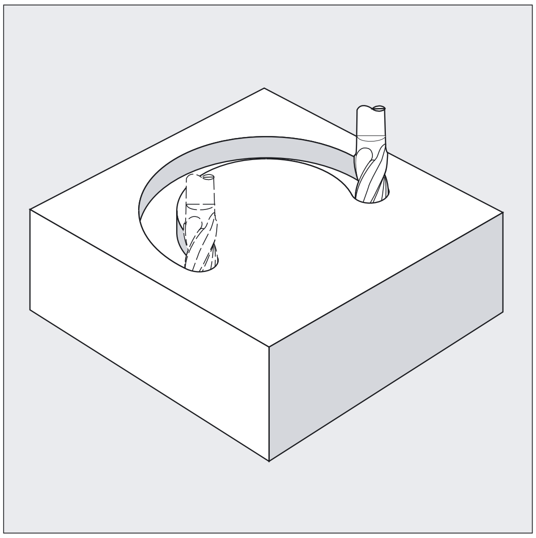


```
N.. ...
N120 G0 X12 Z0
N125 G1 X40 Z-25 F0.2
N130 G3 X70 Y-75 I-3.335 K-29.25              ;Circle end point, center point in
                                               ;incremental dimensions
Or
N130 G3 X70 Y-75 I=AC(33.33) K=AC(-54.25)     ;Circle end point, center point in
                                               ;absolute dimensions
Or
N130 G3 X70 Z-75 CR=30                         ;Circle end point, circle radius
Or
N130 G3 X70 Z-75 AR=135.944                    ;Opening angle, circle end point
Or
N130 G3 I-3.335 K-29.25 AR=135.944             ;Opening angle, center point in
                                               ;incremental dimensions
Or
N130 G3 I=AC(33.33) K=AC(-54.25)               ;Opening angle, center point in
 AR=135.944                                    ;absolute dimensions
Or
N130 G111 X33.33 Z-54.25                       ;Polar coordinates
N135 G3 RP=30 AP=142.326                       ;Polar coordinates
Or
N130 CIP X70 Z-75 I1=93.33 K1=-54.25           ;Programming a circle with
                                               ;intermediate and end points
N140G1 Z-95
N.. ...
N40 M30                                        ;End of program
```

4.6 Circular interpolation with center point and end point (G2/G3, I=, J=, K=AC...)

Function

Circular interpolation enables machining of full circles or arcs.



The circular movement is described by:

- The end point in Cartesian coordinates X, Y, Z and
- the circle center point at addresses I, J, K.

If the circle is programmed with a center point but no end point, the result is a full circle.

Programming

```
G2/G3 X… Y… Z… I… J… K…
```

Or

```
G2/G3 X… Y… Z… I=AC(…) J=AC(…) K=(AC…)
```

Parameters

G2	Circular interpolation, CW
G3	Circular interpolation counterclockwise
X Y Z	End point in Cartesian coordinates
I	Coordinates of the circle center point in the X direction
J	Coordinates of the circle center point in the Y direction
K	Coordinates of the circle center point in the Z direction
=AC(…)	Absolute dimensions (non-modal)

Note

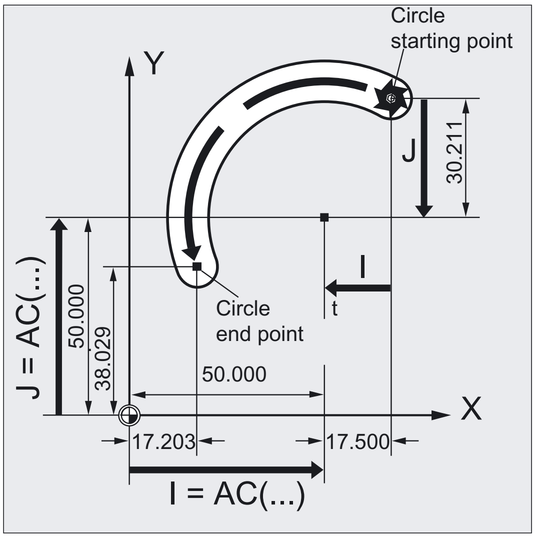
G2 and G3 are modal.

The G90/G91 defaults for absolute or incremental dimensions are only valid for the circle end point.

The center point coordinates I, J, K are normally entered in incremental dimensions with reference to the circle starting point.

You program the absolute center point non-modally with reference to the workpiece zero point with: I=AC(…), J=AC(…), K=AC(…). An interpolation parameter I, J, K with value 0 can be omitted but the second associated parameter must always be specified.

Examples for milling



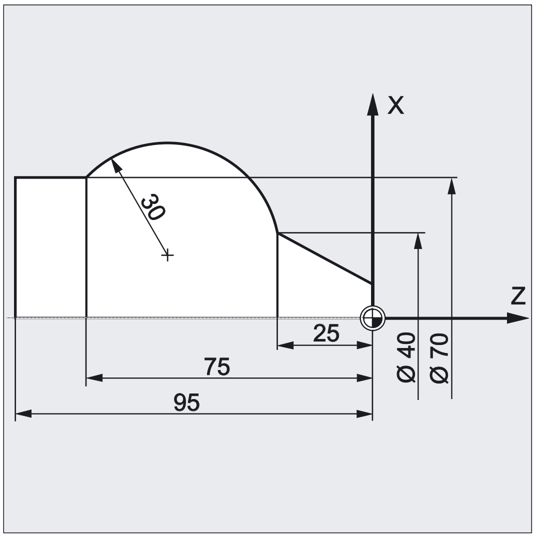
Incremental dimension

```
N10 G0 X67.5 Y80.211
N20 G3 X17.203 Y38.029 I–.5 J–.211 F500
```

Absolute dimensions

```
N10 G0 X67.5 Y80.211
N20 G3 X17.203 Y38.029 I=AC(50) J=AC(50)
```

Examples for turning



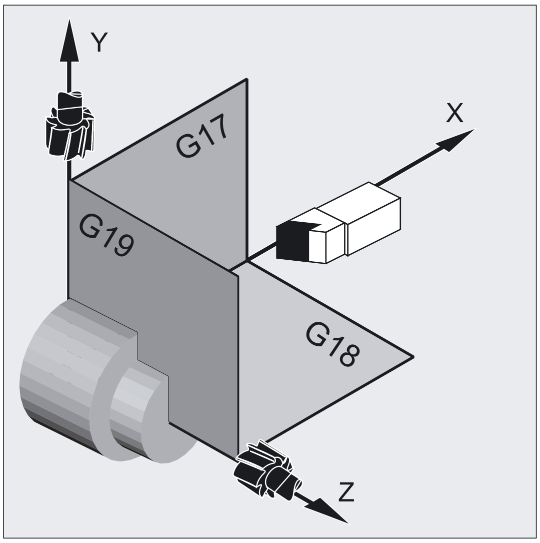
Incremental dimension

```
N120 G0 X12 Z0
N125 G1 X40 Z-25 F0.2
N130 G3 X70 Z-75 I-3.335 K-29.25
N135 G1 Z-95
```

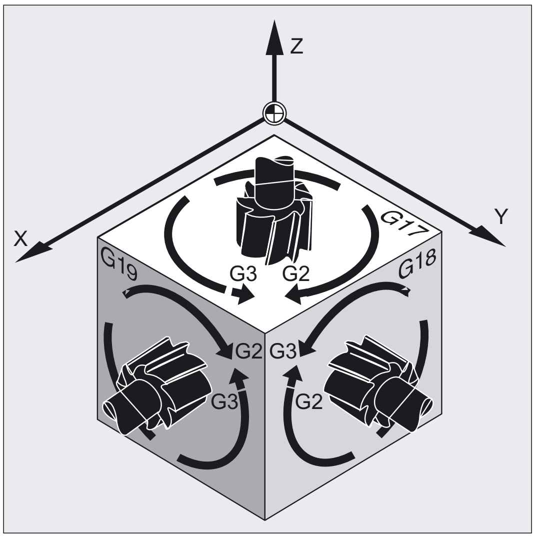
Absolute dimensions

```
N120 G0 X12 Z0
N125 G1 X40 Z-25 F0.2
N130 G3 X70 Z-75 I=AC(33.33) K=AC(-54.25)
N135 G1 Z-95
```

Indication of working plane



The control needs the working plane parameter (G17 to G19) in order to calculate the direction of rotation for the circle – G2 is clockwise or G3 is counterclockwise.



It is advisable to specify the working plane generally.

Exception:

You can also machine circles outside the selected working plane (not with arc angle and helix parameters). In this case, the axis addresses that you specify as an end point determine the circle plane.

Programmed feedrate

You can use FGROUP to specify, which axes are to be traversed with a programmed feedrate. For more information please refer to the Path behavior section.

4.7 Circular interpolation with radius and end point (G2/G3, CR)

The circular movement is described by the:

- Circle radius CR= and
- the end point in Cartesian coordinates X, Y, Z.

In addition to the circle radius, you must also specify the leading sign +/– to indicate whether the traversing angle is to be greater than or less than 180°. A positive leading sign can be omitted.

Note

There is no practical limitation on the maximum size of the programmable radius.

Programming

```
G2/G3 X… Y… Z… CR=
```

Or

```
G2/G3 I… J… K… CR=
```

Parameters

G2	Circular interpolation, CW
G3	Circular interpolation counterclockwise
X Y Z	End point in Cartesian coordinates. These data depend on the motion commands G90/G91 or ...=AC(...)/...=IC(..)
I J K	Circle center point in Cartesian coordinates (in X, Y, Z direction)
	The identifiers have the following meanings:
	I: Coordinate of the circle center point in the X direction
	J: Coordinate of the circle center point in the Y direction
	K: Coordinate of the circle center point in the Z direction
CR=	Circle radius
	Here: CR=+…: Angle less than or equal to 180°
	CR=–…: Angle more than 180°

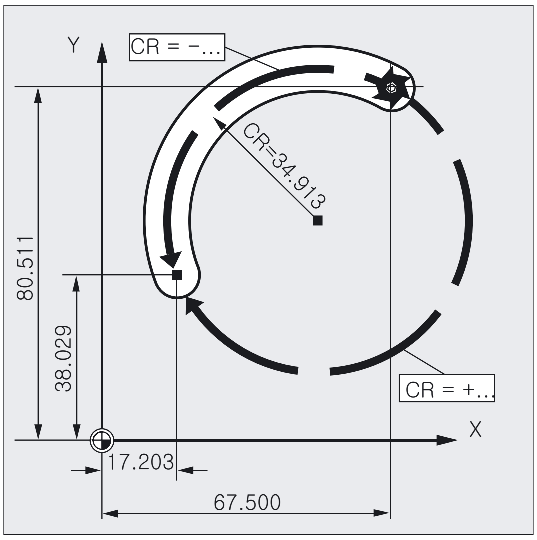
Note

You don't need to specify the center point with this procedure. Full circles (traversing angle 360°) cannot be programmed with CR=, but must be programmed using the circle end point and interpolation parameters.

Example of milling

Programming a circle with radius and end point

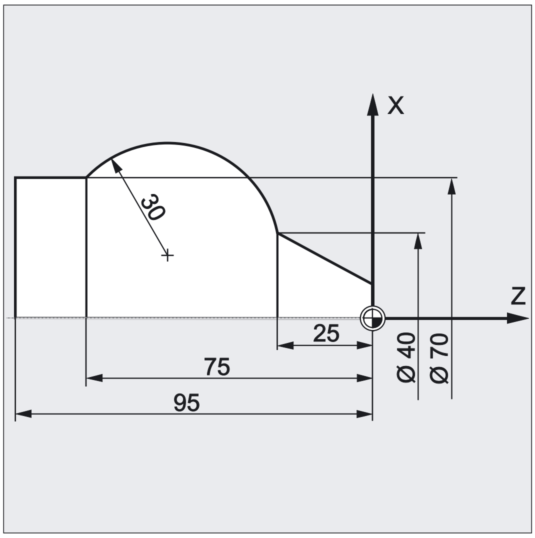
```
N10 G0 X67.5 Y80.511
N20 G3 X17.203 Y38.029 CR=34.913 F500
```



Example of turning

Programming a circle with radius and end point

```
N125 G1 X40 Z-25 F0.2
N130 G3 X70 Z-75 CR=30
N135 G1 Z-95
```



4.8 Circular interpolation with arc angle and center point (G2/G3, AR=)

The circular movement is described by:

- The opening angle AR = and
- the end point in Cartesian coordinates X, Y, Z or
- the circle center at addresses I, J, K

Programming

```
G2/G3 X… Y… Z… AR=
```

or

```
G2/G3 I… J… K… AR=
```

Parameter

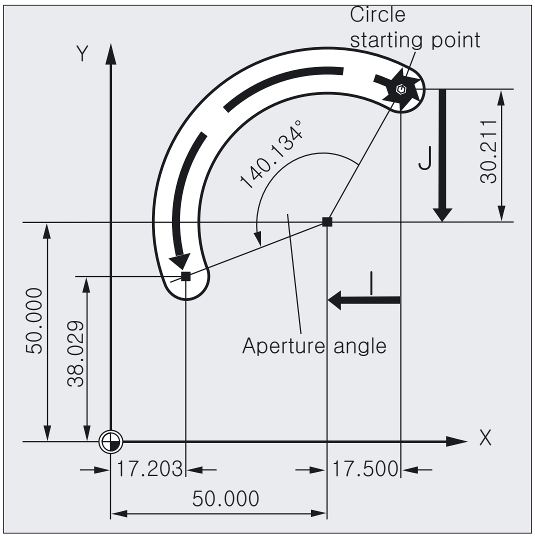
G2	Circular interpolation, CW
G3	Circular interpolation, CCW
X Y Z	End point in Cartesian coordinates
I J K	Circle center in Cartesian coordinates (in X, Y, Z direction) Key: I: Coordinate of the circle center in the X direction J: Coordinate of the circle center in the Y direction K: Coordinate of the circle center in the Z direction
AR=	Opening angle, range of values 0° to 360°
=AC(…)	Absolute dimensioning (non-modal)

Note

Full circles (traversing angle 360°) cannot be programmed with AR=, but must be programmed using the circle end position and interpolation parameters. The center point coordinates **I**, J, K are normally entered in incremental dimensions with reference to the circle starting point.

You program the absolute center point non-modally with reference to the workpiece zero with: **I**=AC(…), J=AC(…), K=AC(…). An interpolation parameter **I**, J, K with value 0 can be omitted but the second associated parameter must always be specified.

Example of milling



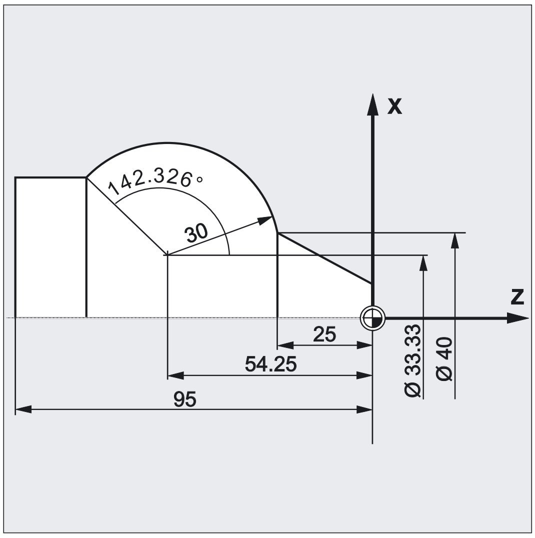
Programming a circle with opening angle and center point or end point

```
N10 G0 X67.5 Y80.211
N20 G3 X17.203 Y38.029 AR=140.134 F500
```

or

```
N20 G3 I-17.5 J-30.211 AR=140.134 F500
```

Example of turning



Programming a circle with opening angle and center point or end point

```
N125 G1 X40 Z-25 F0.2
N130 G3 X70 Z-75 AR=135.944
```

or

```
N130 G3 I-3.335 K-29.25 AR=135.944
```

or

```
N130 G3 I=AC(33.33) K=AC(-54.25) AR=135.944
N135 G1 Z-95
```

4.9 Circular interpolation with polar coordinates (G2/G3, AP=, RP=)

The circular movement is described by:

- the polar angle AP=
- and the polar radius RP=

The following rule applies:

The pole lies at the circle center.

The polar radius corresponds to the circle radius.

Programming

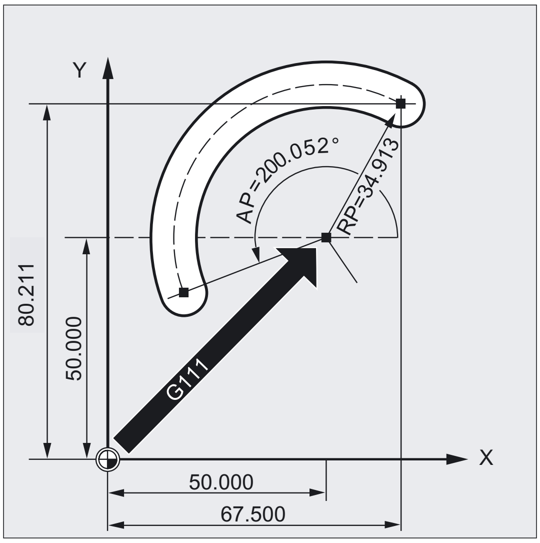
```
G2/G3 AP= RP=
```

Parameter

G2	Circular interpolation, CW
G3	Circular interpolation, CCW
X Y Z	End point in Cartesian coordinates
AP=	End point in polar coordinates, in this case the polar angle

RP=	End point in polar coordinates, in this case polar radius corresponds to circle radius

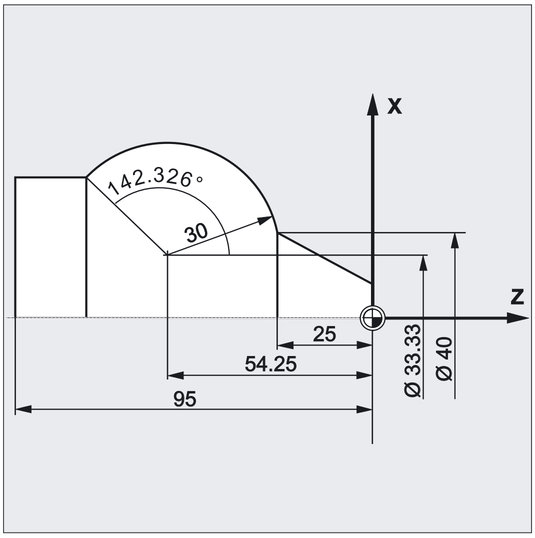
Example of milling



Programming a circle with polar coordinates

```
N10 G0 X67.5 Y80.211
N20 G111 X50 Y50
N30 G3 RP=34.913 AP=200.052 F500
```

Example of turning



Programming a circle with polar coordinates

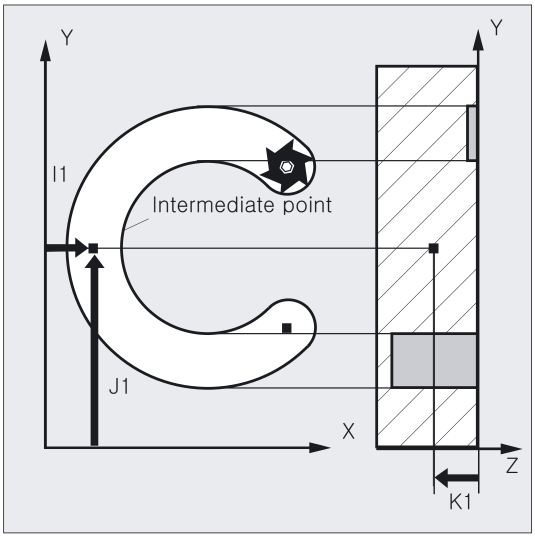
```
N125 G1 X40 Z-25 F0.2
N130 G111 X33.33 Z-54.25
N135 G3 RP=30 AP=142.326
N140 G1 Z-95
```

4.10 Circular interpolation with intermediate and end points (CIP)

You can use CIP to program arcs. These arcs can also be inclined in space. In this case, you describe the intermediate and end points with three coordinates.

The circular movement is described by:

- the intermediate point at addresses I1=, J1=, K1= and
- the end point in Cartesian coordinates X, Y, Z.



The traversing direction is determined by the order of the starting point, intermediate point and end point.

Programming

```
CIP X… Y… Z… I1=AC(…) J1=AC(…) K1=(AC…)
```

Parameters

CIP	Circular interpolation through intermediate point
X Y Z	End point in Cartesian coordinates. These data depend on the motion commands G90/G91 or ...=AC(...)/...=IC(..)
I1= J1= K1=	Circle center point in Cartesian coordinates (in X, Y, Z direction) The identifiers have the following meanings: I: Coordinate of the circle center point in the □X direction J: Coordinate of the circle center point in the □Y direction K: Coordinate of the circle center point in the □Z direction
=AC(…)	Absolute dimensions (non-modal)
=IC(...)	Incremental dimensions (non-modal)

Note

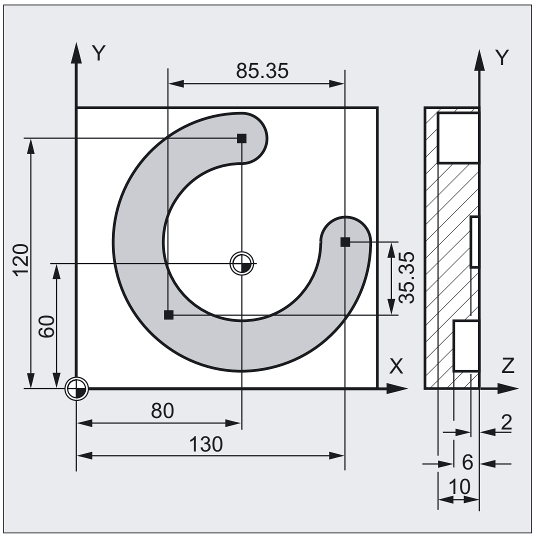
CIP is modal.

Input in absolute and incremental dimensions

The G90/G91 defaults for absolute or incremental dimensions are valid for the intermediate and circle end points.

With G91, the circle starting point is used as the reference for the intermediate point and end point.

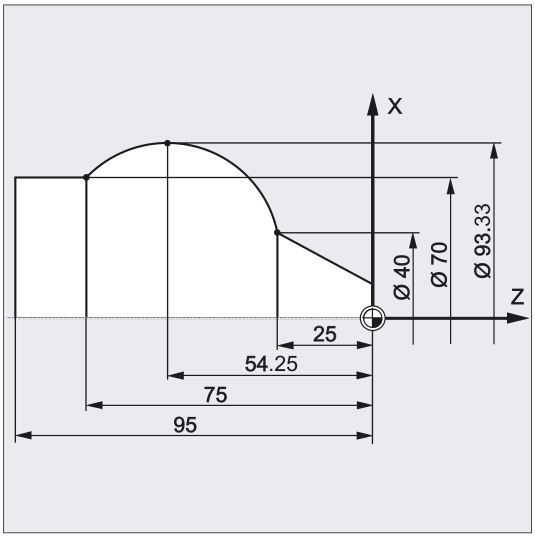
Example of milling



In order to machine an inclined circular groove, a circle is described by specifying the intermediate point with 3 interpolation parameters, and the end point with 3 coordinates.

```
N10 G0 G90 X130 Y60 S800 M3              ;Approach starting point
N20 G17 G1 Z-2 F100                      ;Tool infeed
N30 CIP X80 Y120 Z-10                    ;Circle end point and intermediate point:
 I1= IC(-85.35)J1=IC(-35.35) K1=-6       ;Coordinates for all three geometry axes
N40 M30                                  ;End of program
```

Example of turning



```
N125 G1 X40 Z-25 F0.2
N130 CIP X70 Z-75 I1=IC(26.665)
 K1=IC(-29.25)
or
N130 CIP X70 Z-75 I1=93.33 K1=-54.25
N135 G1 Z-95
```

4.11 Circular interpolation with tangential transition (CT)

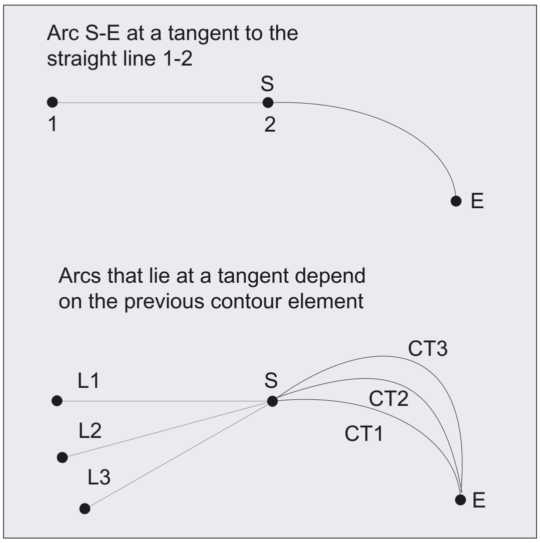
Function

The Tangential transition function is an expansion of the circle programming.

The circle is defined by

- the start and end point and
- the tangent direction at the start point.

The G code CT produces an arc that lies at a tangent to the contour element programmed previously.


Arc S-E at a tangent to the straight line 1-2
S
1
2
E
Arcs that lie at a tangent depend on the previous contour element
CT3
L1
S
CT2
L2
CT1
L3
E

Determining the direction of the tangent

The direction of tangent at the start point of a CT block is determined from the end tangent of the programmed contour of the previous block with a traversing movement.

Any number of blocks without traversing information may lie between this block and the current block.

Programming

```
CT X… Y… Z…
```

Parameters

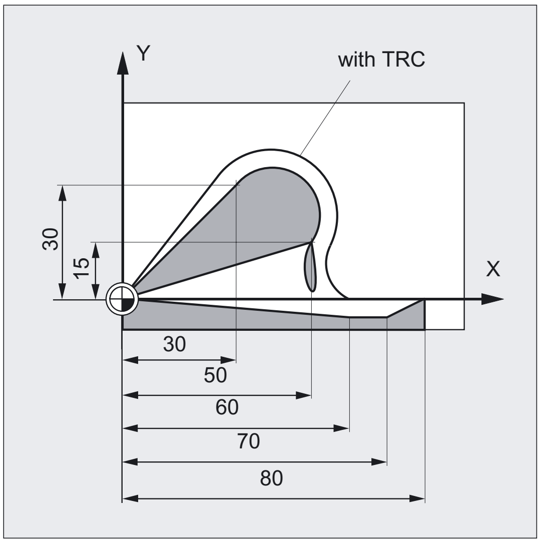
`CT`	`Circle with tangential transition`
`X Y Z`	`End point in Cartesian coordinates`

Note

CT is modal.

As a rule, the circle is uniquely defined by the direction of the tangent, as well as the start and end points.

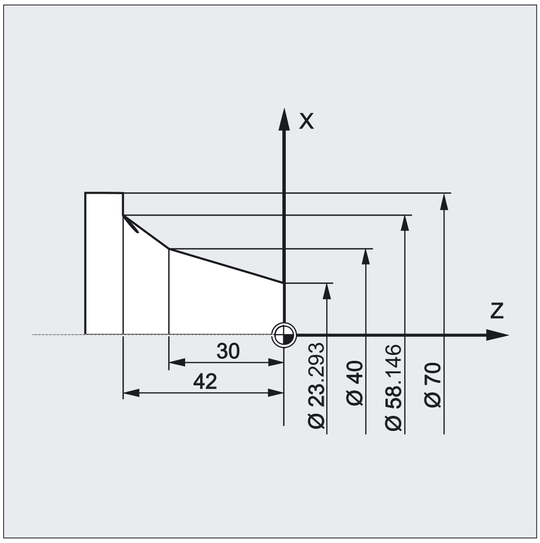
Example of milling



Milling a circular arc with CT following a straight line:

```
N10 G0 X0 Y0 Z0 G90 T1 D1
N20 G41 X30 Y30 G1 F1000                 ;Activate tool radius compensation (TRC)
N30 CT X50 Y15                           ;Program circle with tangential
                                         ;transition
N40 X60 Y-5
N50 G1 X70
N60 G0 G40 X80 Y0 Z20
N70 M30
```

Example of turning



```
N110 G1 X23.293 Z0 F10
N115 X40 Z-30 F0.2
N120 CT X58.146 Z-42                     ;Program circle with tangential
                                         ;transition
N125 G1 X70
```

Description

In the case of splines, the tangential direction is defined by the straight line through the last two points. This direction is generally not the same as the direction at the end point of the spline with A and C splines with active ENAT or EAUTO.

The transition of B splines is always tangential, the tangent direction is defined as for A or C splines and active ETAN.

Frame change

If a frame change takes place between the block defining the tangent and the CT block, the tangent is also subjected to this change.

Limit case

If the extension of the start tangent runs through the end point, a straight line is produced instead of a circle (limit case: circle with infinite radius). In this special case, TURN must either not be programmed or the value must be TURN=0.

Note

When the values tend towards this limit case, circles with an unlimited radius are produced and machining with TURN unequal 0 is generally aborted with an alarm due to violation of the software limits.

Position of the circle plane

The position of the circle plane depends on the active plane (G17-G19).

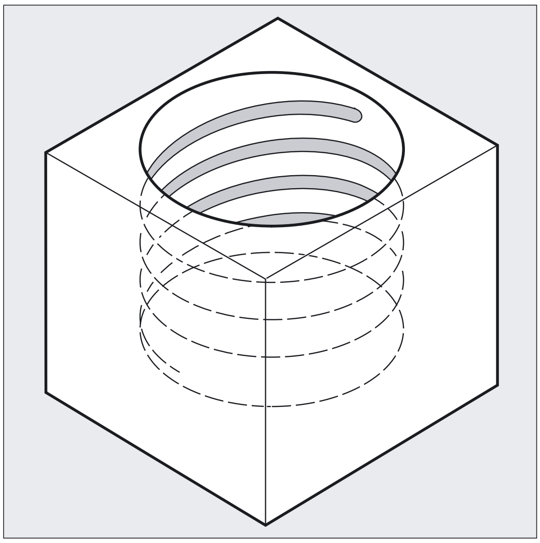
If the tangent of the previous block does not lie in the active plane, its projection in the active plane is used.

If the start and end points do not have the same position components perpendicular to the active plane, a helix is produced instead of a circle.

4.12 Helical interpolation (G2/G3, TURN=)

Function

Helical interpolation (helix interpolation) can be used to manufacture threads and oil grooves, for example.



In helical interpolation, two movements are superimposed and executed in parallel:

- a horizontal circular movement, on which
- a vertical linear movement is superimposed.

Programming

```
G2/G3 X… Y… Z… I… J… K… TURN=
```

Or

```
G2/G3 X… Y… Z… I… J… K… TURN=
```

Or

```
G2/G3 AR=… I… J… K… TURN=
```

Or

```
G2/G3 AR=… X… Y… Z… TURN=
```

Or

```
G2/G3 AP… RP=… TURN=
```

Parameters

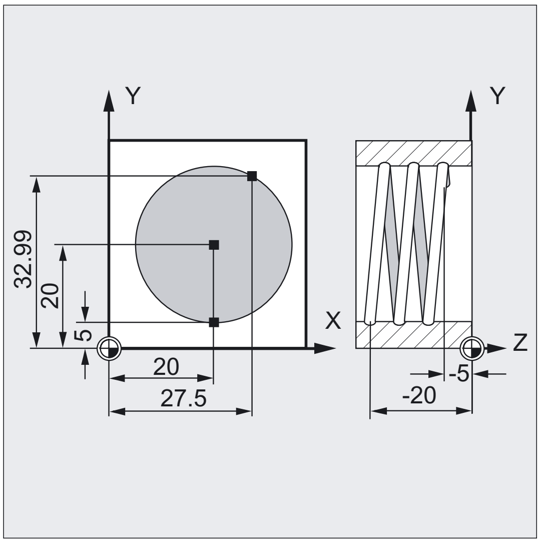
G2	Travel on a circular path in clockwise direction
G3	Travel on a circular path in counterclockwise direction
X Y Z	End point in Cartesian coordinates
I J K	Circle center point in Cartesian coordinates
AR	Opening angle
TURN=	Number of additional circle passes within the range 0 to 999
AP=	Polar angle
RP=	Polar radius

Note

G2 and G3 are modal.

The circular movement is performed on the axes specified by the working plane.

Example

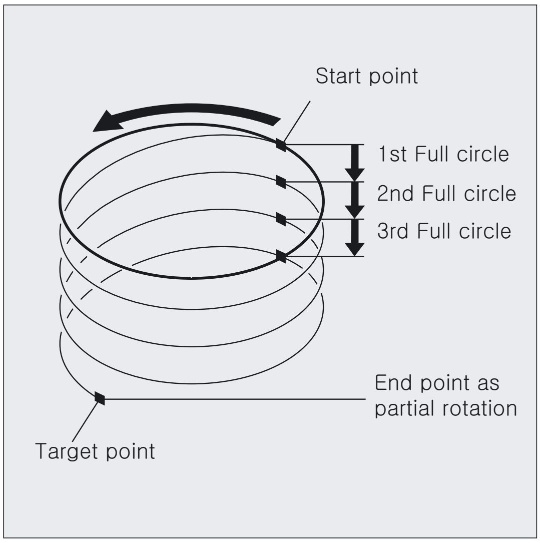


```
N10 G17 G0 X27.5 Y32.99 Z3          ;Approach start position
N20 G1 Z-5 F50                      ;Tool infeed
N30 G3 X20 Y5 Z-20 I=AC(20)         ;Helix with following parameters: Execute
 J=AC (20) TURN=2                   ;2 full circles from start position,
                                    ;then approach end point
N40 M30                             ;End of program
```

Sequence of motions

1. Approach starting point
2. With TURN= execute the full circles programmed
3. Approach circle end position, e.g., as part rotation
4. Execute steps 2 and 3 across the infeed depth.

The pitch, with which the helix is to be machined is calculated from the number of full circles plus the programmed circle end position (executed across the infeed depth).



Programming the end point for helical interpolation

Please refer to circular interpolation for a detailed description of the interpolation parameters.

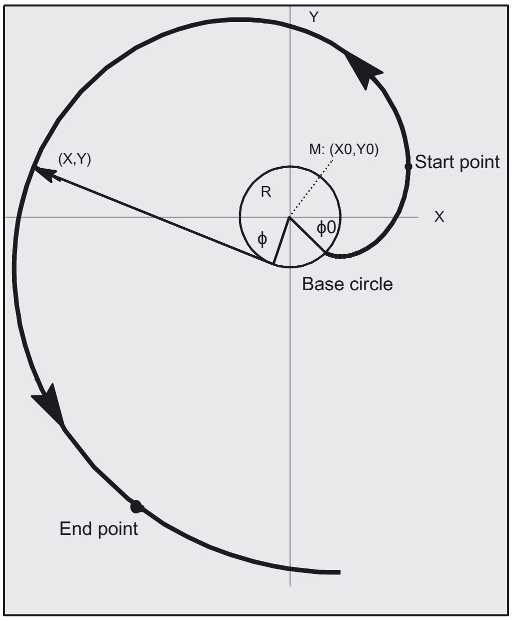
Programmed feedrate

For helical interpolation, it is advisable to specify a programmed feedrate override (CFC). You can use FGROUP to specify, which axes are to be traversed with a programmed feedrate. For more information please refer to the Path behavior section.

4.13 Involute interpolation (INVCW, INVCCW)

Function

The involute of the circle is a curve traced out from the end point on a "piece of string" unwinding from the curve. Involute interpolation allows trajectories along an involute. It takes place in the plane, in which the base circle is defined. Start and end points outside this plane result in superimposition on a curve in space, analogous to helical interpolation with circles.



When paths perpendicular to the active plane are also programmed, it is possible to traverse an involute in space (comparable to helical interpolation with circles).

Programming

```
INVCW X... Y... Z... I... J... K... CR=...
```

Or

```
INVCCW X... Y... Z... I... J... K... CR=...
```

Or

```
INVCW I... J... K... CR=... AR=...
```

Or

```
INVCCW I... J... K... CR=... AR=...
```

Parameters

INVCW	Travel on an involute in clockwise direction
INVCCW	Travel on an involute path in counterclockwise direction
X Y Z	End point in Cartesian coordinates
I J K	Center point of base circle in Cartesian coordinates
CR=	Radius of base circle
AR=	Opening angle (angle of rotation)

Supplementary condition

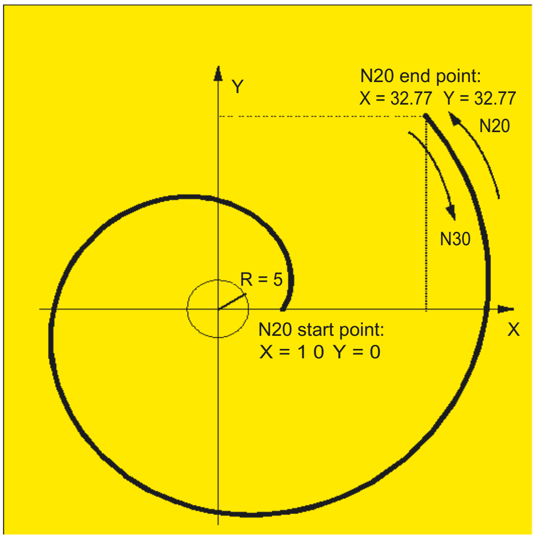
Both the start point and the end point must be outside the area of the base circle of the involute (circle with radius CR around the center point determined by I, J and K). If this condition is not fulfilled, an alarm is generated and the program run aborted.

Note

For more information about machine data and supplementary conditions that are relevant to involute interpolation, please see **References:** /FB1/, A2 section "Settings for involute interpolation".

Example of counterclockwise involute and back as clockwise involute

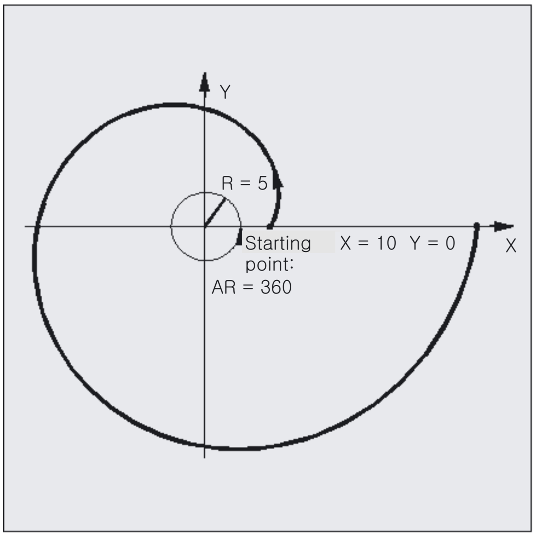
Counterclockwise involute according to programming method 1 from start to end point and back again (clockwise involute)



```
N10 G1 X10 Y0 F5000                     ;Approach start position
N15 G17                                 ;Select X/Y plane
N20 INVCCW X32.77 Y32.77 CR=5 I-10 J0   ;E. counterclockwise, end point,
                                        ;radius, center point relative to
                                        ;start point
N30 INVCW X10 Y0 CR=5 I-32.77 J-32.77   ;Start point is end point from N20
                                        ;End point is start point from N20,
                                        ;radius, center point relative to new
                                        ;start point is equal to previous
                                        ;center point
...
```

Example of counterclockwise involute with end point over angle of rotation

Specification of end point via angle of rotation

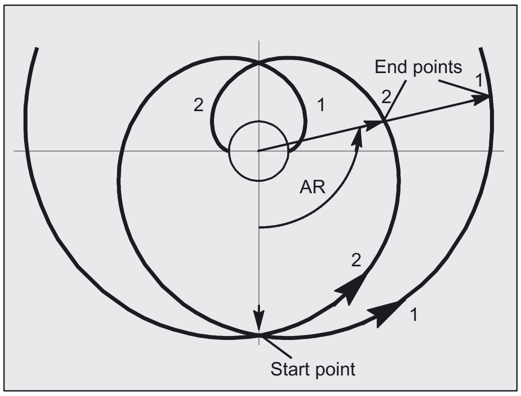


```
N10 G1 X10 Y0 F5000                    ;Approach start position
N15 G17                                ;Select X/Y plane
N20 INVCCW CR=5 I-10 J0 AR=360         ;Counterclockwise involute, away from
                                       ;base circle (pos. angle setting) with
                                       ;one full rotation
...
```

Description

Programming methods

5. Direct programming of the end point with X, Y or X, Y, Z
6. Programming of the angle of rotation between the start and end vectors with AR=angle (cf. also programming of the arc angle when programming circles). If the angle of rotation is positive (AR > 0), the path on the involute moves away from the base circle; with a negative angle of rotation (AR < 0), the path on the involute moves towards the base circle. The maximum angle of rotation for AR < 0 is restricted by the fact that the end point must always lie outside the base circle.



Options 1. and 2. are mutually exclusive. Only one of these notations may be used each block.

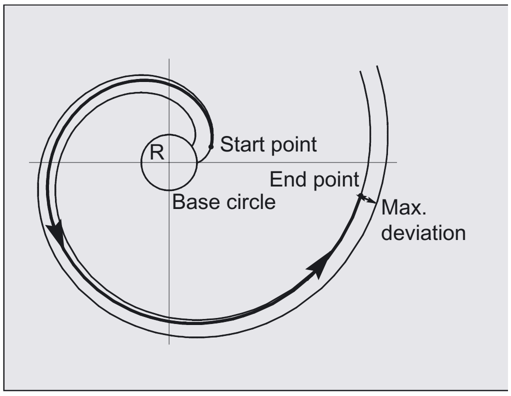
Further information

There are further options when the angle of rotation is programmed with AR. Two different involutes can be implemented (see diagram) by specifying the radius and center point of the base circle as well as the start point and direction of rotation (INVCW/INVCCW). The selected path must be defined unambiguously by the sign of the angle.

The two involutes defined by the start point and the base circle are shown in the diagram above. In this case, end point 1 is approached when AR > 0 is programmed and end point 2 when AR < 0 is programmed.

Accuracy

If the programmed end point does not lie exactly on the involute defined by the start point and base circle, interpolation takes place between the two involutes defined by the start or end point (see diagram). The maximum deviation of the end point is determined by a machine data. If the programmed end point deviates more in the radial direction than the MD setting, an alarm is generated and the program run aborted.



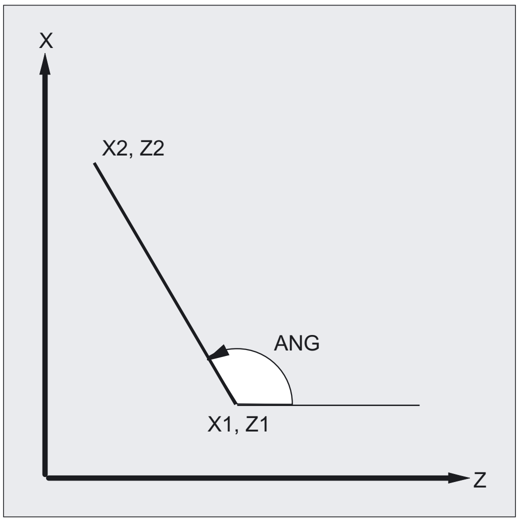
4.14 Contour definitions

4.14.1 Straight line with angle (X2... ANG...)

Function

The end point is defined through specification of

- the angle ANG and
- one of the two coordinates X2 or Z2.



Programming

```
X2… ANG…
```

Parameters

X2 or Z2	End point in Cartesian coordinates X or Z
ANG	Angle

Machine manufacturer

The names for angle (ANG), radius (RND) and chamfer (CHR) can be set in MD, see /FBFA/FB ISO Dialects.

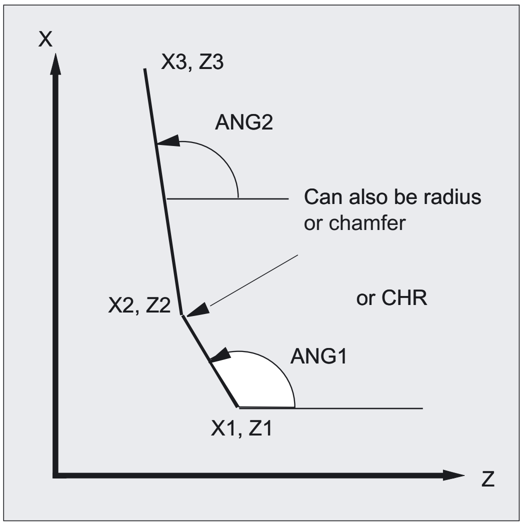
Example

```
N10 X5 Z70 F1000 G18                        ;Approach start position
N20 X88.8 ANG=110 or (Z39.5 ANG=110)        ;Straight line with specified angle
N30 ...
```

4.14.2 Two straight lines (ANG1, X3... Z3... ANG2)

Function

The intersection of the two straight lines can be designed as a corner, curve or chamfer. The end point of the first of the two straight lines can be programmed by defining the coordinates or specifying the angle.



Programming

```
ANG1…
X3… Z3… ANG2…
```

Or

```
X1… Z1…
X3… Z3…
```

Parameters

ANG1=	Angle of the first straight line
ANG2=	Angle of the second straight line
CHR	Chamfer
X1, Z1=	Start coordinates
X2, Z2=	Intersection of the two straight lines
X3=, Z3=	End point of the second straight line

Machine manufacturer

The names for angle (ANG), radius (RND) and chamfer (CHR) can be set in MD, see /FBFA/FB ISO Dialects.

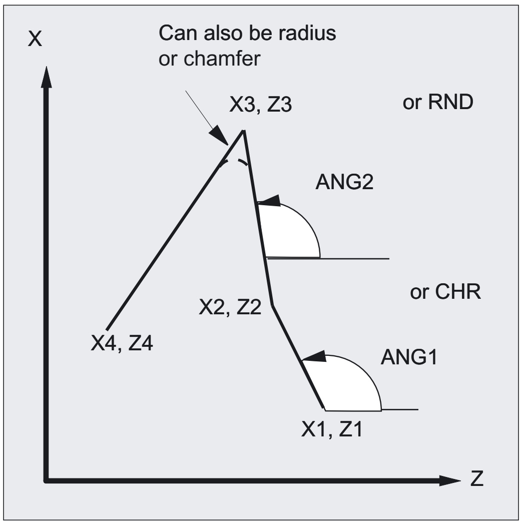
Example

```
N10 X10 Z80 F1000 G18                ;Approach start position
N20 ANG1=148.65 CHR=5.5              ;Straight line with specified angle and
                                     ;chamfer
N30 X85 Z40 ANG2=100                 ;Straight line with specified angle and end
                                     ;point
N40 ...
```

4.14.3 Three straight lines (ANG1, X3... Z3... ANG2, X4... Z4...)

Function

The intersection of the straight lines can be designed as a corner, a curve, or a chamfer. The end point of the third straight line must always be programmed as Cartesian.



Programming

```
ANG1…
X3… Z3… ANG2…
X4… Z4…
```

Or

```
X2… Z2…
X3… Z3…
X4… Z4…
```

Parameters

ANG, ANG2=	Angle of the first/second straight line relative to the abscissa
CHR	Chamfer
RND	Rounding
X1, Z1	Start coordinates of the first straight line
X2, Z2	End point coordinates of the first straight line or starting point of the second straight line.
X3, Z3	End point coordinates of the second straight line or starting point of the third straight line
X4=, Z4=	Endpoint coordinates of the third straight line

Machine manufacturer

The names for angle (ANG), radius (RND) and chamfer (CHR) can be set in MD, see /FBFA/FB ISO Dialects.

Example

```
N10 X10 Z100 F1000 G18                ;Approach start position
N20 ANG1=140 CHR=7.5                  ;Straight line with specified angle and
                                      ;chamfer
N30 X80 Z70 ANG2=95.824 RND=10        ;Straight line on intersection with
                                      ;specified angle and ;rounding
N40 X70 Z50                           ;Straight line on end point
```

4.14.4 End point programming with angle

Function

If the address letter A appears in an NC block, either none, one or both of the axes in the active plane may also be programmed.

Number of programmed axes

- If none of the axes in the active plane is programmed, the block is either the first or second block of a contour consisting of two blocks.
  If it is the second block in this type of contour, it indicates that the start and end points in the active plane are identical. The contour then comprises at most a motion perpendicular to the active plane.
- If exactly one axis in the active plane is programmed, then it is either a single straight line, whose end point is uniquely defined by the angle and programmed Cartesian coordinate, or it is the second block in a contour definition comprising two blocks. In the latter case, the missing coordinate is set to match the last reached (modal) position.
- If two axes are programmed in the current plane, it is the second block of a contour that consists of two blocks. If the current block was not preceded by a block with angle programming and no programmed axes in the current plane, the block in question is not permissible.

Angle A must only be programmed for linear or spline interpolation.

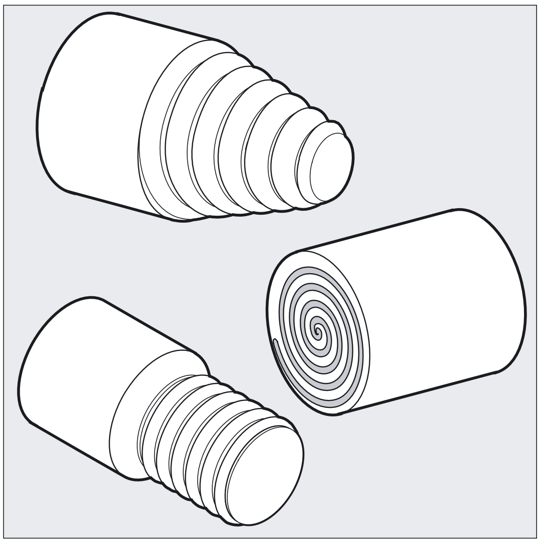
4.15 Thread cutting with constant lead (G33)

Function

With G33 three types of thread

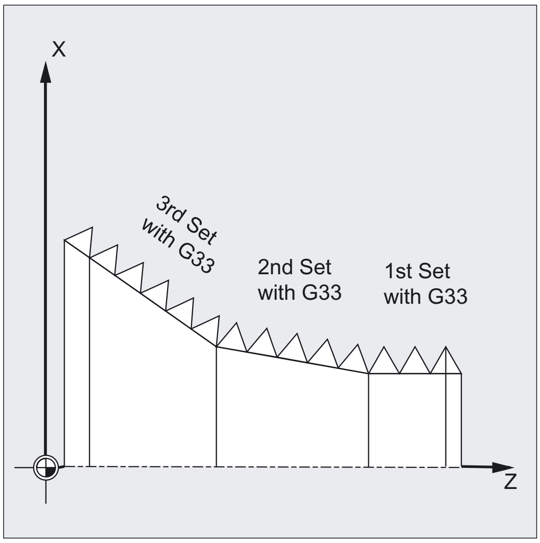
- Cylinder thread
- Face thread
- Taper thread

can be produced with single or multiple threads as right-hand or left-hand thread.



Thread chains

By programming several G33 blocks consecutively, you can align several sets of threads in a sequence. With G64 continuous-path mode, the blocks are interconnected in a look ahead velocity control so that no speed jumps are produced.



Right-hand/left-hand threads

Right-hand or left-hand threads are set according to the spindle direction:

M3: Clockwise

M4: CCW rotation

Programming

Cylinder thread

```
G33 Z… K … SF=…
```

Face thread

```
G33 X… I… SF=…
```

Taper thread

```
G33 X… Z… K… SF=…
```

Or

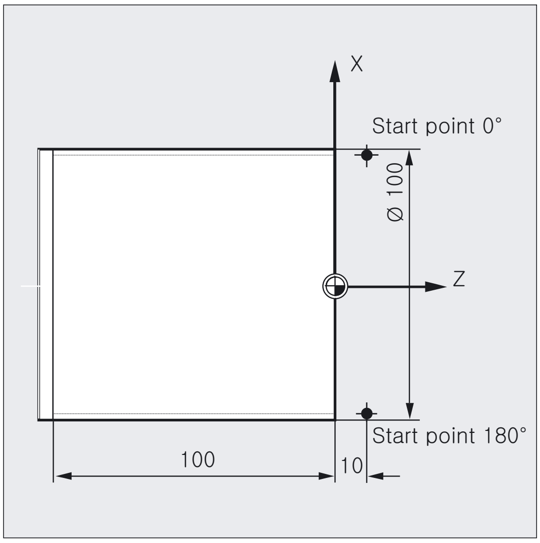
```
G33 X… Z… I… SF=…
```

Parameters

G33	Thread cutting with constant speed
X Y Z	End point in Cartesian coordinates
I	Thread lead in X direction
J	Thread lead in Y direction
K	Thread lead in Z direction
Z	Longitudinal axis
X	Transverse axis
Z... K...	The thread length and thread lead for cylinder threads
X... I...	Thread diameter and thread lead for face threads
I... K...	The dominant direction share for tape thread in X or Z
K (taper thread)	Lead angle <45°, thread lead in longitudinal direction
I (taper thread)	Lead angle >45°, thread lead in transverse direction
I... or K...	I or K can be stated at thread lead = 45°
SF=	Starting point offset, only needed for multiple threads

Example of double cylinder thread with start point offset

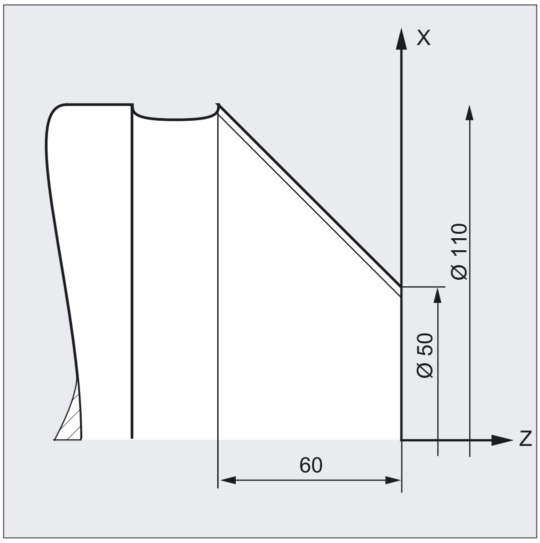
Machining a double cylindrical thread in offset steps with starting point offset 180°.



```
N10 G1 G54 X99 Z10 S500 F100 M3          ;Zero offset, approach
                                         ;start point, spindle on
N20 G33 Z-100 K4                         ;Cylindrical thread: end point in Z
N30 G0 X102                              ;Retract to starting position
N40 G0 Z10
N50 G1 X99
N60 G33 Z-100 K4 SF=180                  ;2. cut: Starting point offset 180°
N70 G0 X110                              ;Retract tool
N80 G0 Z10                               ;End of program
N90 M30
```

Example of taper thread with angle less than 45°

Machining a taper thread



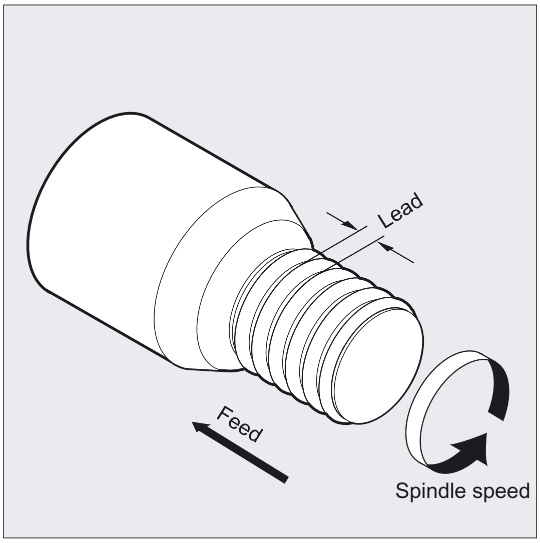
```
N10 G1 X50 Z0 S500 F100 M3              ;Approach starting point, activate spindle
N20 G33 X110 Z-60 K4                    ;Taper thread: End point on Z and X,
                                        ;lead K in Z direction, since angle < 45°
N30 G0 Z0 M30                           ;Retraction, end of program
```

Requirements

Equipment required: speed-controlled spindle with position measurement system.

Operating principle

The control calculates the required feedrate from the programmed spindle speed and the thread lead. The turning tool traverses across the length of the thread in the longitudinal and/or facing direction at this feedrate. The feedrate F is not considered for G33, the limitation to maximum axis speed (rapid traverse) is monitored by the control.

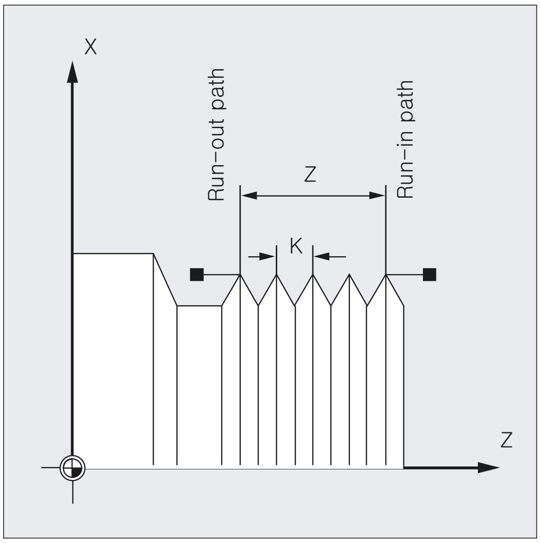


Cylinder thread

A cylinder thread is described by the thread length and thread lead.

The thread length is entered in absolute or incremental dimensions with one of the Cartesian coordinates X, Y, or Z. The Z direction is preferred on turning machines. Allowance must also be made for the run-in and run-out paths, across which the feed is accelerated or decelerated.

The thread lead is entered at addresses I, J, K, on turning machines preferentially with K.

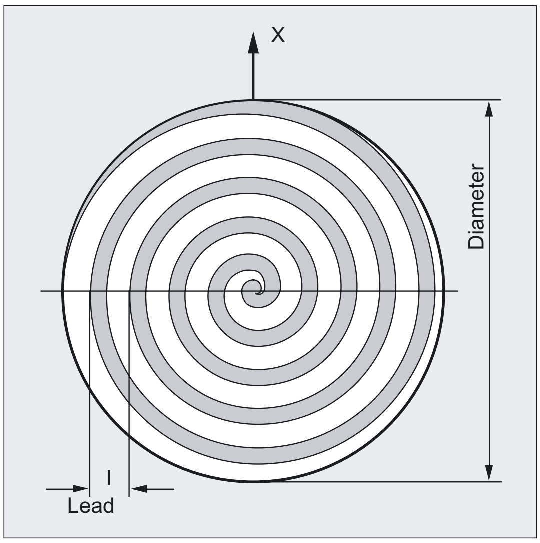


Face thread

The face thread is described by

- thread diameter, preferentially in X direction and
- thread lead, preferentially with I.

Otherwise, the procedure is the same as for cylindrical threads.

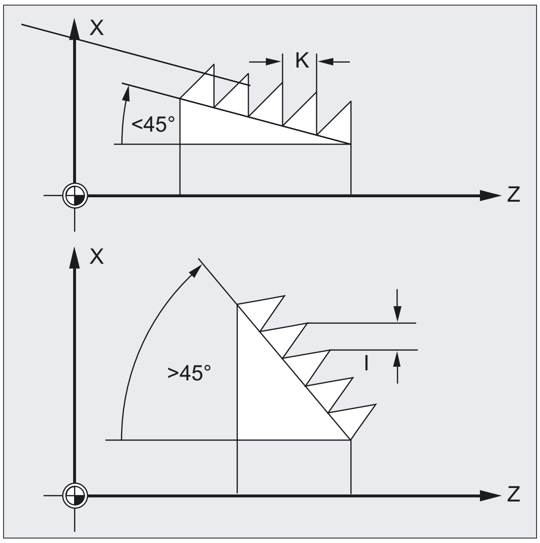


Taper thread

The taper thread is described by the end point in the longitudinal and facing direction (taper contour) and the thread lead.

The taper contour is entered in Cartesian coordinates X, Y, Z in absolute or incremental dimensions - preferentially in the X and Z direction for machining on turning machines. Allowance must also be made for the run-in and run-out paths, across which the feed is accelerated or decelerated.

The parameter for the lead is based on the taper angle (calculated from the longitudinal axis lead angle <45° to the outside of the taper lead angle >45°).



Start point offset SF - production of multi-turn threads

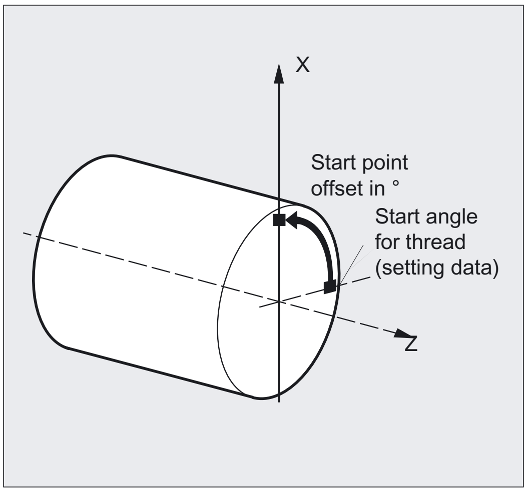
Threads with offset cuts are programmed by specifying starting point offsets in the G33 block.

The start point offset is specified as an absolute angle position at address SF=. The associated setting data is changed accordingly.

Example: SF=45

Meaning: Start offset 45°

Value range: 0.0000 to 359.999 degrees



Note

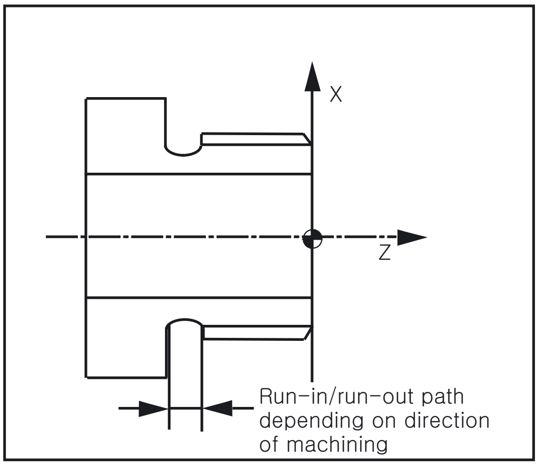
If no starting point offset is specified, the "starting angle for thread" defined in the setting data is used.

4.15.1 Programmable run-in and run-out paths (DITS, DITE)

Function

The commands **DITS** (Displacement Thread Start) and **DITE** (Displacement Thread End) can be used to define the path ramp for acceleration and deceleration, in order to modify the feedrate if the tool run-in and run-out paths are too short:

- **Run-in path too short**:
  The band at the thread run-in provides insufficient space for the tool start ramp - a shorter ramp must therefore be defined with DITS.
- **Run-out path too short**
  The band at the thread run-out provides insufficient space for the tool deceleration ramp, giving rise to **danger of collision** between the workpiece and the tool edge.
  A shorter tool deceleration ramp can be defined with DITE; however, a collision can still occur.
  Remedy: Program a shorter thread, reduce the spindle speed.



Programming

```
DITS=value
DITE=value
```

Parameters

DITS	Thread run-in path
DITE	Thread run-out path
Value	Specification of the run-in and run-out path: -1.0,...n

Note

Only paths, and not positions, are programmed with DITS and DITE.

Machine manufacturer

The DITS and DITE commands are related to the setting data SD 42010: THREAD_RAMP_DISP[0,1], in which the programmed paths are written. If no run-in/deceleration path is programmed before or in the first thread block, the value is determined by the setting in SD 42010, see:
References: /FB/Functional description, V1 Feeds.

Example

```
N...
N40 G90 G0 Z100 X10 SOFT M3 S500
N50 G33 Z50 K5 SF=180 DITS=1 DITE=3          ;Start of corner rounding with Z=53
N60 G0 X20
```

Description

If the run-in and/or run-out path is very short, the acceleration of the thread axis is higher than the configured value. This causes an acceleration overload on the axis.

In this case, alarm 22280 "Programmed run-in path too short" is output for the thread run-in (if configured in MD 11411: ENABLE_ALARM_MASK). The alarm is purely for information and has no effect on parts program execution.

MD 10710: PROG_SD_RESET_SAVE_TAB is used to set the value written by the parts program into the corresponding setting data on RESET. The values are therefore retained after power off/on.

Note

DITE acts at the end of the thread as an approximate distance. This achieves a smooth change in the axis movement.

When a block containing command DITS and/or DITE is loaded to the interpolator, the path programmed in DITS is transferred to SD 42010: THREAD_RAMP_DISP[0] and the path programmed in DITE to SD 42010 THREAD_RAMP_DISP[1].

The programmed run-in path is handled according to the current setting (inches, metric).

4.16 Linear progressive/degressive thread pitch change (G34, G35)

Function

You can use the functions G34/G35 to produce self-cutting threads.

Both the functions G34 and G35 offer the functionality of G33, but provide the additional option of programming a pitch change under F.

Programming

```
G34 X… Y… Z… I… J… K… F…
```

Or

```
G34 X… Y… Z… I… J… K… SF=…
```

Or

```
G35 X… Y… Z… I… J… K… F…
```

Or

```
G35 X… Y… Z… I… J… K… SF=…
```

Parameters

G34	Progressive change in thread lead (tapping with linear pitch increase)
G35	Degressive change in thread lead (tapping with linear pitch decrease)
X Y Z	End point in Cartesian coordinates
I	Thread pitch in X direction
J	Thread lead in Y direction
K	Thread lead in Z direction
F	Thread lead change (in mm/rev^2) If you already know the initial and final lead of a thread, you can calculate the lead change to be programmed according to the following equation: $F = \frac{\lvert k^2_e - k^2_a \rvert}{2*I_G}$ $[\text{mm/rev}^2]$ The identifiers have the following meanings: Ke: thread lead of axis target point coordinate in [mm/rev] Ka: initial thread pitch (progr. under I, J, and K) in [mm/rev] I_G: Thread length in [mm]
SF=	Starting point offset, only needed for multiple threads

Example of lead decrease

N1608 M3 S10	;Spindle speed
N1609 G0 G64 Z40 X216	;Approach start point and thread
N1610 G33 Z0 K100 SF=R14	;With constant pitch 100 mm/rev
N1611 G35 Z-200 K100 F17.045455	;Lead decrease 17.0454 mm/rev^2 ;lead at block end 50 mm/rev
N1612 G33 Z-240 K50	;Traverse thread block without jerk
N1613 G0 X218	;
N1614 G0 Z40	;
N1615 M17	;

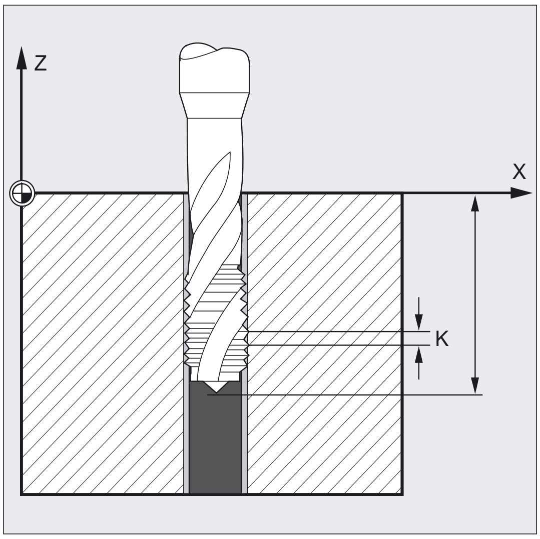
4.17 Tapping without compensating chuck (G331, G332)

Function

With G331/G332 you can rigid tap a thread.

The spindle prepared for tapping can make the following movements in position-controlled operation with distance measuring system:

- **G331** Tapping with thread lead in tapping direction up to end point
- **G332** retraction movement with the same lead as G331



Right-hand/left-hand threads

Right-hand or left-hand threads are defined in axis mode by the sign qualifying the lead:

- Positive lead, clockwise (same as M3)
- Negative lead, counterclockwise (same as M4)

The desired speed is also programmed at address S.

Programming

```
G331 X… Y… Z… I… J… K…
```

Or

```
G332 X… Y… Z… I… J… K…
```

Parameters

G331	Tapping. Tapping is described by the drilling depth (end point of the thread) and the lead.
G332	Tapping retraction. This movement is described with the same lead as the G331 movement. The reversal in the direction of the spindle is performed automatically.
X Y Z	Drilling depth (end point) in a Cartesian coordinate Drilling in X direction, thread lead I Y direction, thread lead J Z direction, thread lead K
I	Thread pitch in X direction
J	Thread lead in Y direction
K	Thread lead in Z direction Value range of lead: ±0.001 to 2000.00 mm/revolution

Note

Both functions G331/G332 are modal.

After G332 (retraction), the next thread can be tapped with G331.

Equipment required: position-controlled spindle with position measurement system.

The spindle must be prepared for tapping with SPOS/SPOSA. It does not work in axis operation, but as position-controlled spindle, see section Feed control and spindle movement "Position-controlled spindle operation".

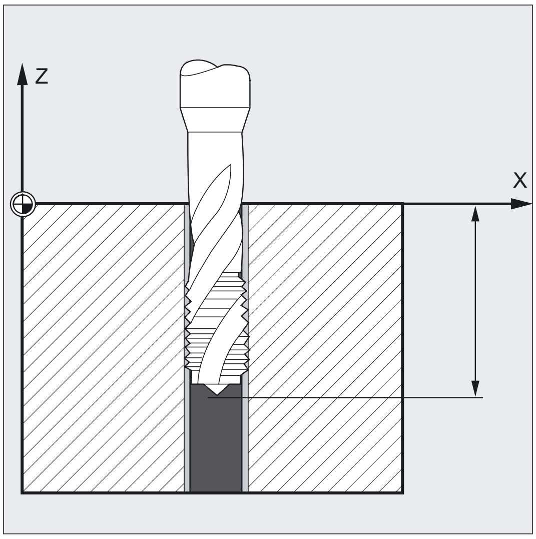
4.18 Tapping with compensating chuck (G63)

Function

You can use G63 to tap threads with compensating chuck. The following are programmed:

- Drilling depth in Cartesian coordinates
- Spindle speed and spindle direction
- Feed

The chuck compensates for any deviations occurring in the path.



Retraction movement

Also programmed with G63, but with the reverse direction of spindle rotation.

Programming

```
G63 X… Y… Z…
```

Parameters

G63	Tapping with compensating chuck.
X Y Z	Drilling depth (end point) in a Cartesian coordinate

Note

G63 is non-modal.

The last programmed interpolation command G0, G1, G2, etc., is reactivated after a block with programmed G63.

Feedrate

Note

The programmed feed must match the ratio of the speed to the thread lead of the tap.

Thumb rule:

Feed F in mm/min = spindle speed S

in rpm x thread lead in mm/rev

Both the feed and the spindle speed override switch are set to 100% with G63.

Example 1

```
N10 SPOS[n]=0                           ;Prepare tapping
N20 G0 X0 Y0 Z2                         ;Approach starting point
N30 G331 Z-50 K-4 S200                  ;Tapping, drilling depth 50, lead K
                                        ;negative = direction of spindle rotation
                                        ;counterclockwise
N40 G332 Z3 K-4                         ;Retract, automatic reversal of direction
N50 G1 F1000 X100 Y100 Z100 S300 M3     ;Spindle reverts to operation in spindle
                                        ;mode
N60 M30                                 ;End of program
```

Example 2

In this example, an M5 thread is to be drilled. The lead of an M5 thread is 0.8 (specified in table).

With a selected speed of 200 rpm, the feed F is 160 mm/min.

```
N10 G1 X0 Y0 Z2 S200 F1000 M3             ;Approach starting point, activate
                                          ;spindle
N20 G63 Z-50 F160                         ;Tap, drilling depth 50
N30 G63 Z3 M4                             ;Retract, programmed reversal of
                                          ;direction
N40 M30                                   ;End of program
```

4.19 Stop with thread cutting (LFOF, LFON, LFTXT, LFWP, LFPOS)

4.19.1 Retraction for thread cutting (LFOF, LFON, LIFTFAST, DILF, ALF)

Function

The function produces a nondestructive interruption with thread cutting (G33). The function cannot be used with tapping (G33). With mixed use of both G33 functions, the response can be parameterized for NC Stop/NC Reset via the machine data. If tapping was interrupted, then there are several possibilities of programming the fast retraction to a certain lift position. Both the length of the retraction path and the retraction direction can be defined as target position.

Programming

`LFON`

Or

`LFOF`

Where

`LIFTFAST=` (if enabled as option)

Or

`DILF=`

Or

`ALF=`

Parameters

LFON	Enable fast retraction for thread cutting (G33)
LFOF	Disable fast retraction for thread cutting (G33)
LIFTFAST	Fast retraction option acts with LFON in every retraction direction
DILF	Determine retraction path (length)
ALF	Define retraction direction for plane to be executed (LFTXT)

Note

LFON or LFOF can always be programmed, they are evaluated only during thread cutting (G33).

Example of enabling fast retraction in tapping

```
N55 M3 S500 G90 G18                             ;Active machining plane
...                                             ;Approach start position
N65 MSG ("thread cutting")                      ;Tool infeed
MM_THREAD:
N67 $AC_LIFTFAST=0                              ;Reset before beginning of
                                                ;thread
N68 G0 Z5
N68 X10
N70 G33 Z30 K5 LFON DILF=10 LFWP ALF=3          ;Enable fast retraction for thread
                                                ;cutting
                          Retraction path =10mm, retraction plane Z/X (due to G18)
                          Retraction direction -X (with ALF=3; retraction direction
                          +X)
N71 G33 Z55 X15 K5
N72 G1                                          ;Deselect thread cutting
N69 IF $AC_LIFTFAST GOTOB MM_THREAD             ;If thread cutting was ;interrupted
N90 MSG ("")
...
N70 M30
```

Example of deactivating fast retraction before tapping.

```
N55 M3 S500 G90 G0 X0 Z0
...
N87 MSG ("tapping")
N88 LFOF                                        ;Deactivate fast retraction before
                                                ;tapping.
N89 CYCLE...                                    ;Tapping cycle with G33
N90 MSG ("")
...
N99 M30
```

Trigger criteria for retraction

- Fast inputs, programmable with SETINT LIFTFAST (if LIFTFAST option is enabled)
- NC Stop/NC Reset

If fast retraction is enabled with LFON, it is active for every movement

Retraction path (DILF)

The retraction path can be defined in the machine data or by programming. After NC Reset, the value in MD 21200: LIFTFAST_DIST is still active.

Retraction direction (ALF)

The retraction direction is controlled in conjunction with **ALF** with vocabulary words **LFTXT, LFWP** and **LFPOS.** With **LFTXT**, the retraction is defined in the tool direction for ALF=1. LFTXT (tangential lifting in tool direction) is set as default. See "Lifting in retraction with LFTXT, LFWP, LFPOS, POLF, POLFMADK and POLFMLIN".

4.19.2 Lifting on retraction (LFTXT, LFWP, LFPOS, POLF, POLFMASK; POLFMLIN)

Function

With the LFTXT, LFWP, LFPOS; POLF commands you can selectively program on lifting the retraction processing from the lift movement up to the lift position. The following are programmed:

- Retraction direction from the path tangent or the active working plane
- Retraction direction toward position programmed
- Absolute retraction position

In this case, the axis relation of the programmed path or the linear retracting movement is not always guaranteed for a certain period of time. The linear relation cannot always be produced according to the dynamic behavior of all participating axes up to reaching the lift position. Axes can be enabled for independent retraction to axis position and to axis position with linear relation.

Programming

```
LFTXT
```

Or

```
LFWP
```

Or

```
LFPOS
```

Or

```
POLF[geo axis name | machine axis name]=
```

Or

```
POLFMASK(axisname1, axisname2, etc.)
```

Or

```
POLFMLIN
```

Parameters

LFTXT	Retraction direction on lifting from the path tangent, default
LFWP	Retraction direction from the active working plane G17, G18, G19
LFPOS	Retraction direction toward position programmed with POLF
POLF	Absolute retraction position of axis, incl. incremental with IC (value) POLF is modal.
POLFMASK	Enable for axes for independent retraction to absolute position
POLFMLIN	Enable for axes for retraction to absolute position in linear relationship. See also functional description Fb3, M3
X, Y, Z	Geometry axes in POLF are interpreted as position in the tool coordinate system (TCS).
X1, Y1, Z1	Machine axes in POLF are interpreted as position in the machine coordinate system (MCS).

Example

Here, the path interpolation of X is suppressed in the event of a stop and a motion executed to position POLF[X] at maximum velocity instead. The motion of the other axes continues to be determined by the programmed contour or the thread lead and spindle speed.

```
N10 G0 G90 X200 Z0 S200 M3
N20 G0 G90 X170
N22 POLF[X]=210 LFPOS
N23 POLFMASK(X)                           ;Activate (enable)
                                          ;rapid lift of
                                          ;the axis X
N25 G33 X100 I10 LFON
N30 X135 Z-45 K10
N40 X155 Z-128 K10
N50 X145 Z-168 K10
N55 X210 I10
N60 G0 Z0 LFOF
N70 POLFMASK()                            ;Block retraction for all axes
M30
```

Description

The retraction direction in connection with **ALF** is controlled using the following keywords:

- **LFTXT**
  The plane in which the fast retraction is executed is calculated from the path tangent and the tool direction (default setting).
- **LFWP**
  The plane in which the fast retraction is executed is the active working plane.
- **LFPOS**
  Retraction of the axis declared with POLFMASK to the absolute axis position programmed with POLF. See also NC controlled retraction in the functional description FB3, M3.
  ALF has no influence on the retraction direction on several axes and also on several axes in linear relation.

The direction is programmed as before in discrete steps of 45 degrees with ALF in the plane of the retraction motion. With **LFTXT**, the retraction is defined in the tool direction for ALF=1.

With **LFWP** the direction in the working plane is derived from the following assignment:

- **G17**: X/Y plane
  ALF=1 Retraction in X direction
  ALF=3 Retraction in Y direction
- **G18**: Z/X plane
  ALF=1 Retraction in Z direction
  ALF=3 Retraction in X direction
- **G19**: Y/Z plane
  ALF=1 Retraction in Y direction
  ALF=3 Retraction in Z direction

Retraction velocity

Retraction with maximum axis velocity.
Can be configured via machine data.

The maximum permissible acceleration/jerk values are used for traversing; they are configured via the machine data.

Note

POLF with POLFMASK/POLFMLIN are not restricted to thread cutting applications. See also functional description FB3, M3.

4.20 Approaching a fixed point (G75)

Function

G75 can be used to approach fixed points, such as tool change locations, loading points, pallet changing points, etc.

The positions of the individual points are specified in the machine coordinate system and stored in the machine parameters.

You can approach these positions from any NC program, irrespective of the current tool or workpiece position.

Programming

```
G75 FP= X1=0 Y1=0 Z1=0 U1=0 …
```

Parameters

G75	Fixed point approach The fixed point approach is described by a fixed point and axes, which are to be traversed to the fixed point FP.
FP=	Number of fixed point to be approached **Number of the fixed point FP=...** If no fixed point number is specified, fixed point 1 is automatically approached.
X1= Y1= Z1=	Machine axes to be traversed to the fixed point **Machine axis addresses X1, Y1 ...** Here, you specify with value 0 the axes, with which the point is to be approached simultaneously. Each axis traverses at the maximum axial velocity.

Note

Two fixed point positions per machine axis can be specified in the machine parameters.

G75 is non-modal.

G75 "Approach fixed point" applies all offset values (DRF, external ZO and overlaid motion). The fixed point corresponds to the actual value in the MCS.

Changes to the DRF and external zero offset while the G75 block is being preprocessed and executed in the main run are not applied. You should prevent this problem by programming STOPPRE in front of the G75 block.

Kinematic transformation must be deselected before fixed point approach is performed.

Example

The tool change location is a fixed point, which is defined in the machine data.

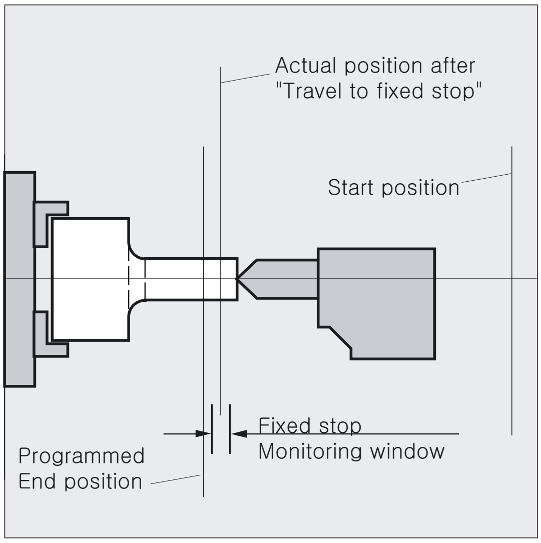
This point can be approached in any NC program with G75.

```
N10 G75 FP=2 X1=0 Y1=0 Z1=0           ;Retract from fixed point 2 on X, Y and
                                      ;Z, e.g., for tool change
N20 G75 X1=0                          ;Approach fixed point X1
N30 M30                               ;End of program
```

4.21 Travel to fixed stop (FXS, FXST, FXSW)

Function

The "travel to fixed stop" (FXS = Fixed Stop) function can be used to build defined forces for clamping workpieces, such as those required for tailstocks, quills and grippers. The function can also be used for the approach of mechanical reference points.



With sufficiently reduced torque, it is also possible to perform simple measurement operations without connecting a probe. The "travel to fixed stop" function can be implemented for axes as well as for spindles with axis-traversing capability.

Programming

```
FXS [axis]=…
FXST [axis]=…
FXSW [axis]=…
```

Parameters

FXS	Select/deselect "travel to fixed stop" function = select; 0 = deselect
FXST	Setting clamping torque Specification in % of maximum drive torque, parameter optional
FXSW	Window width for fixed stop monitoring in mm, inches or degrees; parameter optional
[axis]	Machine axis name Machine axes (X1, Y1, Z1, etc.) are programmed. (See **machine manufacturer's** specifications).

Note

The commands are modal. Addresses FXST and FXSW are optional: If no parameter is specified, the last programmed value or the value set in the relevant machine data applies.

Example of activating travel to fixed stop FXS=1

The movement to the destination point can be described as a path or positioning axis movement. With positioning axes, the function can be performed across block boundaries.

Travel to fixed stop can be performed simultaneously for several axes and parallel to the movement of other axes. The fixed stop must be located between the start and end positions.

```
X250 Y100 F100 FXS[X1]=1 FXST[X1]=12.3 FXSW[X1]=2
```

Meaning:

Axis X1 travels with feed F100 (parameter optional) to destination position X=250 mm.

The clamping torque is 12.3% of the maximum drive torque. Monitoring is performed in a 2 mm wide window.

Caution

It is not permissible to program a new position for any axis/spindle for the "Travel to fixed stop" function has already been activated.

Spindles must be switched to position-controlled mode before the function is selected.

Example of deactivating travel to fixed end stop FXS=0

Deselection of the function triggers a preprocessor stop.

Traversing movements may and should be programmed in a block with FXS=0:

```
X200 Y400 G01 G94 F2000 FXS[X1] = 0
```

Meaning:

Axis X1 is retracted from the fixed stop to position X= 200 mm.

All other parameters are optional.

Caution

The traversing movement to the retraction position must move away from the fixed stop, otherwise damage to the stop or to the machine may result.

The block change takes place when the retraction position has been reached. If no retraction position is specified, the block change takes place immediately the torque limit has been deactivated.

Example of clamping torque FXST, monitoring window FXSW

Any programmed torque limitation FXST is effective from the block start, i.e., the fixed stop is also approached at a reduced torque. FXST and FXSW can be programmed or changed at any time in the parts program:

```
FXST[X1] = 34.57
FXST[X1] = 34.57 FXSW[X1] = 5
FXSW[X1] = 5
```

The changes take effect before traversing movements in the same block.

Programming of a new fixed stop monitoring window causes a change not only in the window width but also in the reference point for the center of the window if the axis has moved prior to reprogramming. The actual position of the machine axis when the window is changed is the new window center point.

Caution

The window must be selected such that only a breakaway from the fixed stop causes the fixed stop monitoring to respond.

Description

The limit stop alarm can be suppressed from the parts program where necessary. This is done by masking the alarm in a machine data and then activating this MD by means of NEWCONF.

The "Travel to fixed stop" commands can be called from synchronized actions/technology cycles. They can be activated without initiation of a motion, the torque is limited instantaneously. As soon as the axis is moved via a setpoint, the limit stop monitor is activated.

Rise ramp

A rise ramp for the new torque limit can be defined by machine data to prevent any abrupt changes to the torque limit setting (e.g., insertion of a quill).

Link and container axes

Travel to fixed stop is also permitted for

- Link axes
- Container axes

The status of the assigned machine axis is unaffected by container switches.

References: Function description /FB2/B3, Multiple Operator Panels and NCUs

This also applies to modal torque limiting with FOCON (see "travel with limited torque/force").

Activation from synchronized actions

Example:

If the anticipated event ($R1) occurs and travel to fixed stop is not already operative, then FXS must be activated for axis Y. The torque must correspond to 10% of the rated torque value. The width of the monitoring window is set to the default.

```
N10 IDS=1 WHENEVER (($R1=1) AND ($AA_FXS[Y]==0)) DO $R1=0 FXS[Y]=1
FXST[Y]=10
```

The normal parts program must ensure that $R1 is set at the desired point in time.

Deactivation from synchronized actions

Example:

If an anticipated event ($R3) has occurred and the status "Limit stop contacted" (system variable $AA_FXS) is reached, then FXS must be deselected.

```
N13 IDS=4 WHENEVER (($R3==1) AND ($AA_FXS[Y]==1)) DO FXS[Y]=0 FA[Y]=1000
POS[Y]=0
```

Fixed stop reached

When the fixed stop has been reached:

- The distance-to-go is deleted and the position setpoint is manipulated,
- The drive torque increases to the programmed limit value FXSW and then remains constant,
- Fixed stop monitoring is activated within the specified window width.

Combinability

Note

"Measure and delete distance-to-go" ("MEAS" command) and "Travel to fixed stop" cannot be programmed in the same block.

Exception: One function acts on a path axis and the other on a positioning axis or both act on positioning axes.

Contour monitoring

Contour monitoring is not performed while "Travel to fixed stop" is active.

Positioning axes

With "Travel to fixed stop" with POSA axes, the block change takes place independently of the fixed stop movement.

Restrictions

Travel to fixed stop cannot be programmed

- for hanging axes (exception: possible on 840D with SIMODRIVE 611 digital),
- for gantry axes,
- for concurrent positioning axes, which are controlled exclusively by the PLC (FXS must be selected from the NC program).
- If the torque limit is reduced too far, the axis will not be able to follow the specified setpoint; the position controller then goes to the limit and the contour deviation increases. In this operating state there can be jerky movements when the torque limit is increased. To ensure that the axis can follow the setpoint, check the contour deviation to make sure it is not greater than the deviation with an unlimited torque.

4.22 Chamfer, rounding (CHF, CHR, RND, RNDM, FRC, FRCM)

Function

You can insert the following elements at a contour corner:

- Chamfer or
- Rounding

If you wish to round several contour corners sequentially by the same method, use command RNDM "Modal rounding". This address is for inserting a rounding between linear and circle contours after each traversing block. This is, for example, for deburring sharp workpiece edges.

You can program the feedrate for the chamfer/rounding with FRC (non-modal) or FRCM (modal).

If FRC/FRCM is not programmed, then the normal path feedrate F is applicable.

Programming

```
CHF=…
```

Or

```
CHR=…
```

Or

```
RND=…
```

Or

```
RNDM=…
```

Or

```
FRC=…
```

Or

```
FRCM=…
```

Parameters

CHF=…	Chamfer the contour corner Value = Length of the chamfer (unit of measurement according to G70/G71)
CHR=…	Chamfer the contour corner Programming the chamfer in the original direction of movement. Value = width of chamfer in direction of motion (unit of measurement as above)
RND=…	Round the contour corner Value = Radius of the rounding (unit of measurement according to G70/G71)
RNDM=…	Modal rounding: Rounding several consecutive contour corners in the same way. Value = Radius of the roundings (unit of measurement according to G70/G71) Rounding is deactivated with RNDM=0.
FRC=…	Non-modal feedrate for chamfer/rounding Value = feedrate in mm/min (G94) or mm/rev (G95); FRC > 0
FRCM=…	Modal feedrate for chamfer/rounding Value = feedrate in mm/min (G94) or mm/rev (G95) =0: The feedrate programmed under F for the chamfer/rounding is active.

Feed FRC (non-modal), FRCM (modal)

To optimize surface quality, it is possible to program a separate feedrate for the chamfer/rounding contour elements. FRC is non-modal and FRCM is modal, see examples.

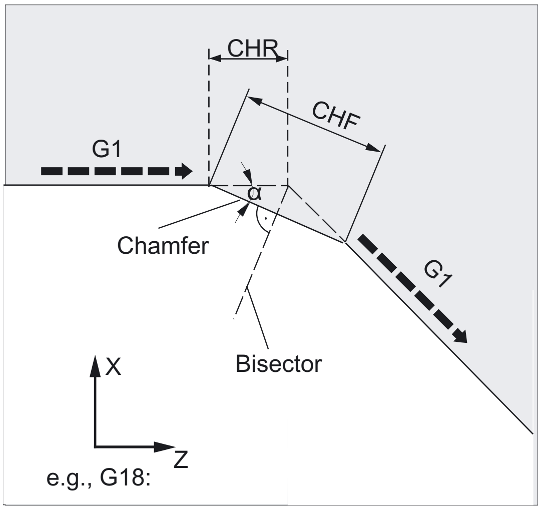
Example of chamfer, CHF/CHR

For the chamfer insert another linear part, the chamfer, between the linear and the circle contours in any combination. The following two options are available:

```
N30 G1 X… Z… F… CHR=2
N40 G1 X… Z…
```

or

```
N30 G1 X… Z… F… CHF=2(cos α ·2)
N40 G1 X… Z…
```

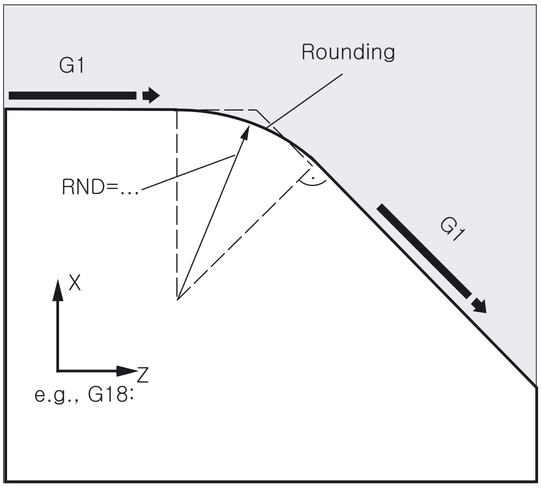


The chamfer is inserted after the block, in which it is programmed. The chamfer is always in the plane activated with G17 to G19.

Example of rounding, RND

A circle contour element can be inserted with tangential link between the linear and the circle contours in any combination.

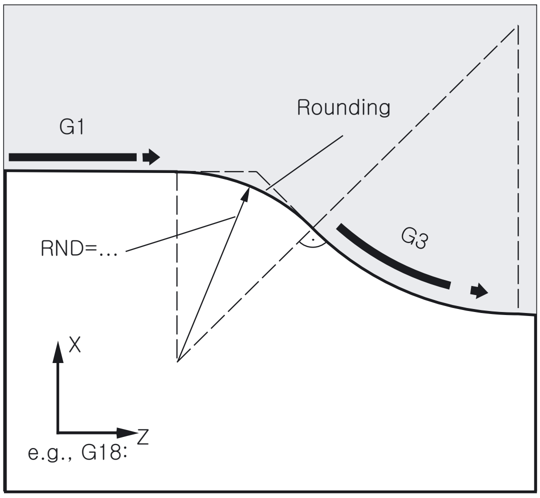
```
N30 G1 X… Z… F… RND=2
```



The rounding is always in the plane activated with G17 to G19. The above figure shows the rounding between two straight lines.

Here, the figure shows the rounding between a straight line and a circle.

```
N30 G1 X… Z… F… RND=2
N40 G3 X… Z… I… K…
```



Example of modal rounding, RNDM

Deburring sharp workpiece edges:

`N30 G1 X… Z… F… RNDM=2` ;modal rounding 2 mm

`N40...`

`N120 RNDM=0` ;deactivate modal rounding

Example of chamfer CHF, rounding FRCM of the following block

MD CHFRND_MODE_MASK Bit0 = 0: Accept technology from following block (default)

```
N10 G0 X0 Y0 G17 F100 G94
N20 G1 X10 CHF=2                      ;Chamfer N20-N30 with F=100 mm/min
N30 Y10 CHF=4                         ;Chamfer N30-N40 with FRC=200 mm/min
N40 X20 CHF=3 FRC=200                 ;Chamfer N40-N60 with FRCM=50 mm/min
N50 RNDM=2 FRCM=50
N60 Y20                               ;Modal rounding N60-N70
                                      ;with FRCM=50 mm/min
N70 X30                               ;Modal rounding N70-N80
                                      ;with FRCM=50 mm/min
N80 Y30 CHF=3 FRC=100                 ;Chamfer N80-N90 with FRC=50 mm/min
                                      ;(modal)
N90 X40                               ;Modal rounding N90-N100
                                      ;with F=100 mm/min (deselect FRCM)
N100 Y40 FRCM=0                       ;Modal rounding N100-N120
                                      ;with G95 FRC=1 mm/rev
N110 S1000 M3
N120 X50 G95 F3 FRC=1
...
M02
```

Example of chamfer CHF, rounding FRCM of the previous block

MD CHFRND_MODE_MASK Bit0 = 1: Accept technology from preceding block (recommended)

```
N10 G0 X0 Y0 G17 F100 G94
N20 G1 X10 CHF=2                          ;Chamfer N20-N30 with F=100 mm/min
N30 Y10 CHF=4 FRC=120                     ;Chamfer N30-N40 with FRC=120 mm/min
N40 X20 CHF=3 FRC=200                     ;Chamfer N40-N60 with FRCM=200 mm/min
N50 RNDM=2 FRCM=50
N60 Y20                                   ;Modal rounding N60-N70
                                          ;with FRCM=50 mm/min
N70 X30                                   ;Modal rounding N70-N80
                                          ;with FRCM=50 mm/min
N80 Y30 CHF=3 FRC=100                     ;Chamfer N80-N90 with FRC=100 mm/min
                                          ;(modal)
N90 X40                                   ;Modal rounding N90-N100
                                          ;with FRCM=50 mm/min
N100 Y40 FRCM=0                           ;Modal rounding N100-N120
                                          ;at F=100 mm/min
N110 S1000 M3
N120 X50 CHF=4 G95 F3 FRC=1               ;Chamfer N120-N130 with G95 FRC=1 mm/rev
N130 Y50                                  ;Modal rounding N130-N140
                                          ;at F=3 mm/rev
N140 X60
...
M02
```

Description

Note

Chamfer/rounding

If the programmed values for chamfer (CHF/CHR) or rounding (RND/RNDM) are too large for the associated contour elements, then the chamfer or rounding are automatically reduced to a suitable value.

No chamfer/rounding is inserted, if

- No straight or circle contour is available in the plane,

- A movement is taking place outside the plane,

- A plane change is taking place, or

The number of blocks - which is specified in the machine data - not containing information for traversing (e.g., only command output), is exceeded.

Note

FRC/FRCM

FRC/FRCM has no effect if a chamfer is being machined with G0; the command can be programmed according to the F value without error message.

The reference to the blocks, in which chamfer and rounding are programmed and to the technology is set in machine data.

FRC is operative only if a chamfer/rounding is programmed in the same block or if RNDM has been activated.

FRC overwrites the F or FRCM value in the current block.

The feedrate programmed under FRC must be greater than zero.

FRCM=0 activates the feed programmed in F for the chamfer/rounding.

If FRCM is programmed, the FRCM value must be reprogrammed, analogous to F, on changeover G94<->95, etc. If only a new F value is programmed, and if FRCM > 0 before the feed type changes, error message 10860 (no feed programmed) will be activated.

Path Action 5

5.1 General notes

5.1.1 Programming path travel behavior

In this section you will find descriptions of commands, with which you can adapt the travel behavior at the block boundaries optimally for special requirements. For instance, you can position axes quickly enough or correspondingly reduce path contours over several blocks taking into account an acceleration limit and the overload factor of the axes. Greater inaccuracies of the path contour occur with increasing speed.

Path commands are programmed with the associated parameters.

Basic description

On change of the movement direction in the continuous-path mode contour transitions are smoothed in that programmed positions are not approached exactly. In this way corners can be moved round with as constant as possible speed or transitions can be optimized with additional commands. With the exact stop function, machining operations can be implemented as precisely as possible with inclusion of additional accuracy criteria. The control determines the velocity control with look ahead automatically for several blocks in advance.

For axes acceleration processes can be activated both conservatively for the mechanical system or time-optimized. Both path axes and positioning axes, geo axes and following axes, which can also change according to program run depending on the relevant blocks or the momentary processing are discussed. The type of feedforward control and which path axes should be operated with feedforward control can also be determined. In machining without feedforward control the maximum permissible contour error can be specified.

A dwell time of a block with implicit pre-processing stop can be generated between two blocks of the NC machining.

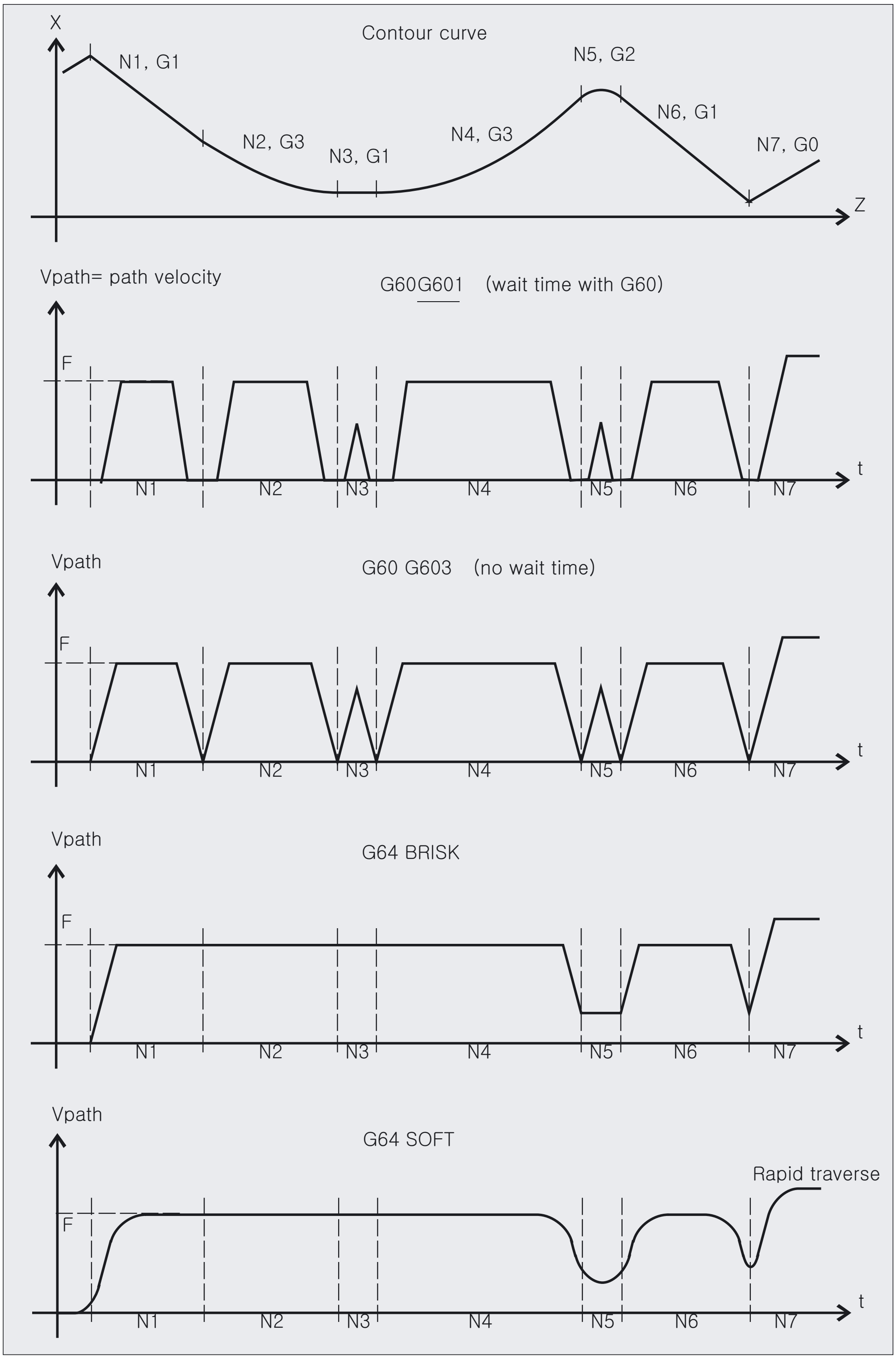
A programming example is given for each typical travel command.

Functions for optimizing travel behavior at block boundaries

The travel behavior at the block boundaries can be optimized with the following functions:

- Setting exact stop to be modally and non-modally effective
- Defining exact stop with additional exact stop windows
- Continuous-path mode with constant speed
- Continuous-path mode with statement of the type of corner rounding
- Continuous-path mode with predictive speed control
- Activating acceleration and speed behavior of axes
- Influencing acceleration of slave axes as a percentage
- Smoothing path velocity
- Travel with feedforward control for increasing path accuracy
- Activate programmable contour accuracy
- Activating programmable dwell time

Overview of the various velocity controls



5.2 Exact stop (G60, G9, G601, G602, G603)

Function

The exact positioning stop functions are used to machine sharp outside corners or to finish inside corners to size.

With the exact stop criteria exact stop window fine and exact stop window coarse, you determine how accurately the corner point is approached and when the change to the next block takes place. At interpolation end you can initiate the block change at the block end when the control has calculated a set speed of zero for the axes involved.

Programming

```
G60
```

Or

```
G9
```

Or

```
G601
```

Or

```
G602
```

Or

```
G603
```

Parameters

G60	Exact stop, modal
G9	Exact stop, non-modal
G601	Step enable if positioning window fine reached
G602	Step enable if positioning window coarse reached
G603	Step enable if setpoint (end of interpolation) reached

Exact stops fine and coarse can be defined for each axis in machine data. The velocity up to reaching the accurate destination position at the end of the block is decelerated to zero.

Note

G601, G602 and G603 are only effective if G60 or G9 are active.

Example

```
N5 G602                                    ;Exact stop window coarse
N10 G0 G60 Z...                            ;Exact stop, modal
N20 X... Z...                              ;G60 continues to act
...
N50 G1 G601                                ;Exact stop window fine
N80 G64 Z...                               ;Switching over to continuous-path mode
...
N100 G0 G9                                 ;Exact stop acts only in this block
N111 ...                                   ;Again continuous-path mode
```

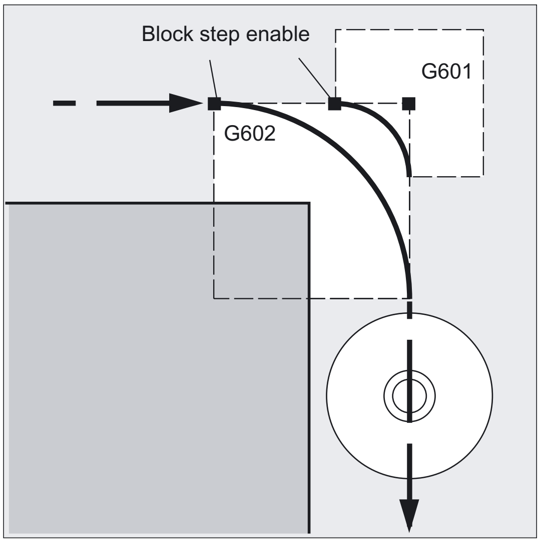
Description

Exact stop, G60, G9

G9 generates the exact stop in the current block. G60 generates the exact stop in the current block and all subsequent blocks.

Continuous-path-mode functions G64 or G641 are used to deactivate G60.

G601/G602



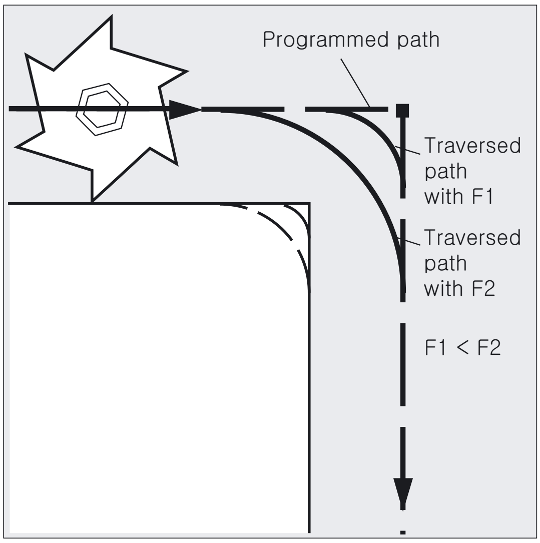
The movement is decelerated and stopped briefly at the corner point.

Note

Set the exact stop limits no tighter than you require. The tighter the limits, the longer it takes to position and approach the target position.

End of interpolation, G603

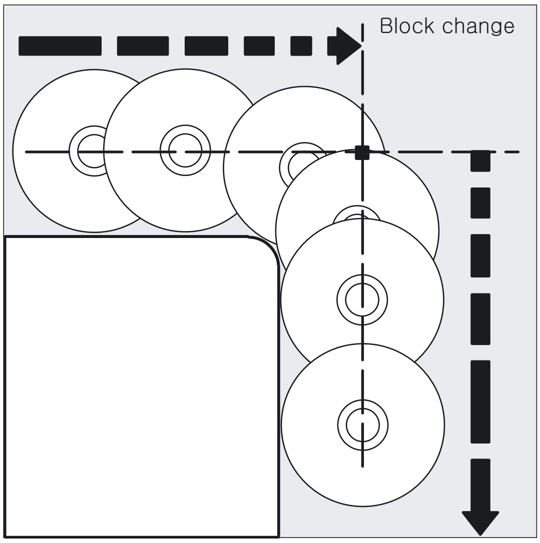
The block change is initiated when the control has calculated a set velocity of zero for the axes involved. At this point, the actual value lags behind by a proportionate factor depending on the dynamic response of the axes and the path velocity. The workpiece corners can now be rounded.



Command outputs

In all three cases, the following applies:

The auxiliary functions programmed in the NC block are enabled after the end of the movement.



Note

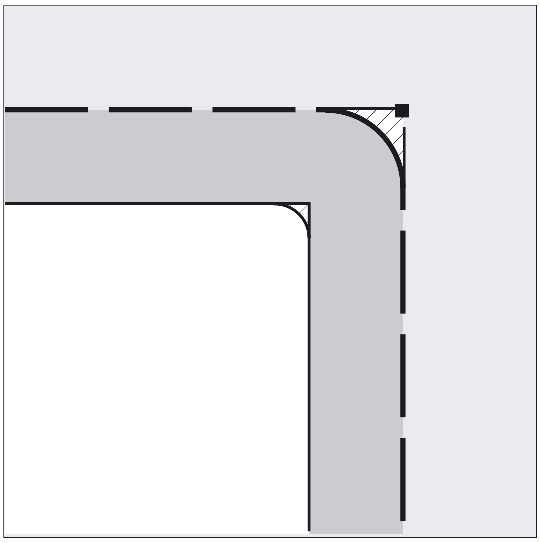
Machine manufacturer

A machine data item can be set for specific channels which determines that default exact stop criteria, which deviate from the programmed criteria, will be applied automatically. These are given priority over the programmed criteria in some cases. Criteria for G0 and the other G commands in the 1st G code group can be stored separately, see functional description FB1, B1.

5.3 Continuous-path mode (G64, G641, G642, G643, G644)

Function

In continuous-path mode, the contour is machined with a constant path velocity. The uniform velocity also establishes better cutting conditions, improves the surface quality and reduces the machining time.



Notice

In continuous-path mode, the programmed contour transitions are not approached exactly.

If a rounding movement initiated by G641, G642, G643, G644 is interrupted, the corner point of the original contour will be used for subsequent repositioning (REPOS), rather than the interruption point.

Sharp corners can be produced with G60 or G9. Continuous-path mode is interrupted by text outputs with "MSG" and blocks, which trigger preprocessing, stop implicitly (e.g., access to particular status data of machine ($A...)). The corresponding situation applies for the auxiliary function outputs.

Programming

`G64`

Or

`G641 ADIS=…`

Or

`G641 ADISPOS=…`

Or

`G642 ADIS=…`

Or

`G642 ADISPOS=…`

Or

`G643 ADIS=…`

Or

`G643 ADISPOS=…`

Or

`G644`

Note

`G644` is not available with an active kinematic transformation. The system switches internally to `G642`.

Parameters

G64	Continuous-path mode
G641	Continuous-path mode with programmable transition rounding
G642	Transition rounding with axial tolerance, with modal activated
G643	Block-internal corner rounding
G644	Corner rounding with greatest possible dynamic response
ADIS=...	Rounding clearance for path functions G1, G2, G3, etc.
ADISPOS=...	Approximate distance for rapid traverse G0

Note

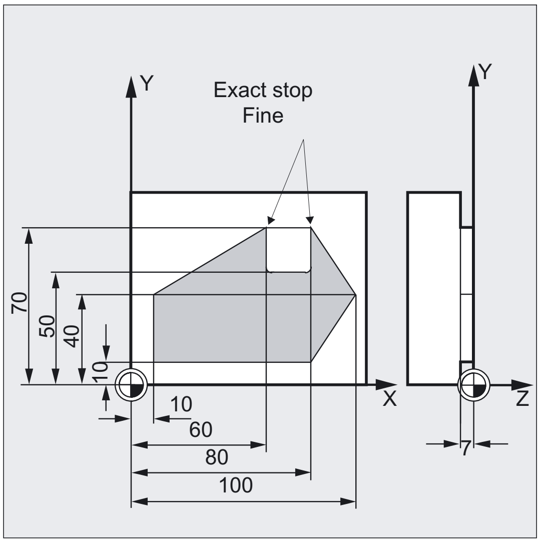
Rounding cannot be used as a substitute for smoothing (RND). The user should not make any assumptions with respect to the appearance of the contour within the rounding area. The type of rounding can depend on dynamic conditions, e.g., on the tool path velocity. Rounding on the contour is therefore only practical with small ADIS values. RND must be used if a defined contour is to be followed at the corner without exception.

`ADISPOS` is used between G0 blocks. This enables the axis movement to be smoothed substantially and the traversing time to be reduced during positioning.

If `ADIS/ADISPOS` is not programmed, a value of zero applies and the traversing behavior therefore corresponds to `G64`. The rounding clearance is automatically reduced (by up to 36%) for short traversing distances.

Example

With this workpiece, the two outside corners at the groove are approached exactly. All other machining takes place in continuous-path mode.



```
N05DIAMOF                                ;Radius as dimension
N10 G17 T1 G41 G0 X10 Y10 Z2 S300 M3     ;Approach starting position,
                                         ;activate spindle, path compensation
N20 G1 Z-7 F8000                         ;Tool infeed
N30 G641 ADIS=0.5                        ;Contour transitions are smoothed
N40 Y40
N50 X60 Y70 G60 G601                     ;Approach exact position with exact
                                         ;stop fine
N60 Y50
N70 X80
N80 Y70
N90 G641 ADIS=0.5 X100 Y40               ;Contour transitions are smoothed
N100 X80 Y 10
N110 X10
N120 G40 G0 X-20                         ;Deactivate path compensation
N130 Z10 M30                             ;Retract tool, end of program
```

Note

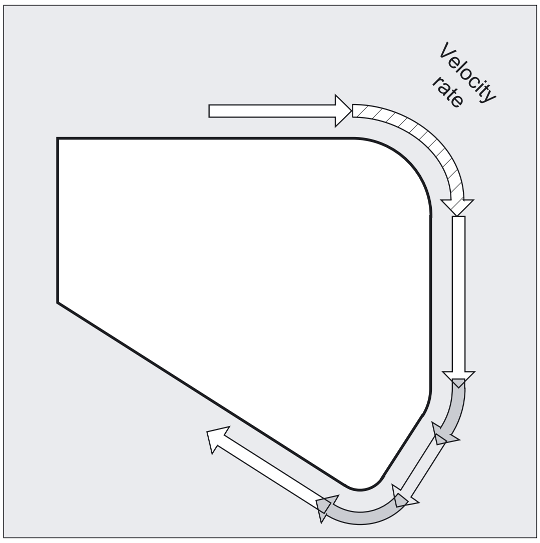
Example for corner rounding with G643, see also: References/PGA/Programming Guide Advanced, Chapter "Settable Path Reference, SPATH, UPATH".

Continuous-path mode, G64

In continuous-path mode, the tool travels across tangential contour transitions with as constant a path velocity as possible (no deceleration at block boundaries). Look Ahead deceleration takes place before corners (G9) and blocks with exact stop (Look Ahead, see following pages).

Corners are also traversed at a constant velocity. In order to minimize the contour error, the velocity is reduced according to an acceleration limit and an overload factor.

References: /FB1/Description of Functions, B1, Continuous-Path Mode.



Note

The overload factor can be set in the machine data 32310. The extent of smoothing of the contour transitions depends on the feedrate and the overload factor. With `G641`, you can specify the desired rounding area explicitly.

Rounding cannot and should not replace the functions for defined smoothing: RND, RNDM, ASPLINE, BSPLINE, CSPLINE.

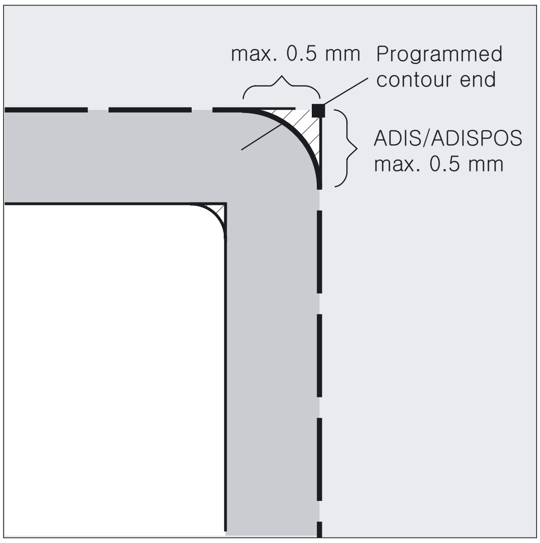
Continuous-path mode with programmable transition rounding, G641

With `G641`, the control inserts transition elements at contour transitions. With `ADIS=`… or `ADISPOS=`…, you can specify the extent, to which the corners are rounded. The effect of `G641` is similar to RNDM; however, it is not restricted to the axes of the working plane.

Example: `N10 G641 ADIS=0.5 G1 X... Y...`

The rounding block must begin 0.5 mm before the programmed end of block at the earliest and must be finished 0.5 mm after the end of the block. This setting remains modal.

`G641` also operates with Look Ahead predictive velocity control. Corner rounding blocks with a high degree of curvature are approached at reduced velocity.



Continuous-path mode G64/G641 over several blocks

The following points should be noted in order to prevent an undesired stop in the path motion (relief cutting):

Auxiliary function outputs trigger a stop (exception: high-speed auxiliary functions and auxiliary functions during movements).

Intermediate blocks, which contain **only** comments, calculation blocks or subprogram calls do not affect the movement.

Extensions of corner rounding

If FGROUP does not contain all the path axes, there is often a step change in the velocity at block boundaries for those axes excluded from FGROUP; the control limits this change in velocity to the permissible values set in MD 32300: MAX_AX_ACCEL and MD 32310: MAX_ACCEL_OVL_FACTOR by reducing the velocity at block changes. This braking operation can be avoided through the application of a rounding function, which "smoothes" the specific positional interrelationship between the path axes.

Corner rounding with G641

You can activate a modal corner rounding action by programming `G641` and specifying a rounding radius with `ADIS` (or `ADISPOS` in rapid traverse). Within this radius about the block change point, the control is free to ignore the path construct and replace it with a dynamically optimized path. **Disadvantage:** Only **one** ADIS value is available for all axes.

Corner rounding with axial precision using G642

`G642` activates corner rounding with modal axial tolerances. Corner rounding is not carried out inside a defined ADIS area but the **axial** tolerances defined with MD 33100: COMPRESS_POS_TOL are maintained. Otherwise, the mode of operation is identical to `G641`.

With `G642`, the rounding travel is determined on the basis of the **shortest** rounding travel of all axes. This value is taken into account when generating a **rounding block**.

Block-internal corner rounding with G643

The **maximum** deviations from the **exact contour** are specified for each axis by machine data MD 33100: COMPRESS_POS_TOL[...] when a contour is smoothed with G643. `G643` is not used to generate a separate rounding block, but axisspecific blockinternal rounding movements are inserted. In the case of `G643`, the rounding travel of each axis may be **different.**

Corner rounding with contour tolerance in G642 and G643

The expansions described below refine the response with `G642` and `G643` and **rounding with contour tolerance** is introduced. When rounding with `G642` and `G643`, the maximum permissible deviations of each axis are normally specified.

MD 20480: SMOOTHING_MODE can be used to configure corner rounding with `G642` and `G643` such that instead of the axis-specific tolerances, a contour tolerance and an orientation tolerance can be specified. In this case, the tolerance of the **contour** and of the **orientation** is set using two independent setting data, which can be programmed in the NC program, so that setting data can be specified differently for each block transition.

Setting data

SD 42465: SMOOTH_CONTUR_TOL

This setting data is used to define the **maximum** tolerance for the contour when rounding.

SD 42466: SMOOTH_ORI_TOL

This setting data is used to define the **maximum** tolerance for the contour when rounding for the **tool orientation** (angle deviation).

This data is only effective if an **orientation transformation** is active. Very different specifications for the contour tolerance and the tolerance of the tool orientation can only have effect with `G643`.

Corner rounding with greatest possible dynamic response in G644

Corner rounding with greatest possible dynamic response is activated with `G644` and with MD 20480. SMOOTHING_MODE configured in the thousands position. The possible settings are as follows:

0:
Input maximum axial deviations using MD 33100: COMPRESS_POS_TOL

1:
Input the maximum rounding travel by programming `ADIS=...` or `ADISPOS=...`

2:
Input the maximum possible frequencies of each axis in the rounding area using MD 32440: LOOKAH_FREQUENCY. The rounding area is defined such that no frequencies in excess of the specified maximum can occur while the rounding motion is in progress.

3:
When rounding with `G644`, neither the tolerance nor the rounding distance are monitored.
Each axis traverses around a corner with the maximum possible dynamic response.
With SOFT, both the maximum acceleration and the maximum jerk of each axis is maintained.
With the BRISK command, the jerk is not limited; instead, each axis travels at the maximum possible acceleration.

References:/FB1/, B1, "Continuous Path Mode, Exact Stop and Look Ahead"

No rounding block/no rounding movement

Command outputs

Auxiliary functions, which are enabled after the end of the movement or before the next movement interrupt continuous-path mode.

Positioning axes

Positioning axes always traverse according to the exact stop principle, positioning window fine (as for G601). If an NC block has to wait for positioning axes, continuous-path mode is interrupted on the path axes.

No corner rounding is performed in the following three situations:

1. Movement **is stopped** between the two blocks. This occurs when ...
   - The following block contains an auxiliary function output before the movement.
   - The following block does not contain a path movement.
   - An axis is traversed for the first time as a path axis for the following block when it was previously a positioning axis.
   - An axis is traversed for the first time as a positioning axis for the following block when it was previously a path axis.
   - Prior to thread cutting: The following block has G33 as a preparatory function and the previous block does not.
   - A changeover between BRISK and SOFT occurs.
   - Axes involved in the transformation are not completely assigned to the path motion (e.g., for oscillation, positioning axes).
2. The rounding block would **slow down** parts program processing. This occurs when ...
   - A rounding block is inserted between very short blocks. Since each block requires at least one interpolation cycle, the added intermediate block would double the machining time.
   - A block transition `G64` (continuous-path mode without corner rounding) can be traversed without a reduction in velocity. Rounding would increase the machining time. This means that the value of the permitted overload factor (MD 32310: MAX_ACCEL_OVL_FACTOR) affects whether a block transition is rounded or not. The overload factor is only taken into account when corner rounding in conjunction `G641/G642`. The overload factor is ignored in corner rounding with `G643`.
   - MD 20490: IGNORE_OVL_FACTOR_FOR_ADIS can be set to TRUE to set this behavior for `G641` and G642.
3. Rounding is **not parameterized**. This occurs with `G641 when ...`
   - `ADISPOS== 0` in G0 blocks (default).
   - `ADIS== 0` in non-G0 blocks (default).
   - For transition from G0 to non-G0 or non-G0 to G0, the smaller value of `ADISPOS` and `ADIS` applies.

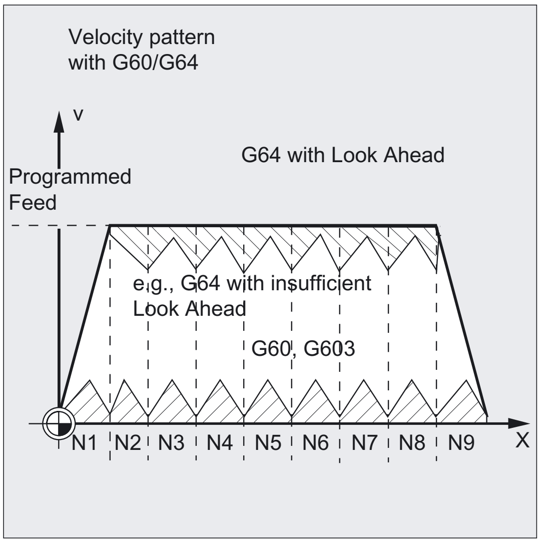
With `G642/G643`, when all axis-specific tolerances equal zero.

Look Ahead speed control

In continuous-path mode with `G64` or `G641`, the control automatically detects the velocity control in advance for several NC blocks. This enables acceleration and deceleration across multiple blocks with almost tangential transitions.

Look Ahead is particularly suitable for the machining of movement sequences comprising short traverse paths with high path feedrates.

The number of NC blocks included in the Look Ahead calculation can be defined in machine data.



Note

Look Ahead across more than one block is an option.

Continuous-path mode in rapid traverse G0

One of the functions G60/G9 or `G64/G641` must also be specified for rapid traverse. Otherwise, the default in the machine data is used.

By setting MD 20490: IGNORE_OVL_FACTOR_FOR_ADIS results in block transitions being smoothed irrespective of the programmed overload factor.

5.4 Acceleration behavior

5.4.1 Acceleration response, BRISK, SOFT, DRIVE

Function

BRISK, BRISKA: The axis slides travel with maximum acceleration until the feedrate is reached. BRISK enables time-optimized machining, but with jumps in the acceleration curve.

SOFT, SOFTA: The axis slides travel with constant acceleration until the feedrate is reached. SOFT acceleration enables higher path accuracy and less wear and tear on the machine.

DRIVE, DRIVEA: The axis slides traverse at the maximum acceleration rate up to the velocity limit set in the machine data. The acceleration rate is then reduced according to machine data until the feedrate is reached. This function allows the acceleration characteristic to be optimally adapted to a specific motor characteristic, for example, for stepper motor applications.

Programming

```
BRISK
BRISKA (axis1,axis2,…)
```

or

```
SOFT
SOFTA(axis,axis2,…)
```

or

```
DRIVE
DRIVEA(axis1,axis2,…)
```

Parameter

BRISK	Abrupt acceleration of path axes
BRISKA (axis1,axis2,…)	Switch on stepped axis acceleration for the programmed axes
SOFT	Jerk-limiting acceleration of path axes
SOFTA(axis,axis2,…)	Switch on jerk-limiting axis acceleration for the programmed axes
DRIVE	Reduction of acceleration above a velocity for path axes that can be set in $MA_ACCEL_REDUCTION_SPEED_POINT (only applicable for FM-NC)
DRIVEA(axis1,axis2,…)	Reduction of acceleration above a velocity for programmed axes that can be set in $MA_ACCEL_REDUCTION_SPEED_POINT (only applicable for FM-NC)
(axis1,axis2,…)	The acceleration pattern set in machine data $MA_POS_AND_JOG_JERK_ENABLE or $MA_ACCEL_TYPE_DRIVE is active for the programmed axes.

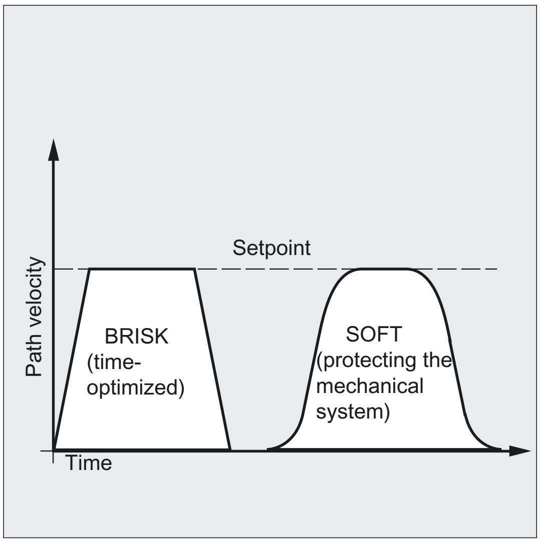
Note

A change between BRISK and SOFT causes a stop at the block transition. The acceleration pattern for the path axes can be defined in machine data.

Apart from the path-related jerk limitation that is effective in the MDA and AUTO modes, there is the axis-related jerk limitation that can influence positioning axes and traversing axes in JOG mode.

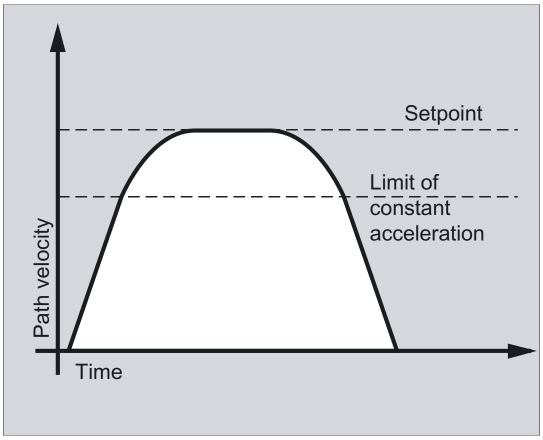
Example of BRISK and SOFT

```
N10 G1 X… Y… F900 SOFT
N20 BRISKA(AX5,AX6)
```



Example of DRIVE, DRIVEA

```
N05 DRIVE
N10 G1 X… Y… F1000
N20 DRIVEA (AX4, AX6)
```



5.4.2 Influence of acceleration on following axes (VELOLIMA, ACCLIMA, JERKLIMA)

Function

The axis couplings described in the Programming Guide, Advanced: Tangential correction, coupled-motion axes, master value coupling, and electronic gearbox have the property of moving following axes/spindles as a function of one or more leading axes/spindles.

The commands for correction of limitation for the dynamic response of the following axis can be issued from the parts program or from synchronous actions. The commands for correction of limitations of the following axis can already be given while axis coupling is active.

Programming

VELOLIMA[AX4]=75	75% of the maximum axial velocity stored in the machine data
ACCLIMA[AX4]=50	50% of the maximum axial acceleration stored in the machine data
JERKLIMA[AX4]=50	50% of the maximum jerk on path motion stored in the machine data

Parameters

VELOLIMA[Ax],	Change to limit for maximum velocity for following axis
ACCLIMA[Ax],	Change to limit for maximum acceleration for following axis
JERKLIMA[Ax],	Change to limit for maximum jerk for following axis

Note

JERLIMA[ax] is not available for all types of connection. Details about the function are described in:
References: Functional description/FB3/, M3, Coupled axes and ESR,/FB2/, S3, synchronized spindles.

Example of electronic gear

Axis 4 is coupled to axis X via an electronic gear coupling. The acceleration capability of the following axis is limited to 70% of maximum acceleration. The maximum permissible velocity is limited to 50% of maximum velocity. After successful activation of the coupling, the maximum permissible velocity is set to 100% again.

```
N120 ACCLIMA[AX4]=70                          ;Reduced maximum acceleration
N130 VELOLIMA[AX4]=50                         ;Reduced maximum velocity
...
N150 EGON(AX4, "FINE", X, 1, 2)               ;Activation of the EG coupling
...
N200 VELOLIMA[AX4]=100                        ;Full maximum velocity
```

Example of influencing master value coupling by static synchronized action

Axis 4 is coupled to X by master value coupling. The acceleration response is limited to 80% by static synchronized action 2 from position 100.

```
N120 IDS=2 WHENEVER $AA_IM[AX4] > 100         ;Synchronized action
 DO ACCLIMA[AX4]=80
N130 LEADON(AX4, X, 2)                        ;Master value coupling on
```

5.4.3 Technology G group (DYNNORM, DYNPOS, DYNROUGH, DYNSEMIFIN, DYNFINISH)

Function

Using the "Technology" G group, the appropriate dynamic response can be activated for five varying technological machining steps.

Machine manufacturer

Dynamic values and G codes can be configured and are, therefore, dependent on machine data settings.

References: /FB3/, B1, "Continuous-path mode"

Programming

`DYNNORM`

Or

`DYNPOS`

Or

`DYNROUGH`

Or

`DYNSEMIFIN`

Or

`DYNFINISH`

Parameters

DYNNORM	Standard dynamic response, as previously (index n=0)
DYNPOS	Dynamic response for positioning mode, tapping (index n=1)
DYNROUGH	Dynamic response for roughing (index n=2)
DYNSEMIFIN	Dynamic response for finishing (index n=3)
DYNFINISH	Dynamic response for smooth-finishing (index n=4)

Write or read specific field element

$MA...[n, X]	Machine data with field element, which affects dynamic response
[<n>, <X>]	Field element with field device n and axis address X
n = 0 to 4	Range of values corresponds to Technology G group.

Note

The dynamic values are already active in the block, in which the associated G code is programmed. Machining is not stopped.

Example

Dynamic values by technology group G code

```
DYNNORM G1 X10                    ;Initial setting
DYNPOS G1 X10 Y20 Z30 F…          ;Positioning mode, tapping
DYNROUGH G1 X10 Y20 Z30 F10000    ;Roughing
DYNSEMIFIN G1 X10 Y20 Z30 F2000   ;Finishing
DYNFINISH G1 X10 Y20 Z30 F1000    ;Smooth-finishing
```

Write or read specific field element

Maximum acceleration for roughing, axis X

```
R1=$MA_MAX_AX_ACCEL[2, X]         ;Read
$MA_MAX_AX_ACCEL[2, X]=5          ;Write
```

5.5 Smoothing the path velocity

Function

A smoother path velocity profile can be achieved with the "Path velocity smoothing" function, which allows special, configurable, machine data and the character of the parts program to be taken into account.

The velocity control function utilizes the specified axial dynamic response. If the programmed feedrate cannot be achieved, the path velocity is brought to the parameterized axial limit values and the limit values of the path (velocity, acceleration, jerk). This can lead to repeated braking and acceleration on the path.

Note

Machine manufacturer

The user can control the path velocity as appropriate for the configurable machine data by making use of the program properties.

References: Functional description /FB1/; B1; "Smoothing the path velocity"

Parameter

Machine manufacturer
Limit values that can be configured in relation to (specially) adjustable parameters of the parts program, using machine data:

- Lengthening the machining time
  The machining time of the part program is specified as percentage. The actual lengthening is according to the worst case of all acceleration processes inside the part program and can even be zero.
- Input of the natural frequencies of the operated axes
  Only acceleration processes that lead to clear excitation of machine axes should be removed.
- Taking the programmed feedrate into account.
  In this case the smoothing factor is observed especially exactly if the override is set to 100%.

Note

Variations in path velocity due to the input of a new feedrate are not changed either. This remains the responsibility of the programmer of the subprogram.

Note

If a short acceleration takes place during a machining function with high path velocity, and is thus followed almost immediately by braking, the reduction in the machining time is only minimal. Acceleration of this kind can, however, have undesirable effects if, for example, it results in machine resonance.

5.6 Traversing with feedforward control (FFWON, FFWOF)

Function

Using feedforward control the velocity-dependent overtravel in path traversing is reduced to zero. Traversing with feedforward control permits higher path accuracy and thus improved machining results.

Programming

```
FFWON
```

Or

```
FFWOF
```

Parameters

FFWON	Activate feedforward control
FFWOF	Deactivate feedforward control

Note

The type of feedforward control and the path axes to which feedforward is to be applied are determined via machine data.

Default: Velocity-dependent feedforward control

Option: Acceleration-dependent feedforward control (not possible with 810D)

Example

```
N10 FFWON
N20 G1 X… Y… F900 SOFT
```

5.7 Contour accuracy (CPRECON, CPRECOF)

Function

In machining operations without feedforward control (FFWON), errors may occur on curved contours as a result of velocity-related differences between setpoint and actual positions.

The programmable contour accuracy function CPRECON makes it possible to store a maximum permissible contour violation in the NC program which must never be overshot. The magnitude of the contour violation is specified with setting data $SC_CONTPREC.

The Look Ahead function allows the entire path to be traversed with the programmed contour accuracy.

Programming

```
CPRECON
```

or

```
CPRECOF
```

Parameter

CPRECON	Activate programmable contour accuracy
CPRECOF	Deactivate programmable contour accuracy

Note

A minimum velocity can be defined via the setting datum $SC_MINFEED, which is not undershot, and the same value can also be written directly out from the parts program via the system variable $SC_CONTPREC.

On the basis of the value of the contour violation $SC_CONTPREC and the servo gain factor (velocity/following error ratio) of the geometry axes concerned, the control calculates the maximum path velocity at which the contour violation produced by the overtravel does not exceed the minimum value stored in the setting data.

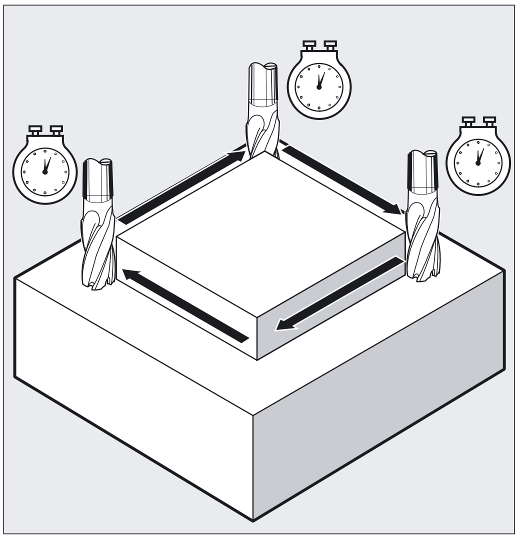
Example

```
N10 X0 Y0 G0
N20 CPRECON                  ;Activate contour accuracy
N30 F10000 G1 G64 X100       ;Machine contour at 10 m/min in continuous-path
                             ;mode
N40 G3 Y20 J10               ;Automatic feed limitation in circular block
N50 X0                       ;Feed without limitation to 10 m/min
```

5.8 Dwell time (G4)

Function

You can use G4 to interrupt workpiece machining between two NC blocks for the programmed length of time, e.g., for relief cutting.



Programming

```
G4 F…
```

Or

```
G4 S…
```

Programming in a separate NC block

Parameters

`G4`	`Activate dwell time, G4 interrupts the continuous-path mode`
`F…`	`Time specified in seconds`
`S…`	`Time specified in revolutions of the master spindle`

Note

The words with F... and S... are used for time specifications only in the block with G4.

Any previously programmed feed F and spindle speed S remain valid.

Example

```
N10 G1 F200 Z-5 S300 M3          ;Feed F; spindle speed S
N20 G4 F3                        ;Dwell time 3 s
N30 X40 Y10
N40 G4 S30                       ;Dwelling 30 revolutions of the spindle,
                                 ;corresponds
                                 ;at S=300 rpm and 100% speed override to:
                                 ;t=0.1 min
N40 X...                         ;Feed and spindle speed remain effective
```

5.9 Internal preprocessor stop

Function

The control generates an internal preprocessor stop on access to machine status data ($A...). If a command, which implicitly causes a preprocessing stop is read in a following block, this block is not executed until all other blocks which are already preprocessed and stored have been executed. The preceding block is halted in exact stop (as with G9).

Programming

Machine status data ($A...) are generated internally by the control.

Parameters

Status data of the machine ($A...)

Example

Machining should be stopped in block N50.

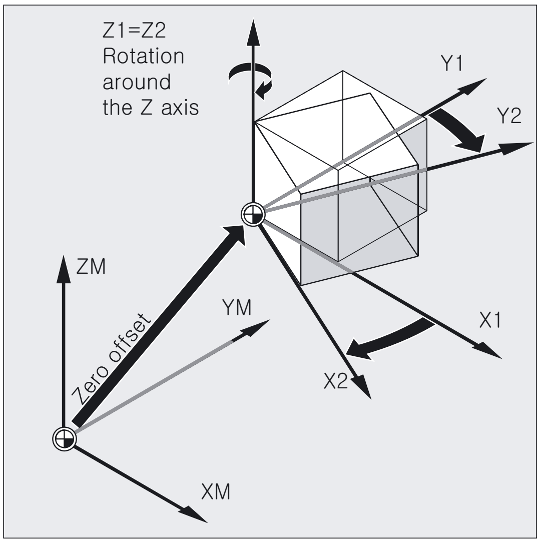
```
N40 POSA[X]=100
N50 IF $AA_IM[X]==R100 GOTOF     ;Access to machine status data ($A...), the
 MARKE1                          ;control generates an internal preprocessor stop
N60 G0 Y100
N70 WAITP(X)
N80 LABEL1:
N40 X...                         ;Feed and spindle speed remain effective
```

Frames 6

6.1 General

Function

Frames are used to describe the position of a destination coordinate system by specifying coordinates or angles starting from the current workpiece coordinate system.



Possible frames:

- Basic frame (basic offset)
- Settable frames (G54...G599)
- Programmable frames

Programming

Frame is the conventional term for a geometrical expression that describes an arithmetic rule, such as translation, rotation and scaling or mirroring.

Parameters

Machine manufacturer

Settable frames (G54...G57, G505... G599): See machine manufacturer's specifications.

Frame components for the programmer

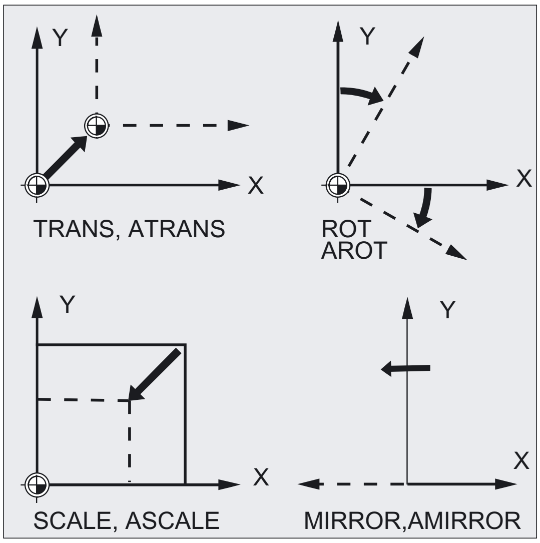
A frame can consist of the following arithmetic rules:

Zero point offset, TRANS, ATRANS

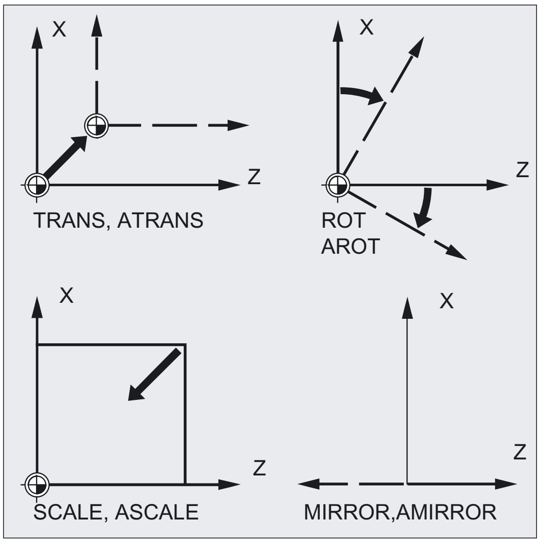
- Rotation, ROT, AROT
- Scale, SCALE, ASCALE
- Mirroring, MIRROR, AMIRROR

These frames can be used individually or in any combination.

Example of frame components in milling



Example of frame components in turning



6.2 Frame instructions

Function

For the possible frames the position of one of the target coordinate systems is defined:

- Basic frame (basic offset)
- Settable frames (G54...G599)
- Programmable frames

In addition to these frames, you can program replacing and additive statements or generate frames as well as frame rotations for tool orientation. Certain set frames or superposed movements and transformations can also be deselected.

Basic frame (basic offset)

The basic frame describes the coordinate transformation from the Basic Coordinate System (BCS) to the Basic Zero System (BZS) and has the same effect as for settable frames.

Settable instructions

Settable instructions are the zero offsets, which can be called from any NC program with the commands G54 to G599. The offset values are predefined by the user and stored in the zero offset memory on the control. This is used to define the Workpiece Coordinate System (WCS).

Programmable instructions

Programmable instructions (TRANS, ROT, etc.) are valid in the current NC program and refer to the settable instructions. The programmable frame is used to define the Workpiece Coordinate System (WCS).

Programming

`TRANS X… Y… Z…` or `ATRANS X… Y… Z…` or

`G58 X… Y… Z… A…` or `G59 X… Y… Z… A…` or

`ROT X… Y… Z…` or `ROT RPL=…` or `AROTX… Y… Z…` or `AROT RPL=…` or

`ROTS X... Y...` or `AROTS X... Y...` or `CROTS X... Y...` or

`SCALE X… Y… Z…` or `ASCALE X… Y… Z…` or

`MIRROR X0 Y0 Z0` or `AMIRROR X0 Y0 Z0` or

`TOFRAME` or `TOFRAMEZ` or `TOFRAMEY` or `TOFRAMEX` or

`TOROTOF` or `TOROT` or `TOROTZ` or `TOROTY` or `TOROTX` or

`PAROT` or `PAROTOF` or

`CORROF(axis,string[axis,string])` or `CORROF(axis,string)` or
`CORROF(axis)` or `CORROF()`

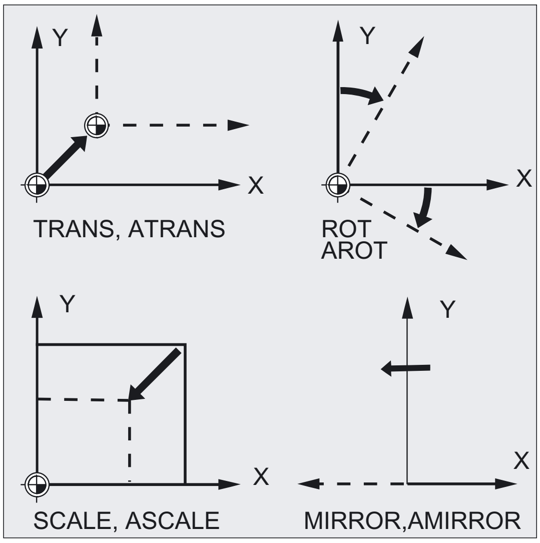
Caution

The above frame instructions are programmed in separate NC blocks and executed in the programmed order.

TRANS, ROT, SCALE and MIRROR instructions

Substituting instructions

TRANS, ROT, SCALE and MIRROR are substituting instructions.



Note

This means that each of these instructions cancels **all** other previously programmed frame instructions.

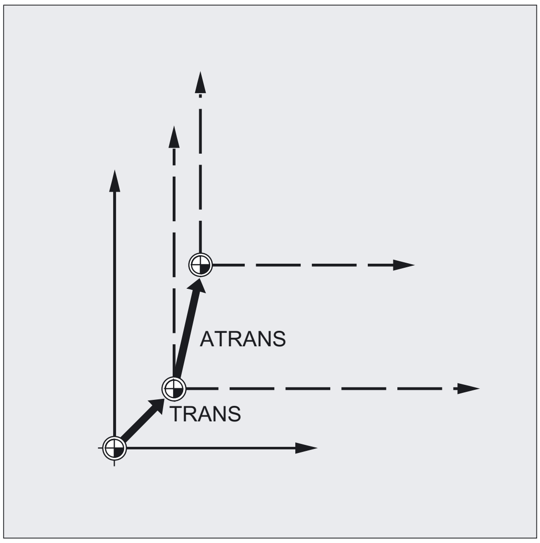
The last called settable zero offset G54 to G599 is used as the reference.

Additive instructions

ATRANS, AROT, ASCALE and AMIRROR are additive instructions. The currently set zero point or the last workpiece zero to be programmed with frame instructions is used as the reference. The above instructions are added to existing frames.

Note

Additive instructions are frequently used in subprograms. The basic functions defined in the main program are not lost after the end of the subprogram if the subprogram has been programmed with the SAVE attribute.



References: /PGA/ Programming Guide Advanced, Section "Subroutines, Macros"

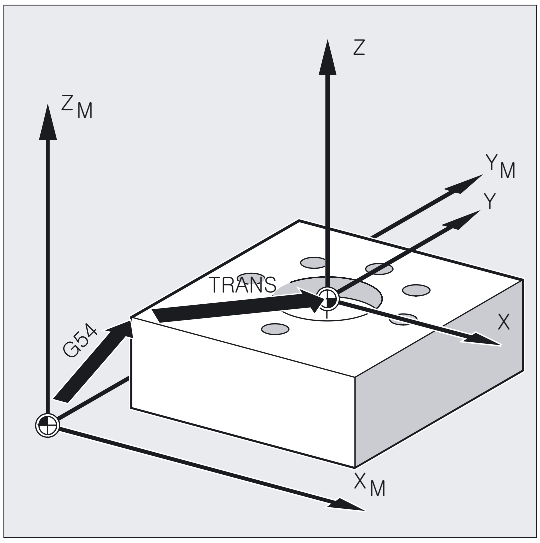
6.3 Programmable zero offset

6.3.1 Zero offset (TRANS, ATRANS)

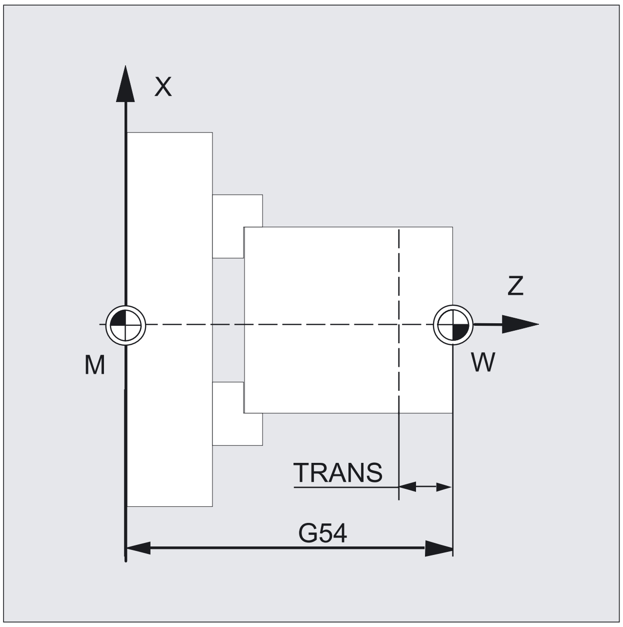
Function

TRANS/ATRANS can be used to program translations for all path and positioning axes in the direction of the specified axis. This allows you to work with different zero points, for example when performing recurring machining processes at different workpiece positions.

Milling:



Turning:



Deactivate programmable zero offset:
For all axes: TRANS (without axis parameter)

Programming

`TRANS X… Y… Z…` (substituting instruction programmed in a separate NC block)

Or

`ATRANS X… Y… Z…` (additive instruction programmed in a separate NC block)

Parameters

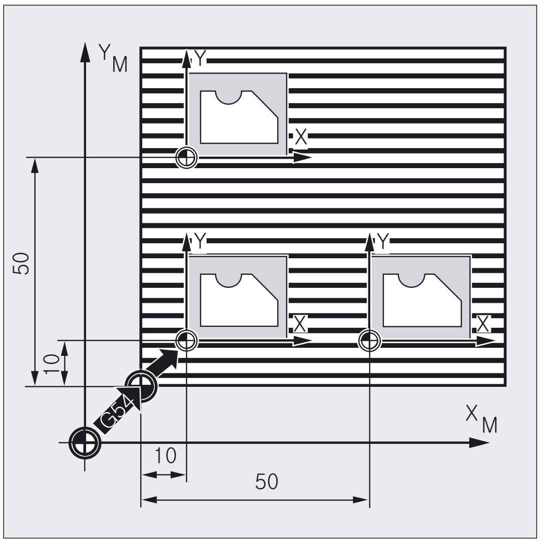
TRANS	Absolute zero offset, with reference to the currently valid workpiece zero set with G54 to G599
ATRANS	as TRANS, but with additive zero offset
X Y Z	Offset value in the direction of the specified geometry axis

Example of milling

With this workpiece, the illustrated shapes recur several times in the same program.

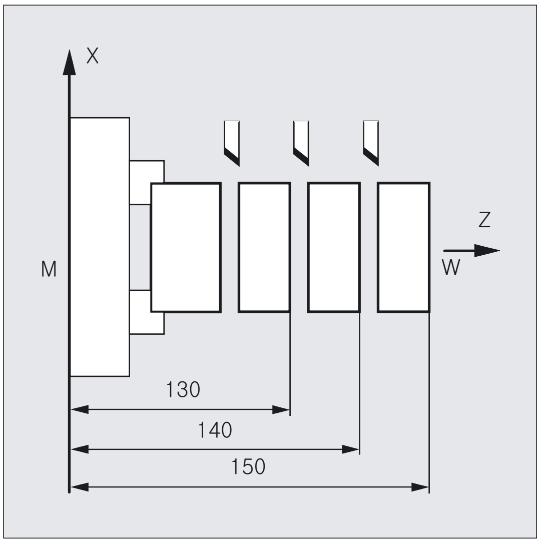
The machining sequence for this shape is stored in a subprogram.

You use the translation to set only those workpiece zeroes and then call up the subprogram.



```
N10 G1 G54                    ;Working plane X/Y, workpiece zero
N20 G0 X0 Y0 Z2               ;Approach starting point
N30 TRANS X10 Y10             ;Absolute offset
N40 L10                       ;Subprogram call
N50 TRANS X50 Y10             ;Absolute offset
N60 L10                       ;Subprogram call
N70 M30                       ;End of program
```

Example of turning



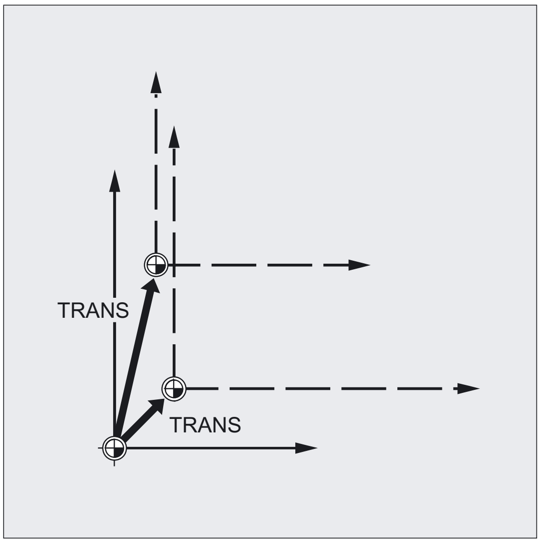
```
N.. ...
N10 TRANS X0 Z150                               ;Absolute offset
N15 L20                                         ;Subprogram call
N20 TRANS X0 Z140 (or ATRANS Z-10)              ;Absolute offset
N25 L20                                         ;Subprogram call
N30 TRANS X0 Z130 (or ATRANS Z-10)              ;Absolute offset
N35 L20                                         ;Subprogram call
N.. ...
```

Substituting instruction, TRANS X Y Z

Translation through the offset values programmed in the specified axis directions (path, synchronized axes and positioning axes). The last specified settable zero offset (G54 to G599) is used as a reference.

Note

The TRANS command cancels all frame components of the previously activated programmable frame.

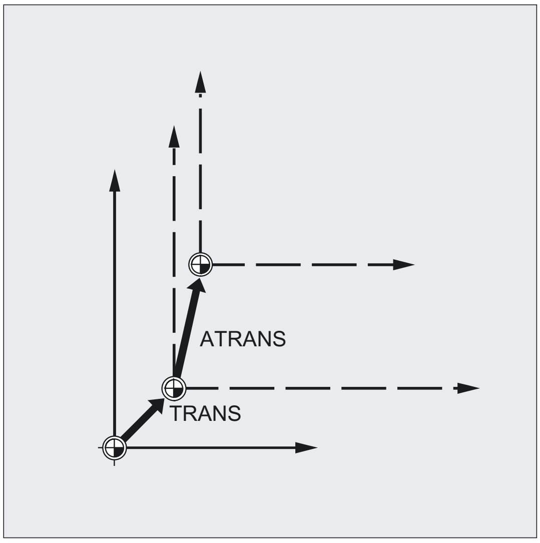


Note

You can use ATRANS to program a translation, which is to be added to existing frames.

Additive instruction, ATRANS X Y Z

Translation through the offset values programmed in the specified axis directions. The currently set or last programmed zero point is used as the reference.



Note

Previously programmed frames are canceled. The settable zero offset remains programmed.

6.3.2 Axial zero offset (G58, G59)

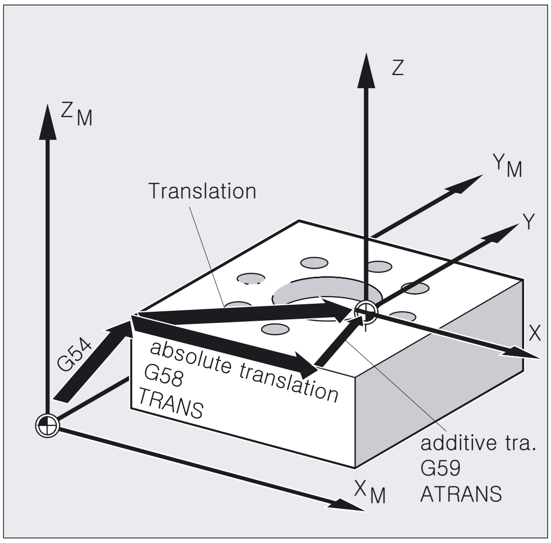
Function

G58 and G59 allow translation components of the programmable zero offset (frame) to be replaced for specific axes. The translation function comprises:

- absolute component (G58, coarse offset)
- additive component (G59, fine offset)

Machine manufacturer

These functions can only be used if the fine offset is configured via machine datum MD 24000: FRAME_ADD_COMPONENTS=1. If G58 or G59 is used without a configured fine offset, alarm "18312 channel %1 block %2 frame: Fine offset not configured" is output.



Programming

`G58 X… Y… Z… A…` (substituting instruction programmed in separate NC block)

Or

`G59 X… Y… Z… A…` (substituting instruction programmed in separate NC block)

Parameters

G58	Replaces the absolute translation component of the programmable zero offset for the specified axis, but the programmed additive offset remains valid, (in relation to the workpiece zero set with G54 to G599)
G59	Replaces the absolute translation component of the programmable zero offset for the specified axis, but the programmed absolute offset remains valid
X Y Z	Offset value in the direction of the specified geometry axis

Example

```
N...
N50 TRANS X10 Y10 Z10    ;Absolute translation component X10 Y10 Z10
N60 ATRANS X5 Y5         ;Additive translation component X5 Y5
                          = total offset X15 Y15 Z10
N70 G58 X20              ;Absolute translation component X20 + addit. X5 Y5
                          = total offset X25 Y15 Z10
N80 G59 X10 Y10          ;Additive translation component X10 Y10 + absolute X20 Y 10
                          = total offset X30 Y20 Z10
N...
```

Description

The absolute translation component is modified by the following commands:

- TRANS
- G58
- CTRANS
- CFINE
- $P_PFRAME[X,TR]

The additive translation component is modified by the following commands:

- ATRANS
- G59
- CTRANS
- CFINE
- $P_PFRAME[X,FI]

The table below describes the effect of various program commands on the absolute and additive offsets.

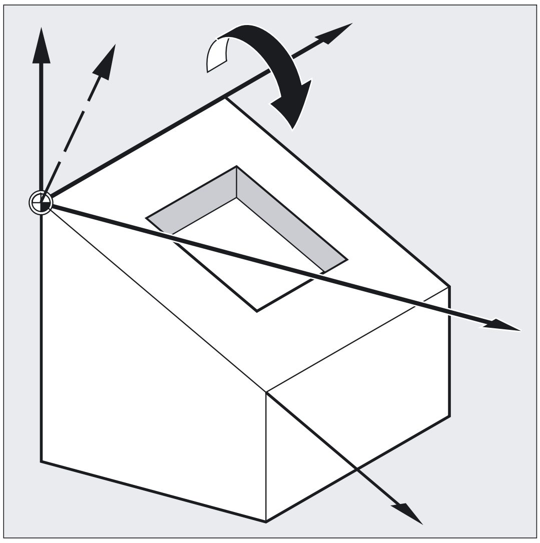
Effect of the additive/absolute offset:

command	Coarse or absolute offset	Fine or additive offset	Comment
TRANS X10	10	unchanged	Absolute offset for X
G58 X10	10	unchanged	Overwrites absolute offset for X
$P_PFRAME[X,TR] = 10	10	unchanged	Progr. offset in X
ATRANS X10	unchanged	Fine (old) + 10	Additive offset for X
G59 X10	unchanged	10	Overwriting additive offset for X
$P_PFRAME[X,FI] = 10	unchanged	10	Progr. fine offset in X
CTRANS(X,10)	10	0	Offset for X
CTRANS()	0	0	Deselection of offset (including fine offset component)
CFINE(X,10)	0	10	fine offset in X

6.4 Programmable rotation (ROT, AROT, RPL)

Function

ROT/AROT can be used to rotate the workpiece coordinate system around each of the geometry axes X, Y, Z or through an angle RPL in the selected working plane G17 to G19 (or around the perpendicular infeed axis). This allows inclined surfaces or several workpiece sides to be machined in one setting.



Programming

`ROT X… Y… Z…` Substituting instruction for rotation in space

Or

`ROT RPL=…` Substituting instruction for rotation in the plane

Or

`AROTX… Y… Z…` Additive instruction for rotation in space

Or

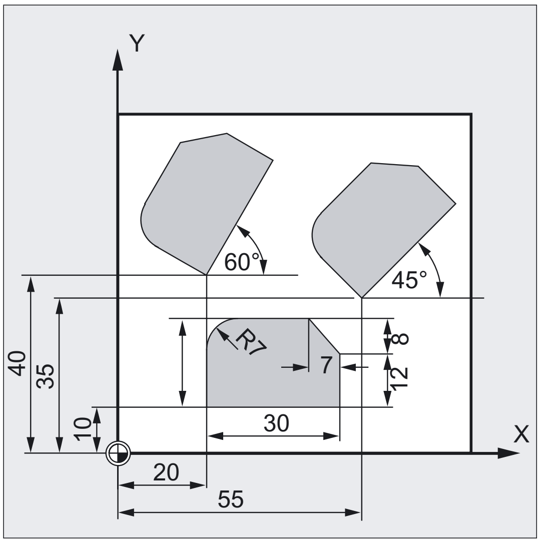
`AROT RPL=…` Additive instruction for rotation in the plane

Each instruction must be programmed in a separate NC block.

Parameters

ROT,	Absolute rotation with reference to the currently valid workpiece zero set with G54 to G599
RPL,	Rotation in the plane: Angle, through which the coordinate system is rotated (plane set with G17-G19). The order, in which the rotation should be performed can be defined via machine data. RPY notation is the default setting (= roll, pitch, yaw) with Z,Y,X
AROT,	Additive rotation with reference to the currently valid set or programmed zero point
X Y Z	Rotation in space: geometry axes, around which the rotation takes place

Example: Rotation in the plane

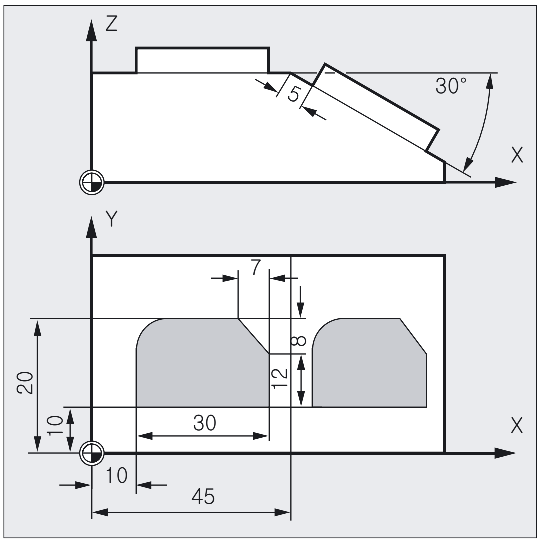


With this workpiece, the illustrated shapes recur several times in the same program. Rotations have to be performed in addition to the translation, because the shapes are not arranged parallel to the axes.

```
N10 G17 G54                  ;Working plane X/Y, workpiece zero
N20 TRANS X20 Y10            ;Absolute offset
N30 L10                      ;Subprogram call
N40 TRANS X55 Y35            ;Absolute offset
N50 AROT RPL=45              ;Rotation of the coordinate system through 45°
N60 L10                      ;Subprogram call
N70 TRANS X20 Y40            ;Absolute offset
                             ;(cancels all previous offsets)
N80 AROT RPL=60              ;Additive rotation through 60°
N90 L10                      ;Subprogram call
N100 G0 X100 Y100            ;Retraction
N110 M30                     ;End of program
```

Example: Rotation in space

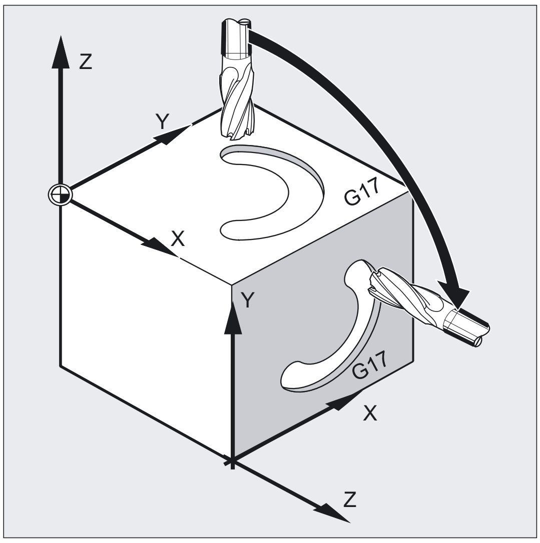
In this example, paraxial and inclined workpiece surfaces are to be machined in one setting. Requirement: The tool must be aligned perpendicular to the inclined surface in the rotated Z direction.



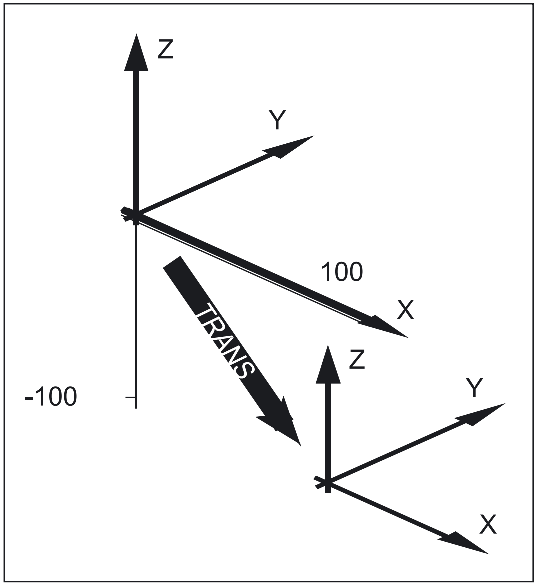
```
N10 G17 G54                  ;Working plane X/Y, workpiece zero
N20 TRANS X10 Y10            ;Absolute offset
N30 L10                      ;Subprogram call
N40 ATRANS X35               ;Additive offset
N50 AROT Y30                 ;Rotation through the Y axis
N60 ATRANS X5                ;Additive offset
N70 L10                      ;Subprogram call
N80 G0 X300 Y100 M30         ;Retraction, end of program
```

Example of multi-side machining

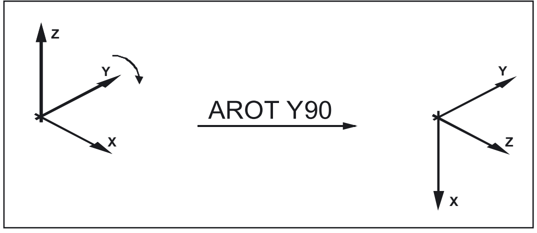
In this example, identical shapes on two perpendicular workpiece surfaces are machined by using subprograms. The setup of the infeed direction, working plane and zero point in the new coordinate system on the right-hand workpiece surface matches that of the top surface. The conditions required for subprogram execution apply as before: Working plane G17, coordinate plane X/Y, infeed direction Z.



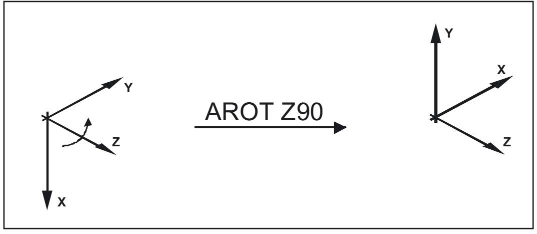
```
N10 G17 G54                   ;Working plane X/Y, workpiece zero
N20 L10                       ;Subprogram call
N30 TRANS X100 Z-100          ;Absolute offset
```



```
N40 AROT Y90                  ;Rotation of the coordinate system through Y
```



```
N50 AROT Z90                  ;Rotation of the coordinate system through Z
```



```
N60 L10                       ;Subprogram call
N70 G0 X300 Y100 M30          ;Retraction, end of program
```

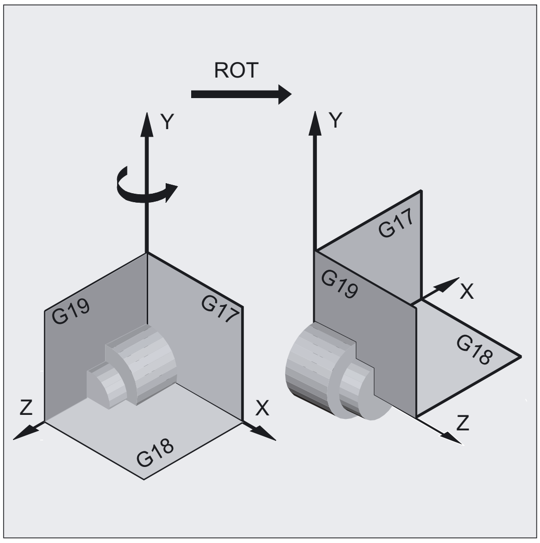
Rotation in the plane

The coordinate system is

- rotated in the plane selected with G17 to G19.
  Substituting instruction, ROT RPL or additive instruction, AROT RPL
- current plane, about which there is rotation with RPL= programmed rotation angle.

Note

See "Rotation in space" for more information.



Plane change

Warning

If you program a change of plane (G17 to G19) after a rotation, the angles of rotation programmed for the axes are retained and continue to apply in the new working plane. It is therefore advisable to deactivate the rotation before a change of plane.

Deactivate rotation

For all axes: ROT (without axis parameter)

Caution

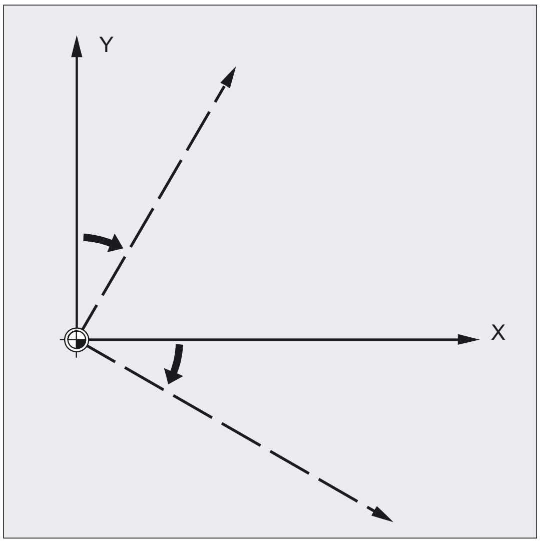
In both cases, all frame components of the previously programmed frame are reset.

Substituting instruction, ROT X Y Z

The coordinate system is rotated through the programmed angle around the specified axes. The center of rotation is the last specified settable zero offset (G54 to G599).

Caution

The ROT command cancels all frame components of the previously activated programmable frame.

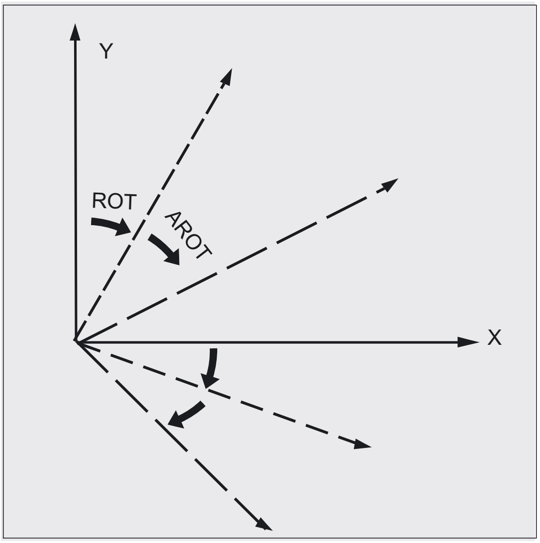


Note

A new rotation based on existing frames is programmed with AROT.

Additive instruction, AROT X Y Z

Rotation through the angle values programmed in the axis direction parameters. The center of rotation is the currently set or last programmed zero point.

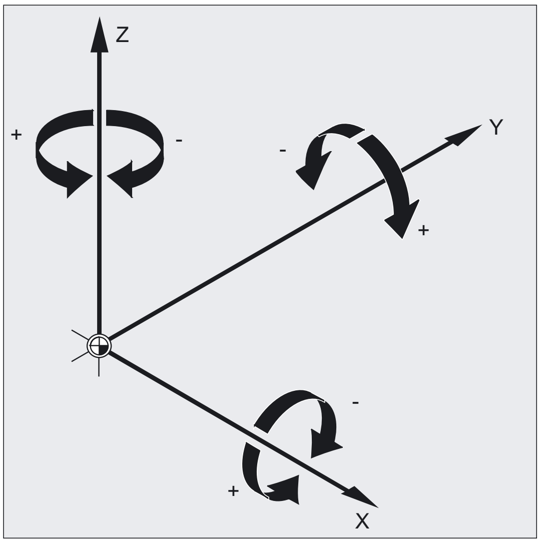


Note

For both instructions, please note the order and direction of rotation, in which the rotations are performed (see next page)!

Direction of rotation

The following is defined as the positive direction of rotation: The view in the direction of the positive coordinate axis and clockwise rotation.



Order of rotation

You can rotate up to three geometry axes simultaneously in one NC block.

The order of the RPY notation or Euler angle, through which the rotations are performed can be defined in machine data.

MD 10600: FRAME_ANGLE_INPUT_MODE =

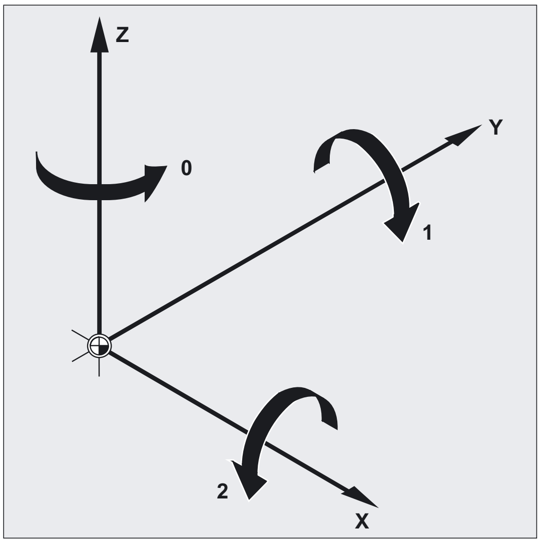
- RPY notation (RPY notation is the default setting)
- Euler angles

After that, the sequence Z, Y, X of the rotation is defined as follows:

Rotation around the 3rd geometry axis (Z)

Rotation around the 2nd geometry axis (Y)

Rotation around the 1st geometry axis (X)



This order applies if the geometry axes are programmed in a **single** block. It also applies irrespective of the input sequence. If only two axes are to be rotated, the parameter for the 3rd axis (value zero) can be omitted.

Value range with RPY angle

The angles are defined uniquely **only** within the following value ranges:

Rotation around 1st geometry axis: 180° ≤ X ≤ +180°

Rotation around 2nd geometry axis: 90° < Y < +90°

Rotation around 3rd geometry axis: 180° ≤ Z ≤ +180°

All possible rotations can be represented with this value range. Values outside the range are normalized by the control into the above range during writing and reading. This value range applies to all frame variables.

Examples of reading back in RPY

$P_UIFR[1] = CROT(X, 10, Y, 90, Z, 40)

returns on reading back

$P_UIFR[1] = CROT(X, 0, Y, 90, Z, 30)

$P_UIFR[1] = CROT(X, 190, Y, 0, Z, -200)

returns on reading back

$P_UIFR[1] = CROT(X, -170, Y, 0, Z, 160)

When frame rotation components are read and written, the value range limits must be observed to ensure that the same results are obtained for read or write, or repeat write operations.

Value range with Euler angle

The angles are defined uniquely **only** within the following value ranges:

Rotation around 1st geometry axis: 0° < X < +180°

Rotation around 2nd geometry axis: 180° ≤ Y ≤ +180°

Rotation around 3rd geometry axis: 180° ≤ Z ≤ +180°

All possible rotations can be represented with this value range. Values outside the range are normalized by the control into the above range. This value range applies to all frame variables.

Caution

To ensure the angles written are read back unambiguously, it is necessary to observe the defined values ranges.

Note

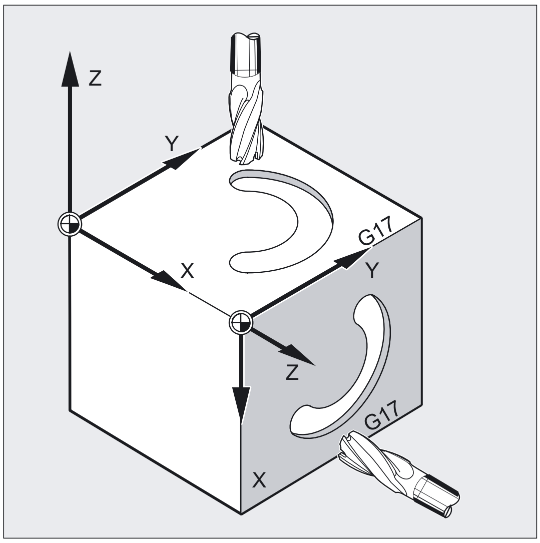
If you want to define the order of the rotations individually, program the desired rotation successively for each axis with AROT.

References: Function Description of the Basic Machine /FB1/, Chapter "Frames"

The working plane also rotates

The working plane defined with G17, G18 or G19 rotates with the spatial rotation.

Example: Working plane G17 X/Y, the workpiece coordinate system is positioned on the top surface of the workpiece. Translation and rotation is used to move the coordinate system to one of the side surfaces. Working plane G17 also rotates. This feature can be used to program plane destination positions in X/Y coordinates and the infeed in the Z direction.



Requirement:
The tool must be positioned perpendicular to the working plane. The positive direction of the infeed axis points in the direction of the toolholder. Specifying CUT2DF activates the tool radius compensation in the rotated plane. For more information please refer to Section "2 1/2 D Tool Compensation, CUT2D CUT2DF".

6.5 Programmable frame rotations with solid angles (ROTS, AROTS, CROTS)

Function

Orientations in space can be specified by means of frame rotations with solid angles ROTS, AROTS, CROTS. Programming commands ROTS and AROTS behave analogously to ROT and AROT.

Programming

When programming the solid angles X and Y the new X-axis lies in the old ZX plane.

```
ROTS X... Y...
```

Or

```
AROTS X... Y...
```

Or

```
CROTS X... Y...
```

When programming solid angles Z and X the new Z-axis lies in the old YZ plane.

```
ROTS Z... X...
```

Or

```
AROTS Z... X...
```

Or

```
CROTS Z... X...
```

When programming the solid angles Y and Z the new Y-axis lies in the old XY plane.

```
ROTS Y... Z...
```

Or

```
AROTS Y... Z...
```

Or

```
CROTS Y... Z...
```

Parameters

ROTS,	Frame rotations with solid angles for spatial orientation of a plane absolute, referred to the currently valid frame with set workpiece zero for G54 to G599.
AROTS,	Frame rotations with solid angles for spatial orientation of a plane additive, referred to the currently valid frame with set or programmed zero point.
CROTS,	Frame rotations with solid angles for spatial orientation of a plane, referred to the valid frame in the data management with rotation in the specified axes.
X Y Z	A maximum of two solid angles may be specified.
RPL	Rotation in the plane: Angle through which the coordinate system is rotated (plane set with G17-G19).

6.6 Programmable scale factor (SCALE, ASCALE)

Function

With SCALE/ASCALE you can program scale factors for all path, synchronized axes and positioning axes in the direction of the axis specified in each case. This enables the size of a shape to be changed. You can thus program taking similar geometrical shapes or different shrinkages into account.

Deactivate scaling factor

For all axes: SCALE (without axis parameter). All frame components of the previously programmed frame are reset.

Programming

`SCALE X… Y… Z…` (substituting instruction programmed in a separate NC block)

Or

`ASCALE X… Y… Z…` (additive instruction programmed in a separate NC block)

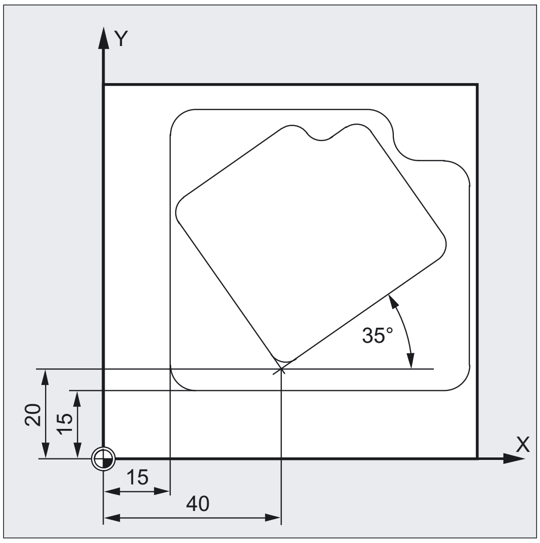
Parameters

SCALE,	Absolute enlargement/reduction with reference to the currently valid coordinate system set with G54 to G599
ASCALE,	Additive enlargement/reduction with reference to the currently valid set or programmed coordinate system
X Y Z	Scale factor in the direction of the specified geometry axis

Example of milling

With this workpiece, the two pockets occur twice, but in different sizes and at different angles to each other. The machining sequence is stored in a subprogram.

Use zero offset and rotation to set each of the workpiece zeroes, reduce the contour with a scale and then call the subprogram up again.



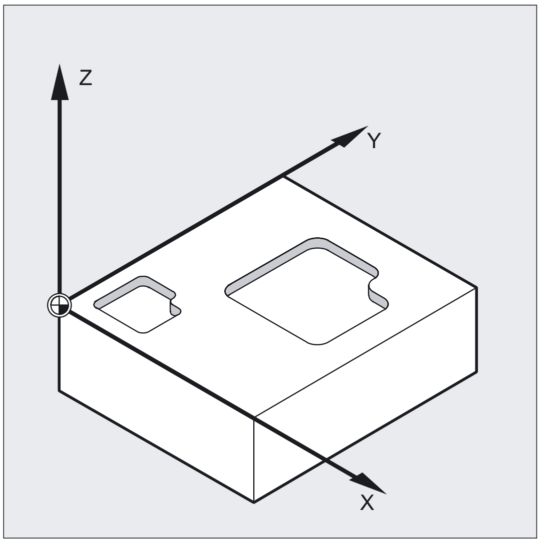
```
N10 G17 G54                  ;Working plane X/Y, workpiece zero
N20 TRANS X15 Y15            ;Absolute offset
N30 L10                      ;Machine large pocket
N40 TRANS X40 Y20            ;Absolute offset
N50 AROT RPL=35              ;Rotation in the plane through 35°
N60 ASCALE X0.7 Y0.7         ;Scale factor for the small pocket
N70 L10                      ;Machine small pocket
N80G0 X300 Y100 M30          ;Retraction, end of program
```

Substituting instruction, SCALE X Y Z

You can specify an individual scale factor for each axis, by which the shape is to be reduced or enlarged. The scale refers to the workpiece coordinate system set with G54 to G57.

Notice

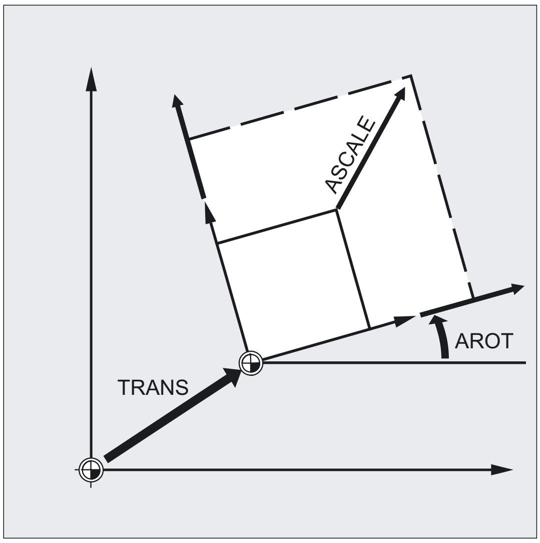
The SCALE command cancels all frame components of the previously activated programmable frame.



Additive instruction, ASCALE X Y Z

You can program scale changes, which are to be added to existing frames by using the ASCALE command. In this case, the last valid scale factor is multiplied by the new one.

The currently set or last programmed coordinate system is used as the reference for the scale change.

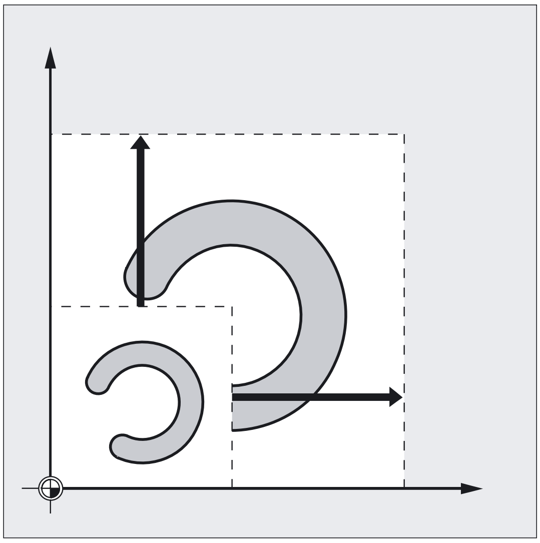


Note

If you program an offset with ATRANS after SCALE, the offset values are also scaled.

Caution

Please take great care when using different scale factors! Example: Circular interpolations can only be scaled using identical factors. You can, however, use different scale factors to program distorted circles, for example.



6.7 Programmable mirroring (MIRROR, AMIRROR)

Function

MIRROR/AMIRROR can be used to mirror workpiece shapes on coordinate axes. All traversing movements, which are programmed after the mirror call, e.g., in the subprogram, are executed in the mirror image.

Programming

`MIRROR X0 Y0 Z0` (substituting instruction programmed in a separate NC block)

Or

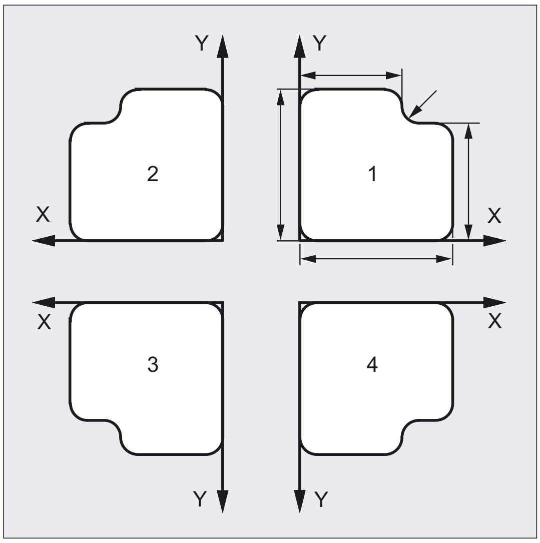
`AMIRROR X0 Y0 Z0` (additive instruction programmed in a separate NC block)

Parameters

MIRROR	Absolute mirror image with reference to the currently valid coordinate system set with G54 to G599
AMIRROR	Additive mirror image with reference to the currently valid set or programmed coordinate system
X Y Z	Geometry axis whose direction is to be changed. The value specified here can be chosen freely, e.g., X0 Y0 Z0.

Example of mirroring milling

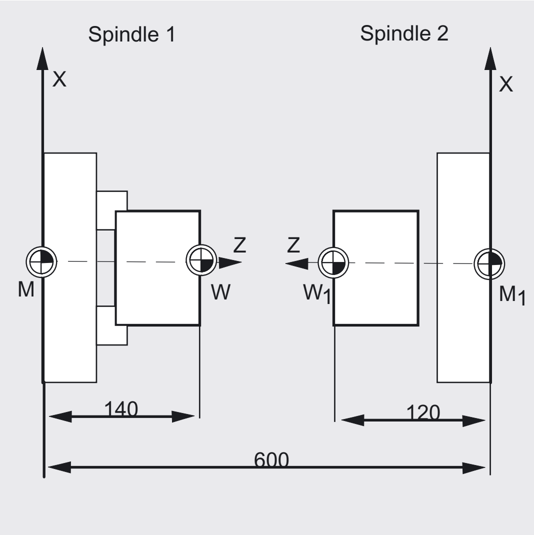
Program the contour shown here once as a subprogram and generate the three other contours with a mirror operation. The workpiece zero is located at the center of the contours.



```
N10 G17 G54                 ;Working plane X/Y, workpiece zero
N20 L10                     ;Machine first contour, top right
N30 MIRROR X0               ;Mirror X axis (the direction is changed in
                            ;X)
N40 L10                     ;Machine second contour, top left
N50 AMIRROR Y0              ;Mirror Y axis (the direction is changed in
                            ;Y)
N60 L10                     ;Machine third contour, bottom left
N70 MIRROR Y0               ;MIRROR cancels previous frames. Mirror
                            ;Y axis (the direction is changed in Y)
N80 L10                     ;Machine fourth contour, bottom right
N90 MIRROR                  ;Deactivate mirroring
N100 G0 X300 Y100 M30       ;Retraction, end of program
```

Example of rotating mirroring

The actual machining is stored as a subprogram and the execution at the respective spindles is done by means of mirrorings and offsets.



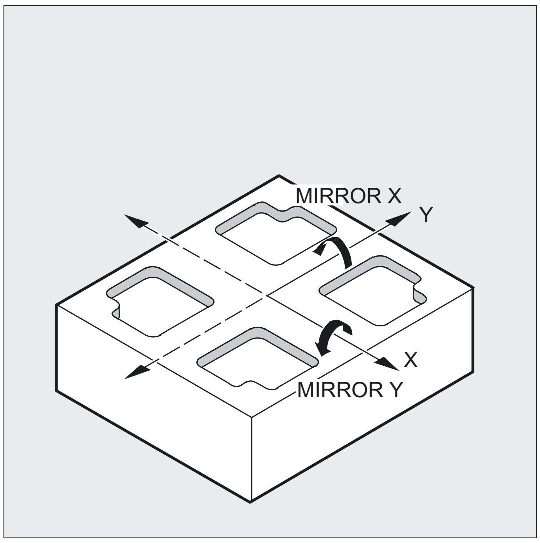
```
N10 TRANS X0 Z140           ;Zero offset to W
N.. ...                     ;Machine first side with spindle 1
N30 TRANS X0 Z600           ;Zero offset to spindle 2
N40 AMIRROR Z0              ;Mirroring of the Z axis
N50 ATRANS Z120             ;Zero offset to W1
N.. ...                     ;Machine second side with spindle 2
```

Substituting instruction, MIRROR X Y Z

The mirror is programmed by means of an axial change of direction in the selected working plane.

Example: Working plane G17 X/Y

The mirror (on the Y axis) requires a change of direction on X and is subsequently programmed with MIRROR X0. The contour is then mirrored on the opposite side of the mirror axis Y.



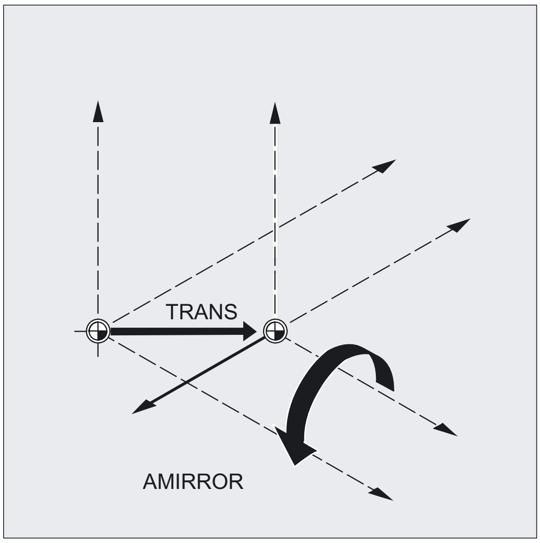
The mirror image refers to the coordinate axes set with G54 to G57.

Caution

The MIRROR command cancels all previously set programmable frames.

Additive instruction, AMIRROR X Y Z

A mirror image, which is to be added to an existing transformation, is programmed with AMIRROR. The currently set or last programmed coordinate system is used as the reference.



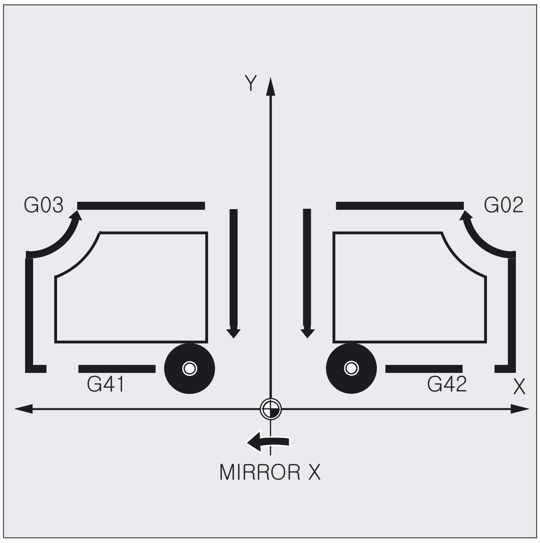
Deactivate mirroring

For all axes: MIRROR (without axis parameter)

All frame components of the previously programmed frame are reset.

Note

The mirror command causes the control to automatically change the path compensation commands (G41/G42 or G42/G41) according to the new machining direction.



The same applies to the direction of circle rotation (G2/G3 or G3/G2).

Note

If you program an additive rotation with AROT after MIRROR, you may have to work with reversed directions of rotation (positive/negative or negative/positive). Mirrors on the geometry axes are converted automatically by the control into rotations and, where appropriate, mirrors on the mirror axis specified in the machine data. This also applies to settable zero offsets.

Machine manufacturer

- You can set the axis, around which mirroring is performed, via machine data MD.
  MD 10610 = 0: Mirroring is performed in relation to the programmed axis (negation of values).
  MD 10610 = 1 or 2 or 3: Depending on the data setting, mirroring is performed in relation to a specific reference axis (1=X axis; 2=Y axis; 3=Z axis) and rotations of two other geometry axes.
- MD10612 MIRROR_TOGGLE = 0 can be used to define that the programmed values are always evaluated. A value of 0, i.e., MIRROR X0, deactivates the mirroring of the axis, and values not equal to 0 cause the axis to be mirrored if it is not already mirrored.

6.8 Frame generation according to tool orientation (TOFRAME, TOROT, PAROT)

Function

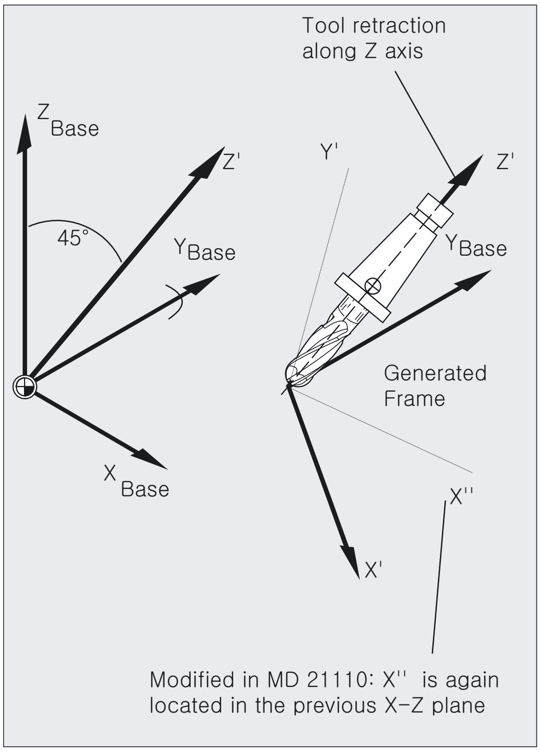
TOFRAME generates a rectangular frame whose Z axis coincides with the current tool orientation. You can use this function to retract the tool after a tool breakage in a 5-axis program without collision, simply by retracting the Z axis. The resulting frame describing the orientation is written in the system variable for the programmable frame $P_PFRAME.

Only the rotation component is overwritten with TOROT in the programmed frame. All other components remain unchanged.

PAROT aligns the workpiece on the workpiece coordinate system (WCS).

Machine manufacturer

The position of the two axes X and Y can be defined in MD21110: X_AXES_IN_OLD_X_Z_PLANE where X is rotated about Z into the existing X-Z plane.



Programming

TOFRAME	Frame rotation in tool direction
Or	
TOFRAMEZ or TOFRAMEY or TOFRAMEX	Z/Y/X axis parallel to tool orientation
Or	
TOROTOF	Frame rotation in tool direction OFF
Or frame rotation on with	
TOROT or TOROTZ or TOROTY or TOROTX	Z/Y/X axis parallel to tool orientation
Or	
PAROT	Align workpiece coordinate system (WCS) on workpiece
Or	
PAROTOF	Deactivate workpiece-related frame rotation

Parameters

TOFRAME	Frame rotation in tool direction The new frame, whose Z axis is pointing in the tool direction, is applicable after the block containing TOFRAME. TOROTOF deactivates the frame rotation in tool direction.
TOFRAMEZ	Z axis parallel to tool orientation
TOFRAMEY	Y axis parallel to tool orientation
TOFRAMEX	X axis parallel to tool orientation
TOROTOF	Frame rotation in tool direction OFF
TOROT	Frame rotation ON Z axis parallel to tool orientation. The rotation defined by TOROT is the same as that defined with TOFRAME.
TOROTZ	Frame rotation ON Z axis parallel to tool orientation
TOROTY	Frame rotation ON Y axis parallel to tool orientation
TOROTX	Frame rotation ON X axis parallel to tool orientation
PAROT	Align workpiece coordinate system (WCS) on workpiece. Translations, scaling and mirroring in the active frame remain valid. The workpiece-related frame rotation activated with PAROT is deactivated with PAROTOF.
PAROTOF	Deactivate workpiece-related frame rotation

Milling with working plane G17

TOFRAME or TOROT defines frames whose Z axes point in the tool direction. This definition is tailored to milling operations, for which working plane G17 X/Y of the 1st – 2nd geometry axis is typically active.

Turning with working plane G18 or G19

Turning operations in particular, and active G18 or G19 in general, require frames, with which the tool is aligned in the X or Y axis. A frame of this type can be defined with G codes

- TOFRAMEX TOROTX
- TOFRAMEY TOROTY
- TOFRAMEZ

TOROTZ. This functionality of TOFRAME and TOFRAMEZ or TOROT and TOROTZ is identical in each case.

Example of TOFRAME

```
N100 G0 G53 X100 Z100 D0
N120 TOFRAME
N140 G91 Z20                   ;Frame TOFRAME is included, all programmed geometry
                               ;axis movements
                               ;refer to TOFRAME
N160 X50
...
```

Milling with working plane G17

TOFRAME or TOROT defines frames whose Z axes point in the tool direction. This definition is tailored to milling operations, for which working plane G17 X/Y of the 1st – 2nd geometry axis is typically active.

Turning with working plane G18 or G19

Turning operations in particular, and active G18 or G19 in general, require frames, with which the tool is aligned in the X or Y axis. A frame of this type can be defined with G codes

- TOFRAMEX TOROTX
- TOFRAMEY TOROTY
- TOFRAMEZ

TOROTZ. This functionality of TOFRAME and TOFRAMEZ or TOROT and TOROTZ is identical in each case.

Assigning axis direction

If one of the G codes TOFRAMEX, TOFRAMEY, TOROTX, TOROTY is programmed instead of TOFRAME(Z) or TOROT(Z), the axis directions are assigned as shown in the table below:

TOFRAME (Z), TOROT (Z)	**TOFRAMEY, TOROTY**	**TOFRAMEX, TOROTX**	
Z	Y	X	Tool direction (applicate)
X	Z	Y	Secondary axis (abscissa)
Y	X	Z	Secondary axis (ordinate)

Note

After tool orientation has been programmed with TOFRAME, all the programmed geometry axis movements refer to the frame generated by this programming.

Note

Separate system frame for TOFRAME or TOROT

The frames resulting from TOFRAME or TOROT can be written in a separate system frame $P_TOOLFRAME.

This can be done by setting bit 3 in machine data MD 28082: MM_SYSTEM_FRAME_MASK. The programmable frame remains unchanged. Differences occur when the programmable frame is processed further elsewhere.

Note

NC command TOROT ensures consistent programming with active orientable tool carriers for each kinematic type. Just as in the situation for rotatable toolholders, PAROT can be used to activate a rotation of the work table. This defines a frame, which changes the position of the workpiece coordinate system in such a way that no compensatory movement is performed on the machine. Language command PAROT is not rejected if no orientable toolholder is active.

References: For further explanations about machines with orientable toolholder, see:

/PGA/, Programming Guide Advanced, chapter "Tool Orientation"
/FB1/Description of Functions, W1 "Orientable Toolholders”

6.9 Deselect frame (G53, G153, SUPA, G500)

Function

When executing certain processes, such as approaching the tool change location or initial setting, various frame components must be defined and suppressed at different times. Set frames can either be deactivated modally or suppressed non-modally.

The programmable frames are cleared by specifying the TRANS, ROT, SCALE, MIRROR component without an axis.

Deactivate coordinate transformation

A distinction must be made here between non-modal suppression and modal deactivation.

Programming

```
G53
```

Or

```
G153
```

Or

```
SUPA
```

Or

```
G500
```

Parameters

Non-modal suppression:

G53	Deactivation of all programmable and settable frames
G153	Deactivation of all programmable, settable and basic frames
SUPA	Deactivation of all programmable, settable frames, DRF handwheel offsets, external zero offsets and preset offset

Modal deactivation:

G500	Deactivation of all settable frames if G500 does not contain a value

Deleting FRAMES:

TRANS, ROT, SCALE, MIRROR	Delete programmable frames without axis specification

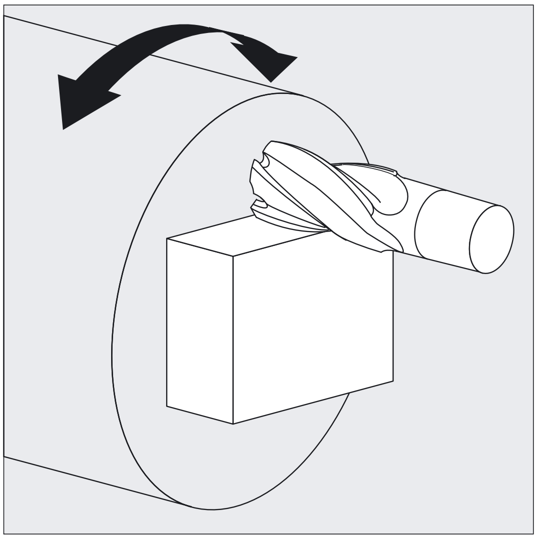
References: For further explanations of offsets with frames, see:
/FB1/Description of Functions, C2 "Axes, Coordinate Systems, Frames".

6.10 Milling on turned parts (TRANSMIT)

Function

The TRANSMIT function enables the following:

- Face machining on turned parts in the turning clamp (drill-holes, contours).
- A cartesian coordinate system can be used to program these machining operations.
- The control maps the programmed traversing movements of the Cartesian coordinate system onto the traversing movements of the real machine axes (standard situation):
  - Rotary axis
  - Infeed axis perpendicular to rotary axis
  - Longitudinal axis parallel to rotary axis (the linear axes are located vertically above each other).
- A tool center offset relative to the turning center is permitted.
- The velocity control makes allowance for the limits defined for the rotations.



For more detailed information, see
References: /PGA/ Programming Guide Advanced, Chapter "Transformations"

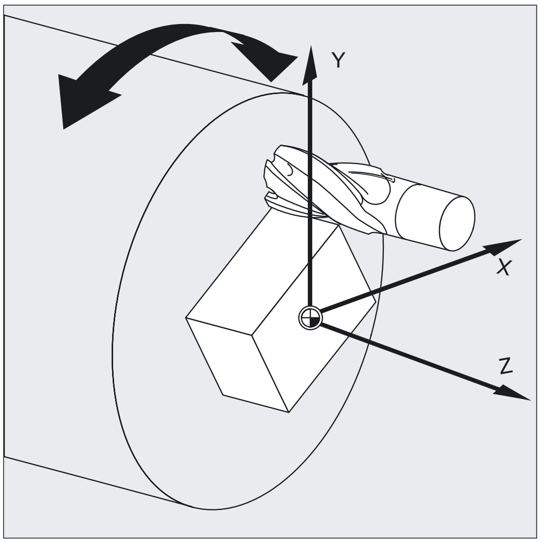
Programming

`TRANSMIT` or `TRANSMIT(n)`

Parameter

TRANSMIT	Activates the first declared TRANSMIT function
TRANSMIT(n)	Activates the nth declared TRANSMIT function; n can be up to 2 (TRANSMIT(1) is the same as TRANSMIT).

Example of activating TRANSMIT function



```
N10 T1 D1 G54 G17 G90 F5000 G94       ;Tool selection
N20 G0 X20 Z10 SPOS=45                ;Approach start position
N30 TRANSMIT                          ;Activate TRANSMIT function
N40 ROT RPL=-                         ;Set frame
N50 ATRANS X
N60 G1 X10 Y                          ;Four-edge roughing
N70 X-
N80 Y10
N90 X10
N100 Y-
N110 ...
```

6.11 Cylinder surface transformation (TRACYL)

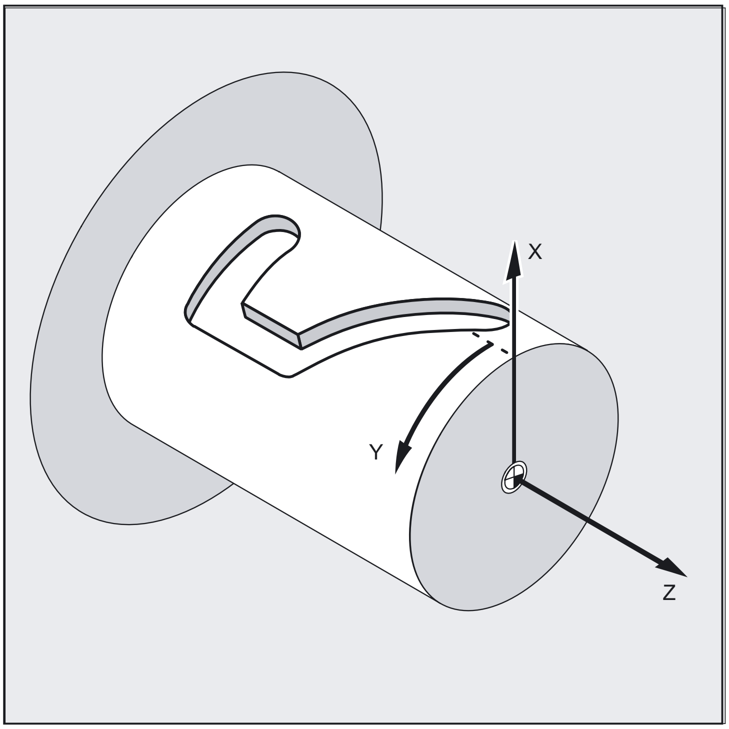
Function

The TRACYL cylinder surface transformation function can be used to:

Machine

- Longitudinal grooves on cylindrical bodies,
- Transverse grooves on cylindrical bodies,
- Grooves with any path on cylindrical bodies.

The path of the grooves is programmed with reference to the unwrapped, level surface of the cylinder.



For more detailed information, see
References: /PGA/ Programming Guide, Advanced, Chapter "Transformations"

Programming

`TRACYL(d)` or `TRACYL(d,t)`

Parameter

TRACYL(d)	Activates the first declared TRACYL function
TRACYL (d,n)	Activates the nth declared TRACYL function. N can be up to 2, TRACYL(d,1) is the same as TRACYL(d).
d	Value for the current diameter of the cylinder to be machined.

6.12 Deselect transformation (TRAFOOF)

Function

An active TRANSMIT polar transformation, TRACYL cylinder surface transformation, TRAAANG inclined axis transformation or TRAORI orientation transformation can be deactivated with TRAFOOF.

Programming

`TRANSMIT` or `TRACYL` or `TRAANG` or `TRAORI`

`TRAFOOF`

Parameters

TRANSMIT	TRANSMIT function (polar transformation) is active
TRACYL	TRACYL function (cylinder surface transformation) is active
TRAANG	TRAANG function (inclined axis transformation) is active
TRAORI	TRAORI function (orientation transformation) is active
TRAFOOF	Deactivate an active transformation within the relevant channel

Note

An active TRANSMIT transformation is likewise deactivated if one of the other transformations is activated in the relevant channel (e.g., TRACYL, TRAANG, TRAORI).

An active TRACYL, TRAANG or TRAORI transformation can also be deactivated by activating one of the other transformations within the channel concerned.

6.13 Deselect DRF (handwheel) offsets, overlaid motions (DRFOF, CORROF)

Function

DRFOF can be used to deactivate all active axes of the channel for DRF handwheel offsets. For instance, if a particular axis with an overlaid motion or a position offset interpolates, the instruction CORRROF can be used to deactivate either the DRF offsets or the position offset for this axis. The axis is not traversed.

Programming

```
DRFOF
```

Or

```
CORROF(axis,string[axis,string])
```

Or

```
CORROF(axis,string)
```

Or

```
CORROF(axis)
```

Or

```
CORROF()
```

Parameters

Modal deactivation:

DRFOF	Deactivation (deselection) of DRF handwheel offsets for all active axes in the channel
CORROF(axis,DRF[AXIS,AA_OFF])	Deactivation (deselection) of axial DRF offsets and the position offset for individual axes as a result of $AA_OFF
CORROF(axis)	All active overlaid motions are deselected
Axis	Axis identifiers (for channel, geometry or machine axis)
String == DRF	DRF offset of axis is deselected
String == AA_OFF	Position offset of axis is deselected due to $AA_OFF

The following expansions are possible:

String == ETRANS	An active zero offset is deselected
String == FTOCOF,	Acts like FTOCOF (deactivate online tool offset)

Example of axial DRF deselection

A DRF offset is generated in the X axis by DRF handwheel traversal. No DRF offsets are operative for any other axes in the channel.

`N10 CORROF(X,"DRF")` acts like DRFOF( )

A DRF offset is generated in the X and Y axes by DRF handwheel traversal. No DRF offsets are operative for any other axes in the channel.

```
N10 CORROF(X,"DRF")                       ;Only the DRF offset of the X axis is
                                          ;deselected, the X axis does not move
                                          ;The DRF offset of the Y axis is retained
                                          ;Both offsets would have been deselected
                                          ;with DRFOF()
```

Example of axial DRF deselection and $AA_OFF deselection

A DRF offset is generated in the X axis by DRF handwheel traversal. No DRF offsets are operative for any other axes in the channel.

```
N10 WHEN TRUE DO $AA_OFF[X] = 10          ;A position offset == 10 is interpolated
G4 F5                                     ;for the X axis
N70 CORROF(X,"DRF",X,"AA_OFF")            ;Only the DRF offset of the X axis is
                                          ;deselected, the X axis does not move
                                          ;The DRF offset of the Y axis is ;retained
```

Example of deselecting AA_OFF

A position offset of the X axis is deselected with: CORROF(X,"AA_OFF") with $AA_OFF[X] = 0 and added to the current position of the X axis.

The following programming example shows the relevant programming commands for the X axis that was previously interpolated with a position offset of 10:

```
N10 WHEN TRUE DO $AA_OFF[X] = 10          ;A position offset == 10 is interpolated
G4 F5                                     ;for the X axis
N80 CORROF(X,"AA_OFF")                    ;Delete position offset of X axis
                                          ;the X axis does not move
```

Description

CORROF

A preprocessing stop is initiated and the position component of the deselected overlaid motion (DRF offset or position offset) is transferred to the position in the basic coordinate system. Since no axis is traversed, the value of $AA_IM[axis] does not change. Owing to the deselected overlaid motion, only the value of system variable $AA_IW[axis] is altered.

After the position offset, e.g., for one axis, has been deselected by
$AA_OFF, the system variable
$AA_OFF_VAL of this axis is zero.

Setting bit 2 of MD 36750: AA_OFF_MODE to "1" when $AA_OFF is changed enables interpolation of the position offset as an overlaid motion in JOG mode.

Note

CORROF is possible only from the parts program, not via synchronized actions.

Alarm 21660 is output if a synchronized action is active when the position offset is deselected via parts program command CORROF(axis,"AA_OFF"). $AA_OFF is deselected simultaneously and not set again. If the synchronized action becomes active later in the block after CORROF, $AA_OFF remains set and a position offset is interpolated.

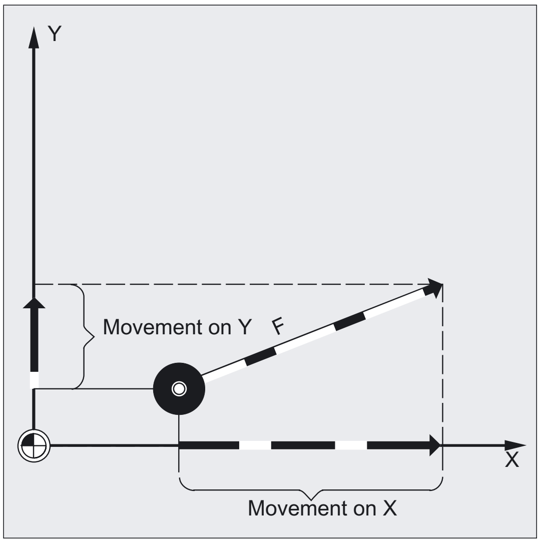
If a CORROF command has been programmed for an axis and this axis is active in a different channel, then an axis replacement sends the axis to the other channel with machine data 30552: AUTO_GET_TYPE = 0. This causes the DRF offset and any other position offset to be deselected.

Feedrate Control and Spindle Motion 7

7.1 Feedrate (G93, G94, G95 or F..., FGROUP, FGREF)

Function

You can use the above commands to set the feedrates in the NC program for all axes participating in the machining sequence.



The path feedrate is generally composed of the individual speed components of all geometry axes participating in the movement and refers to the center point of the cutter or the tip of the turning tool.

The following feedrate types can be programmed:

- Path feedrate with G commands G93, G94, G95 on axes participating in movement
- Feedrate F for path axes
- Feedrate F for synchronized axes
- Feedrate F applies to all axes programmed under FGROUP
- Feedrate for synchronized-/path axes with limit speed FL

Note

The inverse-time feedrate 1/min G93 is not implemented for 802D.

Programming

`G93` or `G94` or `G95`

`F…`

or

`FGROUP (X, Y, Z, A, B, …)`

or

`FL[axis]=…`

or

`FGREF[axis name]`=reference radius

Parameters

G93	Inverse-time feedrate 1/rpm
G94	Feedrate in mm/min or inches/min or in deg/min
G95	Feedrate in mm/rev or inches/rev with reference to the speed of the master spindle – generally the cutting spindle or the main spindle on the turning machine
F…	Feedrate value in unit defined by G93, G94, G95
FGROUP	Feedrate value F valid for all axes specified in FGROUP
FL	Limit speed for synchronized/path axes; the unit set with G94 applies (max. rapid traverse) One FL value can be programmed per axis. The axis identifiers of the basic coordinate system must be used (channel axes or geometry axes).
FGREF	Effective radius (reference radius) for the rotary axes entered in FGROUP
Axis	Channel axis or geometry axes or orientation axes
X Y Z	Movement of the specified geometry axis
A B C	Axis identifier for rotary axis to be traversed

Example of operating principle of FGROUP

The following example illustrates the effect of FGROUP on the path and the path feedrate. The variable $AC_TIME contains the time from the start of the block in seconds. It can only be used in synchronized actions. See /FBSY/, Synchronized Actions.

```
N100 G0 X0 A0
N110 FGROUP(X,A)
N120 G91 G1 G710 F100         ;Feedrate=100 mm/min or 100 deg/min
N130 DO $R1=$AC_TIME
N140 X10                      ;Feedrate=100 mm/min, path dist.=10 mm, R1=approx. 6 s
N150 DO $R2=$AC_TIME
N160 X10 A10                  ;Feedrate=100 mm/min, path dist.=14.14 mm, R2=approx.
                              ;8 s
N170 DO $R3=$AC_TIME
N180 A10                      ;Feedrate=100 degrees/min, path dist.=10 degrees,
                              ;R3=approx. 6 s
N190 DO $R4=$AC_TIME
N200 X0.001 A10               ;Feedrate=100 mm/min, path dist.=10 mm, R4=approx. 6 s
N210 G700 F100                ;Feedrate=2540 mm/min or 100 degrees/min
N220 DO $R5=$AC_TIME
N230 X10                      ;Feedrate=2540 mm/min, path dist.=254 mm, R5=approx.
                              ;6 s
N240 DO $R6=$AC_TIME
N250 X10 A10                  ;Feedrate=2540 mm/min, path dist.=254.2 mm,
                              ;R6=approx. 6 s
N260 DO $R7=$AC_TIME
```

```
N270 A10                        ;Feedrate=100 degrees/min, path dist.=10 degrees,
                                ;R7=approx. 6 s
N280 DO $R8=$AC_TIME
N290 X0.001 A10                 ;Feedrate=2540 mm/min, path dist.=10 mm, R8=approx.
                                ;0.288 s
N300 FGREF[A]=360/(2*$PI)       ;1 degree=1 inch above the effective radius
N310 DO $R9=$AC_TIME
N320 X0.001 A10                 ;Feedrate=2540 mm/min, path dist.=254 mm, R9=approx.
                                ;6 s
N330 M30
```

Example of traversing synchronized axes with limit speed FL

The path velocity of the path axes is reduced if the synchronized axis reaches the limit speed.

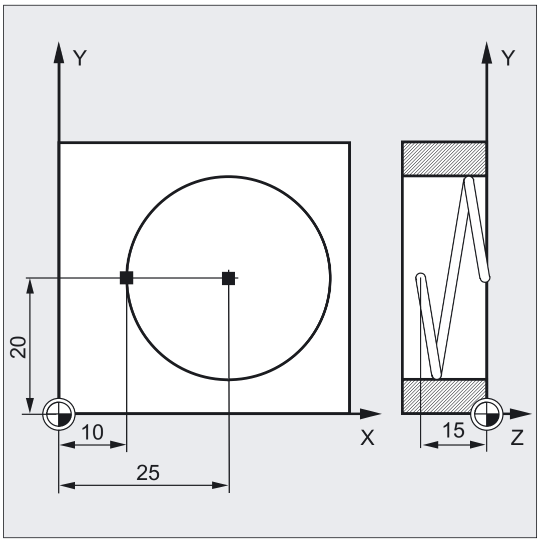
Example, Z is a synchronized axis:

```
N10 G0 X0 Y0
N20 FGROUP(X)
N30 G1 X1000 Y1000 G94 F1000 FL[Y]=500
N40 Z-50
```

One FL value can be programmed per axis. The axis identifiers of the basic coordinate system must be used (channel axes or geometry axes).

Example of helical interpolation

Path axes X and Y traverse with the programmed feedrate, the infeed axis Z is a synchronized axis.



```
N10 G17 G94 G1 Z0 F500              ;Tool infeed
N20 X10 Y20                         ;Approach start position
N25 FGROUP(X, Y)                    ;Axes X/Y are path axes, Z is a
                                    ;synchronized axis
N30 G2 X10 Y20 Z-15 I15 J0 F1000    ;On the circular path, the feedrate is
FL[Z]=200                           ;1000 mm/min.
                                    ;Traversing in the Z direction is synchronized.
...
N100 FL[Z]=$MA_AX_VELO_LIMIT[0,Z]   ;The limit velocity is deselected
                                    ;when the velocity
                                    ;value is read from the MD.
N110 M30                            ;End of program
```

Feedrate G93, G94, G95

All of the commands are modal. If the G feedrate command is switched between G93, G94 or G95 the path feedrate must be reprogrammed. The feedrate can also be specified in deg/rev when machining with rotary axes.

Feedrate F for path axes

The feedrate is specified with address F. Depending on the default setting in the machine data, the units of measurement specified with the G commands are either in mm or inch.

One F value can be programmed per NC block. The unit for the feedrate is defined in one of the above G commands. The feed F acts only on path axes and remains active until a new feedrate is programmed. Separators are permitted after address F.

Example: F100 or F 100 or F.5 or F=2*FEED

Feedrate for synchronized axes

The feedrate F programmed at address F applies to all the path axes programmed in the block, but not to synchronized axes. The synchronized axes are controlled such that they require the same time for their path as the path axes, and all axes reach their end point at the same time.

Traverse synchronized axes with limit speed FL

With this command, synchronized/path axes are traversed at their limit speed FL.

Traverse synchronized axes with path velocity F, FGROUP

With FGROUP, you define whether a path axis is to be traversed with path feed or as a synchronized axis. In helical interpolation, for example, you can define that only two geometry axes, X and Y, are to be traversed at the programmed feedrate. The infeed axis Z is the synchronized axis in this case.

Example: `N10 FGROUP(X, Y)`

Change FGROUP

1. By reprogramming another FGROUP instruction.
   Example: FGROUP(X, Y, Z)
2. With FGROUP () without axis parameter

Afterwards, the initial setting in the machine data applies – the geometry axes again traverse in the path axis grouping.

Note

You must program channel axis names with FGROUP.

Caution

The FGREF evaluation also works if only rotary axes are programmed in the block. The normal F value interpretation as degree/min applies in this case only if the radius reference corresponds to the FGREF default, when

G71/G710: FGREF[A]=57.296

G70/G700: FGREF[A]=57.296/25.4

Units of measurement and calculation

Machine manufacturer

See machine manufacturer's specifications.

Units of measurement for feedrate F

You can use the following G commands to define the units of measurement for the feed input. Feedrate functions are not affected by G70/G71.

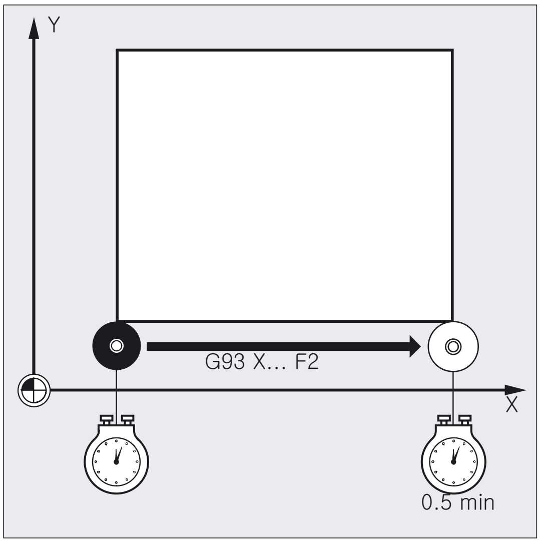
Note

With G700/G710, feedrate values F are interpreted as geometrical parameters in the measuring system set by G function (G700: [inch/min]; G710: [mm/min]).

Feedrate G93

Unit 1/rpm. The inverse-time feedrate specifies the time required to execute the motion commands in a block.

Example: `N10 G93 G01 X100 F2` means: the programmed path is traversed in 0.5 min.



Note

If the path lengths vary greatly from block to block, a new F value should be specified in each block with G93. The feedrate can also be specified in deg/rev when machining with rotary axes.

Unit of measurement for synchronized axes with limit speed FL

The unit of measurement set for F by G command (G70/G71) also applies to FL. If FL is not programmed, rapid traverse velocity is used. FL is deselected by assignment to MD $MA_AX_VELO_LIMIT.

Unit of measurement for rotary and linear axes

For linear and rotary axes, which are combined with FGROUP and traverse a path together, the feed is interpreted in the unit of measurement of the linear axes. Depending on the default for G94/G95: mm/min or inch/min and mm/rev or inch/rev.

The tangential velocity of the rotary axis in mm/min or inch/min is calculated according to the following formula:

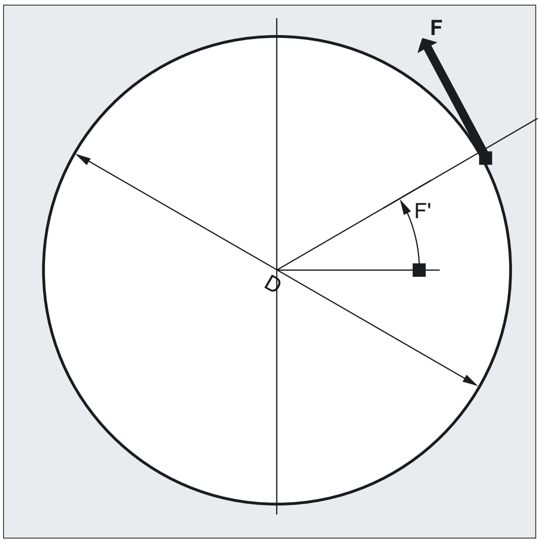
*F[mm/min] = F'[degrees/min] * π * D[mm]/360[degrees]*

F: Tangential velocity

F': Angular velocity

π: Circle constant

D: Diameter



Traverse rotary axes with path velocity F, FGREF

For machining operations, in which the tool or the workpiece or both are moved by a rotary axis, the effective machining feedrate is to be interpreted as a path feed in the usual way by reference to the F value. This requires the specification of an effective radius (reference radius) **FGREF** for each of the rotary axes involved.

The unit of the reference radius depends on the G70/G71/G700/G710 setting.

All axes involved must be included in the FGROUP command, as before, in order to be evaluated in the calculation of the path feed.

In order to ensure compatibility with the behavior with no FGREF programming, the factor 1 degree = 1mm is activated on system powerup and RESET. This corresponds to a reference radius of FGREF=360 mm/(2π)=57.296 mm.

Note

This default setting is independent of the active basic system MD 10240: SCALING_SYSTEM_IS_METRIC and of the currently active inch/metric G code.

Special situations with the following programming:

```
N100 FGROUP(X,Y,Z,A)
N110 G1 G91 A10 F100
N120 G1 G91 A10 X0.0001 F100
```

the F value programmed in N110 is evaluated as a rotary axis feedrate in deg/min, while the feedrate weighting in N120 is either 100 inch/min or 100 mm/min depending on the currently active inch/metric setting.

Path reference factors for orientation axes with FGREF

With orientation axes the mode of operation of the FGREF[ ] factors is dependent on whether the change in the orientation of the tool is implemented by rotary axis or vector interpolation.

In the case of **rotary axis interpolation**, the relevant FGREF factors of the orientation axes are calculated, as for rotary axes, individually as reference radius for the axis paths.

In the case of **vector interpolation**, an effective FGREF factor, which is calculated as the geometric mean value of the individual FGREF factors, is applied.

*FGREF[eff] = n-te root of:[(FGREF[A] * FGREF[B]...)]*

They are:

A: Axis identifier of 1st orientation axis

B: Axis identifier of 2nd orientation axis

C: Axis identifier of 3rd orientation axis Number of orientation axes

Example: There are two orientation axes for a standard 5-axis transformation, and the effective factor is thus the root of the product of the two axial factors:

*FGREF[eff] = square root of:[(FGREF[A] * FGREF[B])]*

Note

With the effective factor for orientation axes FGREF, it is therefore possible to define a reference point on the tool, to which the programmed path feedrate refers.

7.2 Traversing positioning axes (POS, POSA, POSP, FA, WAITP, WAITMC)

Function

Positioning axes are traversed independently of the path axes at a separate, axis-specific feedrate. There are no interpolation commands. With the POS/POSA/POSP commands, the positioning axes are traversed and the sequence of motions coordinated at the same time. The following are typical examples of positioning axes: pallet feed equipment, gauging stations or similar.

WAITP enables you to identify a position in the NC program where the program is to wait until an axis programmed with POSA in a previous NC block has reached its end position.

With WAITMO, the next NC block is loaded immediately when the wait marker is received.

Programming

`POS[axis]=…`

or

`POSA [axis]=…`

or

`POSP [axis]=(…,…,…)`

or

`FA [axis]=…`

or

`WAITP (axis)=…` (programming must be written in a separate NC block)

or

`WAITMC(marker)=…`

Parameters

POS [axis]=	Position the axis; the next NC block is not enabled until the position has been reached
POSA [axis]=	Position the axis; the next NC block is enabled, even if the position has not been reached
POSP [axis]=(,,)	Approach end position in sections. The first value indicates the end position; the second the length of the section. Approaching the end position is defined in the third value with 0 or 1
FA[axis]=	Feedrate for the positioning axis, up to 5 per NC block
WAITP (axis)	Waiting for end of travel of axis. With WAITP, an axis can be made available for traversing as a reciprocating axis or as a concurrent positioning axis (by PLC).
WAITMC (marker)	During the braking ramp, WAIITMC loads the next NC block immediately when the WAIT marker is received.
Axis	Channel axes or geometry axes
Marker, ,	An axis is only decelerated if the marker has not yet been reached or if a different search criterion prevents the block change.

Example of traveling with POSA[…]=

On accessing status data of the machine ($A...), the control generates an internal preprocessing stop, processing is halted until all previously prepared and stored blocks have been executed in full.

```
N40 POSA[X]=100
N50 IF $AA_IM[X]==R100 GOTOF LABEL1       ;Access to machine status data
N60 G0 Y100
N70 WAITP(X)
N80 LABEL1:
N...
```

Example of waiting for end of travel with WAITP(...)

Pallet feed equipment

Axis U: Pallet store, transporting the pallet to the working area

Axis V: Transfer line to a measuring station, where sampling controls are carried out:

```
N10 FA[U]=100 FA[V]=100                   ;Axis-specific feedrate functions for
                                          ;each positioning axis U and V
N20 POSA[V]=90 POSA[U]=100 G0 X50 Y70     ;Traverse positioning and path axes
N50 WAITP(U)                              ;Execution of the program is only
                                          ;continued when axis U has reached the
                                          ;end position programmed in N20.
N60 …
```

Traveling with POSA[…]=

The axis indicated in square brackets is traversed to the end position. The block step enable or program execution is not affected by POSA. The movement to the end position can be performed during execution of subsequent NC blocks.

⚠ Caution

Internal preprocessor stop

If a command, which implicitly causes a preprocessing stop, is read in a following block, this block is not executed until all other blocks, which are already preprocessed and stored have been executed. The preceding block is halted in exact stop (as with G9).

Traveling with POS[…]=

The next block is only executed when all axes programmed under POS have reached their end positions.

Traveling with POSP[...]=

POSP is deployed especially for programming oscillating motions, see

/PGA/, Programming Guide Advanced chapter "Asynchronous Oscillation".

Wait for end of travel with WAITP(...)

After WAITP, assignment of the axis to the NC program is no longer valid; this applies until the axis is programmed again. This axis can then be operated as a positioning axis through the PLC, or as a reciprocating axis from the NC program/PLC or HMI.

Block change in the braking ramp with IPOBRKA and WAITMC(...)

An axis is only decelerated if the marker has not yet been reached or if a different search criterion prevents the block change. After a WAITMC, the axes start immediately if no other search criterion prevents block change.

7.3 Position-controlled spindle operation (SPCON, SPCOF)

Function

In some cases, position-controlled operation of the spindle may be advisable, e.g., in conjunction with large-pitch thread cutting with G33, higher quality can be achieved.

Note

The command requires up to three interpolation cycles.

Programming

`SPCON or SPCON(n)` Activate position control

or

`SPCOF or SPCOF(n)` Deactivate position control, switch to speed control

or

`SPCON(n, m, 0)` Activate position control for multiple spindles in a block

or

`SPCOF(n, m, 0)` Deactivate position control for multiple spindles in a block

Parameter

SPCON SPCON(n)	Switch master spindle or spindle number n from speed control to position control
SPCOF SPCOF(n)	Switch master spindle or spindle number n back from position control to speed control
SPCON SPCON(n, m, 0)	Several spindles with number n can be switched from speed control to position control in one block
SPCOF SPCOF(n, m, 0)	Several spindles with number n can be switched back from position control to speed control in one block
n	Integers from 1 ... n of spindle number
m	Integers from 1 ... m of master spindle

Note

SPCON has modal action and remains valid until SPCOF.

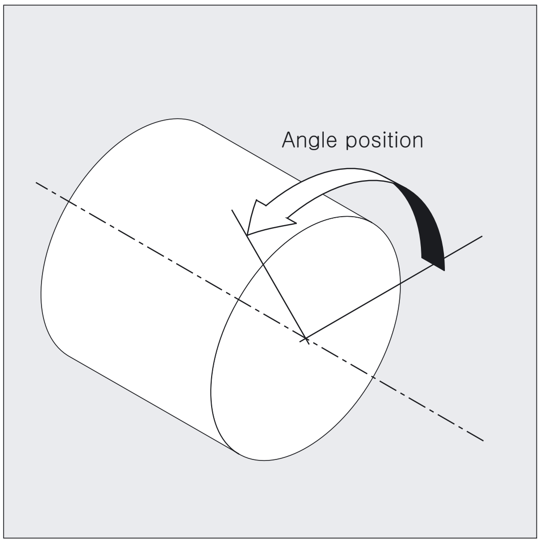
The speed is specified with S... M3, M4 and M5 apply in respect of the directions of rotation and spindle stop.

With synchronized spindle setpoint value linkage, the master spindle must be operated in position-control mode.

7.4 Positioning spindles (position-controlled axis operation) (SPOS, M19 and SPOSA)

Function

With SPOS, M19 and SPOSA, you can position spindles at specific angular positions, e.g., during tool change. In order to synchronize spindle movements, WAITS can be used to wait until the spindle position is reached.



The spindle can also be operated as a path axis, synchronized axis or positioning axis at the address defined in the machine data. When the axis identifier is specified, the spindle is in axis mode. M70 switches the spindle directly to axis mode.

Switching off

SPOS, M19 and SPOSA effect a temporary switchover to position-controlled operation until the next M3 or M4 or M5 or M41 to M45. If the position control was activated with SPCON prior to SPOS, then this remains active until SPCOF is issued.

Programming

`SPOS=… or SPOS[n]=…`

or

`M19 or M[n]=19`

or

`SPOSA=… or SPOSA[n]=…`

or

`M70` or `Mn=7`

or

`FINEA=…` or `FINEA[n]=…`

or

`COARSEA=…` or `COARSEA[n]=…`

or

`IPOENDA=…` or `IPOENDA[n]=…`

or

`IPOBRKA=…` or `IPOBRKA(axis[,REAL])` (programmed in a separate NC block)

or

`WAITS` or `WAITS(n,m)` (programmed in a separate NC block)

Parameters

SPOS= SPOS[n]=	Position master spindle (SPOS) or spindle number n (SPOS[n]); the next NC block is not enabled until the position has been reached.
M19 M[n]=19	Position master spindle (M19) or spindle number n (M[n]=19); the next NC block is not enabled until the position has been reached.
SPOSA= SPOSA[n]=	Position master spindle with SPOSA or spindle number n (SPOSA[n]). The next NC block is enabled, even if the position has not been reached.
M70 Mn=70	Switch over master spindle (M70) or spindle number n (Mn=70) to axis operation. No defined position is approached. The NC block is enabled after the switchover has been performed.
FINEA=	Motion end when "Exact stop fine" reached
FINEA[Sn]=	End of positioning for specified spindle Sn
COARSEA=	Motion end when "Exact stop coarse" reached
COARSEA[Sn]=	End of positioning for specified spindle Sn
IPOENDA=	End of motion when “IPO stop” is reached
IPOENDA[Sn]=	End of positioning for specified spindle Sn
IPOBRKA= IPOBRKA(axis[,Real])=	End of motion criterion from moment of application of braking ramp at 100% down to end of braking ramp at 0% and identical to IPOENDA.
	IPOBKRA must be programmed in round parenthesis "()".
WAITS WAITS(n,m)	Wait for spindle position to be reached, spindle stop after M5, spindle speed after M3/M4
	WAITS applies to the master spindle, WAITS(..., ...) for the specified spindle numbers
n	Integers from 1 ... n of spindle number
m	Integers from 1 ... m of master spindle
Sn	nth Spindle number, 0 to max. spindle number
Axis	Channel identifier
Real	Percentage specification 100-0% referred to the braking ramp for block changes. If no % is specified, the current value of the setting data is applied.

Specify spindle position

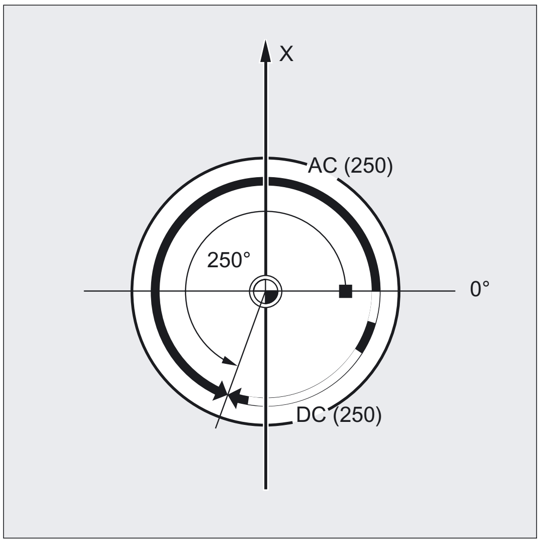
The spindle position is specified in degrees. Three spindle positions are possible for each NC block. If nothing is specified, traversing automatically takes place as for DC. With incremental dimensioning IC (INC), spindle positioning can take place over several revolutions.

AC(…)	Absolute dimension, value range AC: 0…359.9999 degrees
IC(…)	Absolute dimension, value range IC: 0…±99 999.999 degrees
DC(…)	Approach absolute value directly
ACN(…)	Absolute dimension, approach in negative direction
ACP(…)	Absolute dimension, approach in positive direction

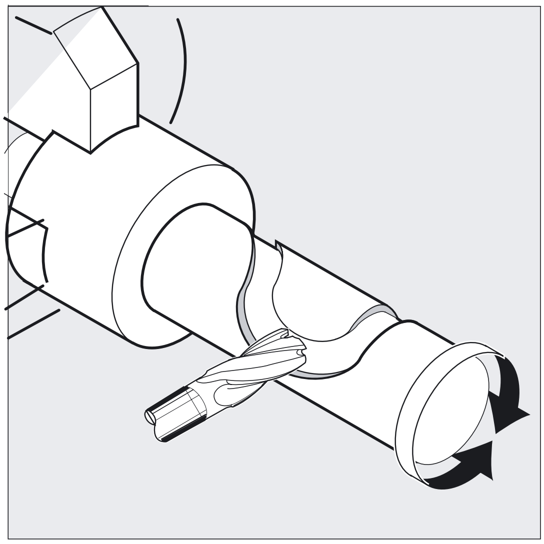
Example of positioning spindle with negative direction of rotation

Position spindle 2 at 250° in negative direction of rotation.

```
N10 SPOSA[2]=ACN(250)        ;The spindle decelerates if necessary and accelerates
                             ;in the opposite direction to the positioning
                             ;movement
```



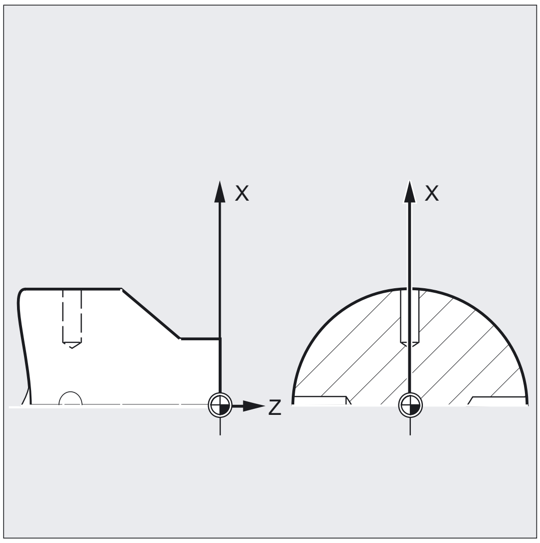
Example of positioning spindles for position-controlled axis operation



```
...
N10 M3 S500
...
N90 SPOS[2]=0 or          ;Position control on, spindle 2 positioned to 0, axis mode
                          ;can be used in the next block.
 M2=70                    ;Spindle 2 is switched to axis mode
N100 X50 C180             ;Spindle 2 (C axis) is traversed with linear interpolation
                          ;synchronous to X.
N110 Z20 SPOS[2]=90       ;Spindle 2 is positioned to 90 degrees.
```

Example of drilling cross holes in turned part

Cross holes are to be drilled in this turned part. The running drive spindle (master spindle) is stopped at zero degrees and then successively turned through 90°, stopped and so on.



```
....
N110 S2=1000 M2=3          ;Switch on cross drilling attachment
N120 SPOSA=DC(0)           ;Position main spindle directly at 0°,
                           ;the program will advance to the next block immediately
N125 G0 X34 Z-35           ;Switch on the drill while the spindle is being positioned
N130 WAITS                 ;Wait until the main spindle reaches its position
N135 G1 G94 X10 F250       ;Feedrate in mm/min (G96 is suitable only for the multi-
                           ;edge turning
                           ;tool and synchronous spindle, but not for
                           ;power tools on the cross slide)
N140G0 X34
N145 SPOS=IC(90)           ;The spindle is positioned through 90° with read halt in a
                           ;positive direction
N150 G1 X10
N155 G0 X34
N160 SPOS=AC(180)          ;The spindle is positioned at 180° with respect to the
                           ;spindle zero point
N165 G1 X10
N170 G0 X34
N175 SPOS=IC(90)           ;The spindle turns in
                           ;a positive direction through 90° from the absolute 180°
                           ;position, ending up in the
                           ;absolute 270° position.
N180 G1 X10
N185 G0 X50
...
```

Requirements

The spindle must be capable of operation in position-control mode.

Position with SPOSA=, SPOSA[n]=

The block step enable or program execution is not affected by SPOSA. The spindle positioning can be performed during execution of subsequent NC blocks. The program moves onto the next block if all the functions (except for spindle) programmed in the current block have reached their block end criterion. The spindle positioning operation may be programmed over several blocks (see WAITS).

Notice

If a command which implicitly causes a preprocessor stop is read in a subsequent block, execution of this block is delayed until all positioning spindles are stationary.

Positioning with SPOS=, SPOS[n]= and positioning with M19=, M19[n]=

The block step enabling condition is fulfilled when all functions programmed in the block have reached their block end criterion (e.g., all auxiliary functions acknowledged by the PLC, all axes have reached end point) and the spindle has reached the programmed position.

Speed of the movements

The speed or delay response for positioning is stored in the machine data and can be programmed.

Specify spindle position

As the commands G90/G91 have no effect here, the corresponding units of measurements such as AC, IC, ACN, ACP explicitly apply. If nothing is specified, traversing automatically takes place as for DC.

End of positioning

Programmable by means of the following commands: FINEA [Sn], COARSEA [Sn], IPOENDA [Sn].

Settable block change time

For single axis interpolation mode, a new end of motion can be set in addition to the existing end of motion criteria based on FINEA, COARSEA, IPOENDA.The new criterion can be set within the braking ramp (100-0%) using IPOBRKA.

The program advances to the next block if the end of motion criteria for all spindles or axes programmed in the current block plus the block change criterion for path interpolation are fulfilled. Example:

```
N10 POS[X]=100
N20 IPOBRKA(X,100)
N30 POS[X]=200
N40 POS[X]=250
N50 POS[X]=0
N60 X10 F100
N70 M30
```

Block changes if the X axis has reached position 100 and exact stop fine. Activate block change criterion IPOBRKA braking ramp. Block change commences as soon as the X axis starts to decelerate. The X axis does not brake at position 200, but moves on to position 250; as soon as the X axis starts to brake, the block changes. The X axis brakes and returns to position 0, the block is changed at position 0 and exact stop fine.

Synchronizing spindle motions WAITS, WAITS(n,m)

WAITS can be used to identify a point at which the NC program waits until one or more spindles programmed with SPOSA in a previous NC block have reached their positions.

Example: The block waits until spindles 2 and 3 have reached the positions specified in block N10.

```
N10 SPOSA[2]=180 SPOSA[3]=0
N20…N30
N40 WAITS(2,3)
```

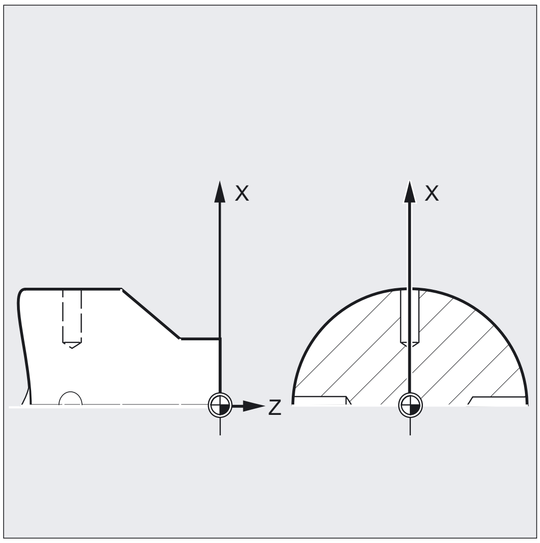
WAITS can be used after M5 to wait until the spindle(s) has(have) stopped. WAITS can be used after M3/M4 to wait until the spindle(s) has(have) reached the specified speed/direction of rotation.

Note

If the spindle has not yet been synchronized with synchronization marks, the positive direction of rotation is taken from the machine data (state on delivery).

Position spindle from rotation (M3/M4)

When M3 or M4 is active, the spindle comes to a standstill at the programmed value.



There is no difference between DC and AC dimensioning. In both cases, rotation continues in the direction selected by M3/M4 until the absolute end position is reached. With ACN and ACP, deceleration takes place if necessary, and the appropriate approach direction is followed. With IC, the spindle rotates additionally to the specified value starting at the current spindle position.

When M3 or M4 is active, the spindle decelerates if necessary, and accelerates in the programmed direction of rotation.

Position a spindle from standstill (M5)

The exact programmed distance is traversed from standstill (M5).

7.5 Feedrate for positioning axes/spindles (FA, FPR, FPRAON, FPRAOF)

Function

Positioning axes, such as workpiece transport systems, tool turrets and end supports, are traversed independently of the path and synchronized axes. A separate feedrate is therefore defined for each positioning axis. Example: FA[A1]=500

FPRAON can be used to axially activate the revolutional feedrate for positioning axes and spindles, and FPRAOF can be used to deactivate the respective axis again.

Programming

`FA [axis]=...`

`FA[SPI(spindle)]=…` or `FA[S…]=…`

or

`FPR (rotary axis)` or `FPR(SPI(spindle))` or `FPR(S…)`

or

`FPRAON (axis, rotary axis)`

or

`FPRAON(axis,SPI(spindle))` or `FPRAON(axis,S…)`

or

`FPRAON(SPI(spindle),rotary axis)` or `FPRAON(S…,rotary axis)`

or

`FPRAON(SPI(spindle),SPI(spindle))` or `FPRAON(S…,S…)`

or

`FPRAOF(axis,SPI(spindle),…)` or `FPRAOF(axis,S…,…)`

Parameters

FA[axis]	Feedrate for the specified positioning axis in mm/min or inch/min or deg/min
FA[SPI(spindle)]	Positioning velocity (axial feed)
FA[S…]	for the specified spindles in deg/min.
FPR	Identification of the rotary axis or spindle whose revolutional feedrate programmed in G95 is to be used as the basis for the revolutional feedrate of the **path and synchronized axes.**
FPRAON	Activate revolutional feedrate for **positioning axes and spindles** axially. The first command identifies the positioning axis/spindle □that is to be traversed at a revolutional feedrate. The second command identifies the rotary axis/spindle from which the feedrate must be derived.
FPRAOF	Deactivate revolutional feedrate. Specification of axis or spindle that is to stop traversing at a revolutional feedrate.
SPI	Converts the spindle number into an axis identifier; the transfer parameter must contain a valid spindle number. SPI is used for the indirect definition of a spindle number.
Axis	Positioning axes or geometry axes
Range of values	…999 999.999 mm/min, degree/min
	…39 999.9999 inch/min

Note

The programmed feedrate FA[...] is modal.

Up to 5 feeds for positioning axes or spindles can be programmed in each NC block.

Example of synchronous spindle link

With synchronous spindle link, the positioning speed of the following spindle can be programmed independently of the master spindle – for example, for positioning operations.

Example: `FA[S2]=100`

The spindle identifiers SPI(...) and S... are identical in terms of function.

Example of calculating the derived feedrate FPR

The derived feedrate is calculated according to the following formula:
Derived feedrate = programmed feedrate * Absolute master feedrate

Example: Path axes X, Y must be traversed at the revolutional feedrate derived from rotary axis A:

```
N40 FPR(A)
N50 G95 X50 Y50 F500
```

Feedrate FA[…]

The feedrate is always G94. When G70/G71 is active, the unit of measurement is metric/inches according to the default setting in the machine data. G700/G710 can be used to modify the unit of measurement in the program.

Notice

If no FA is programmed, the value defined in the machine data applies.

Feedrate FPR[...]

As an extension of the G95 command (revolutional feedrate referring to the master spindle), FPR allows the revolutional feedrate to be derived from any chosen spindle or rotary axis. G95 FPR(...) is valid for path and synchronized axes.

If the rotary axis/spindle specified in the FPR command is operating on position control, then the setpoint linkage is active. Otherwise the actual-value linkage is effective.

Feedrate FPRAON(…,…), FPRAOF(…,…)

The FPRAON command makes it possible to derive the revolutional feedrate for specific positioning axes and spindles from the current feedrate of another rotary axis or spindle.

The first command identifies the axis/spindle that must be traversed at a revolutional feedrate. The second command identifies the rotary axis/spindle that is to supply the feedrate. The command need not be specified a second time. If it is not, the feedrate is derived from the master spindle.

The revolutional feedrate can be deactivated for one or several axes/spindles simultaneously with the FPRAOF command. The feedrate is calculated in the same way as for FPR(...).

Examples: The revolutional feedrate for master spindle 1 must be derived from spindle 2.

```
N30 FPRAON(S1,S2)
N40 SPOS=150
N50 FPRAOF(S1)
```

The revolutional feedrate for positioning axis X must be derived from the master spindle. The positioning axis is traversing at 500 mm/revolution of the master spindle.

```
N30 FPRAON(X)
N40 POS[X]=50 FA[X]=500
N50 FPRAOF(S1)
```

7.6 Percentage feedrate override (OVR, OVRA)

Function

You can use the programmable feedrate override to change the velocity of path axes, positioning axes, and spindles via a command in the NC program.

Programming

```
OVR=…
```

or

```
OVRA[axis]=…
```

or

```
OVRA[SPI(spindle)]=… or OVRA[S…]=…
```

Parameter

OVR	Feedrate change in percent for path feedrate F
OVRA	Feedrate change in percent for positioning feedrate FA or for spindle speed S
SPI	Converts the spindle number into an axis identifier; the transfer parameter must contain a valid spindle number. The spindle identifiers SPI(...) and S... are identical in terms of function.
Axis	Positioning axes or geometry axes
Range of values	…200%, integers; with path and rapid traverse override, the maximum velocities set in the machine data are not overshot.

Example of programmed feed rate change

The programmed feedrate change refers to or is combined with the feedrate override set on the machine control panel.

Example:

Set feedrate override 80%

Programmed feedrate override OVR=50

The programmed path feedrate F1000 is changed to F400 (1000 * 0.8 * 0.5).

```
N10 OVR=25 OVRA[A1]=70       ;Path feedrate 25%, Positioning feedrate for A1 70%.
N20 OVRA[SPI(1)]=35          ;Speed for spindle 1 35%.
or
N20 OVRA[S1]=35
```

7.7 Feedrate with handwheel override (FD, FDA)

Function

With these functions, you can use the handwheel to traverse path and positioning axes (position parameter) or change the axis velocities (speed override) during program execution. The handwheel override is frequently used for grinding operations.

Notice

Only speed override can be used for path axes. The path feedrate F and the handwheel override FD may not be programmed in the same NC block.

Programming

`FD=…`

or

`FDA[axis]=0` or `FDA[axis]=…`

or

`FDA[axis]=...`

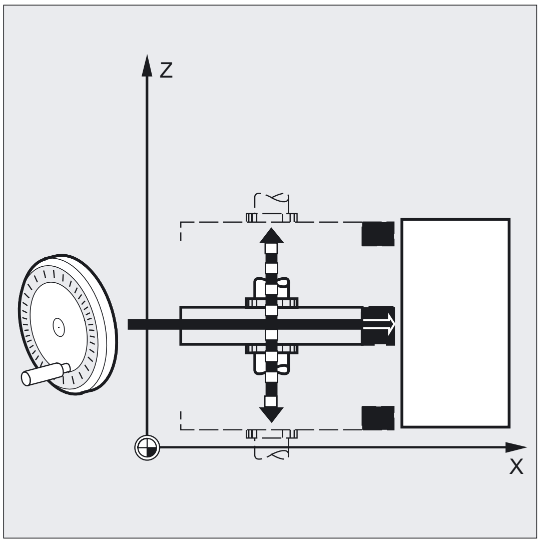
Parameters

FD=…	Handwheel travel for path axes with feedrate override
FDA[axis]=0	Handwheel travel for positioning axes according to position parameter
FDA[axis]=...	Handwheel travel for positioning axes with feedrate override
Axis	Positioning axes or geometry axes

The handwheel override function is non-modal. The function is deactivated in the next NC block and the NC program continues to be executed.

Example

Path specification: The grinding wheel oscillating in the Z direction is moved to the workpiece in the X direction using the handwheel.



The operator can then adjust the position of the tool until the spark generation is constant. When "Delete distance-to-go" is activated, the program goes to the next NC block and machining continues in NC mode.

Requirements

A handwheel must be assigned to the axes to be traversed for the handwheel override function. For the precise approach see HMI Operator's Guide. The number of handwheel pulses per graduated position is defined in machine data.

Traverse path axes with handwheel override, FD

The following preconditions apply to handwheel overrides for path axes:

In the NC block with the programmed handwheel override

- a G1, G2 or G3 motion command must be active,
- exact stop G60 must be switched on, and
- the path feedrate must be specified with G94 mm/min or inch/min.

Feed override

The feed override acts only on the programmed feed, not on the travel movement generated by the handwheel (except if feed override = 0).

Example:

```
N10 G1 X… Y… F500…
N50 X… Y… FD=700
```

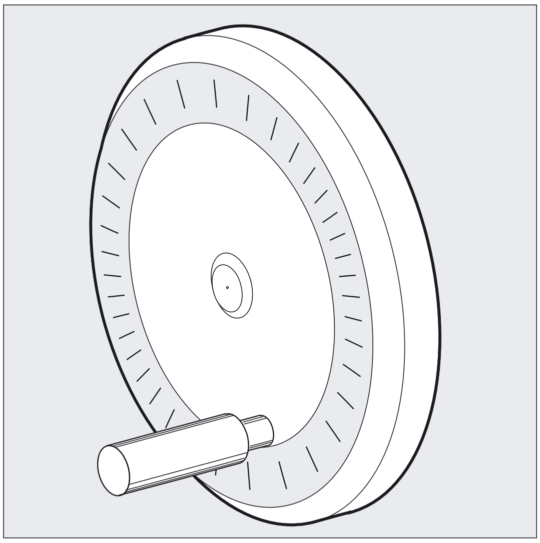
The feedrate is accelerated to 700 mm/min in block N50. The path velocity can be increased or reduced according to the direction of rotation on the handwheel.

Note

It is not possible to traverse in the opposite direction.

Handwheel travel with path default for positioning axes, FDA[axis]=0

In NC blocks with programmed FDA[axis]=0, the feed is set to zero in order that the program does not generate any travel movement. The programmed travel movement to the target position is now controlled exclusively by the operator rotating the handwheel.



Example: `N20 POS[V]=90 FDA[V]=0`

The automatic travel movement is stopped in block N20. The operator can now move the axis manually using the handwheel.

Direction of movement, travel velocity

The axes accurately follow the path set by the handwheel in the direction of the leading sign. Depending on the direction of rotation, you can travel forwards or backwards – the faster you turn the handwheel, the higher the travel velocity.

Traversing range

The traversing range is limited by the starting position and the end point programmed with the positioning command.

Handwheel travel with velocity overlay, FDA[axis]=...

In NC blocks with programmed FDA[...], the feedrate from the last programmed FA value is accelerated or decelerated to the value programmed under FDA. Starting from the current feedrate FDA, you can turn the handwheel to accelerate the programmed movement to the target position or delay it to zero. The values defined in the machine data are used for the maximum velocity.

Example:

```
N10 POS[U]=10 FDA[U]=100
POSA[V]=20 FDA[V]=150
```

Note

With velocity override of path axes, you always control the path velocity with the handwheel of the 1st geometry axis.

Traversing range

The traversing range is limited by the starting position and the programmed end point.

Manual override in automatic mode

The manual override function in automatic mode for POS/A axes has two different effects that are analogous to Jog functions.

1. Path override: FDA [ax] = 0
   The axis does not move. Handwheel impulses received for each IPO cycle are traversed directionally and path-specifically. When the target position is achieved, the axis decelerates.
2. Velocity override FDA [ax] > 0
   The axis approaches the target position at the programmed axis speed. This enables target position to be reached even without handwheel impulses. Pulses received for each IPO cycle are converted to an accumulative change in the existing velocity. Impulses in the traversing direction increase the velocity. Velocity is limited to the maximum axis velocity MAX_AX_VELO. Impulses away from the traversing direction decrease the velocity. The minimum velocity is 0.

7.8 Percentage acceleration override (ACC option)

Function

In critical program sections, it may be necessary to limit the acceleration to below the maximum values, e.g., to prevent mechanical vibrations from occurring.

You can use the programmable acceleration override to change the acceleration for each path axis or spindle via a command in the NC program. The limit is effective for all types of interpolation. The values defined in the machine data determine the 100% acceleration.

Programming

```
ACC[axis]=…
```

Or deactivate

```
ACC[axis]=100, program start, reset
```

Or

`ACC[SPI(spindle)]=…` or `ACC(S…)`

Parameters

ACC	Change in acceleration in percent for the specified path axis or change in speed for the specified spindle. Range of values: 1..200%, integers
SPI	Converts the spindle number into an axis identifier; the transfer parameter must contain a valid spindle number. The spindle identifiers SPI(...) and S... are identical in terms of function.
Axis	Channel axis name of the path axis, e.g., with X

Note

Please note that the maximum permissible values of the machine manufacturer can be exceeded with a higher acceleration rate.

Example

```
N50 ACC[X]=80
```

Meaning: Traverse the axis slide in the X direction with only 80% acceleration.

```
N60 ACC[SPI(1)]=50
```

Or

```
ACC[S1]=50
```

Meaning: Accelerate or decelerate spindle 1 with only 50% of the maximum acceleration. The spindle identifiers SPI(...) and S... are identical in terms of function.

Acceleration override programmed with ACC

The acceleration override programmed with ACC[ ] is always taken into consideration on output in system variable $AA_ACC. Readout in the parts program and in synchronized actions takes place at different times in the NC processing run.

In the part program

The value described in the parts program is only considered in the system variables $AA_ACC as described in the parts program, if ACC was not changed in the meantime by a synchronized action.

In synchronized actions

The following thus applies: the value written to the synchronized action is only considered in the system variables $AA_ACC as written to the synchronized action if ACC was not changed in the meantime by a parts program.

The defined acceleration can also be changed via synchronized actions. See /FBSY/, Synchronized Actions.

Example: `N100 EVERY $A_IN[1] DO POS[X]=50 FA[X]=2000 ACC[X]=140`
After RESET the last programmed value is retained. The current acceleration value can be called with the system variables $AA_ACC[<axis>].

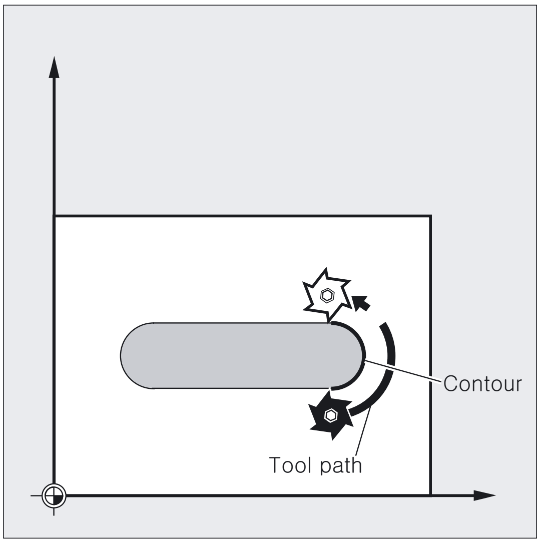
7.9 Feedrate optimization for curved path sections (CFTCP, CFC, CFIN)

Function

The programmed feedrate initially refers to the cutter center path when the G41/G42 override is activated for the cutter radius (cf. chapter "Frames").

When you mill a circle – the same applies to polynomial and spline interpolation – the extent of the change in feedrate at the cutter edge is such that it can have a considerable effect on the quality of the machined part.

Example: you are milling a small external radius with a large tool. The path that the outside of the cutter needs to cover is much longer than the path along the contour.



You therefore work with a very small feedrate on the contour. In order to avoid effects like this, you should regulate the feedrate for curved contours accordingly.

Programming

`CFTCP` Constant feedrate on cutter center-point path, deactivate feedrate override

Or

`CFC` Constant feedrate only on contour

Or

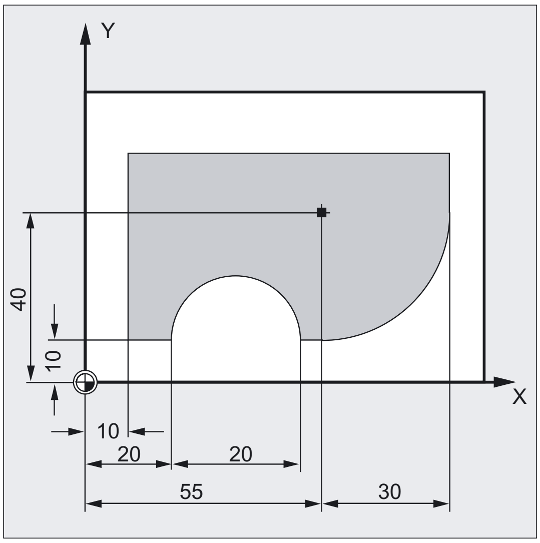
`CFIN` Constant feedrate only on inside radii, no increment at outer radii

Parameters

CFTCP	Constant feedrate on cutter center-point path. The control keeps the feedrate constant, feed overrides are deactivated.
CFC	Constant feed at contour (tool edge). This function is set as the default.
CFIN	Constant feed at tool edge for concave contours only, otherwise on the cutter center path. The feedrate velocity is reduced at inside radii.

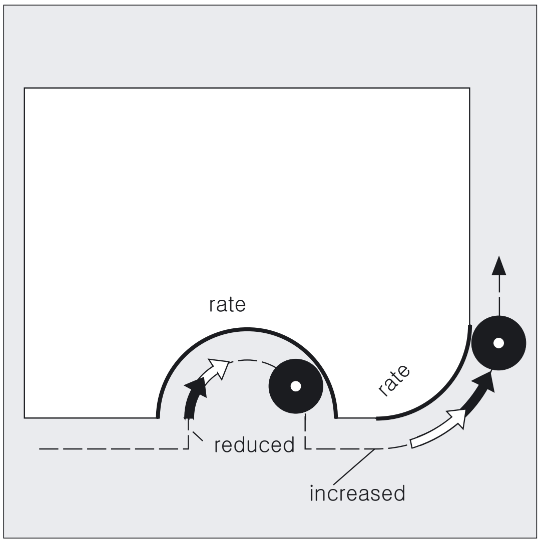
Example of milling

In this example, the contour is first machined with a CFC-compensated feed. During finishing, the cutting base is additionally machined with CFIN. This prevents the cutting base from being damaged at outside radii by too high a feedrate.



```
N10 G17 G54 G64 T1 M6
N20 S3000 M3 CFC F500 G41
N30 G0 X-10
N40 Y0 Z-10                  ;Infeed to first cutting depth
N50 CONTOUR1                 ;Subprogram call
N40 CFIN Z-25                ;Infeed to second cutting depth
N50 CONTOUR1                 ;Subprogram call
N60 Y120
N70 X200 M30
```

Constant feedrate on contour with CFC



The feedrate is reduced for inside radii and increased for outside radii. This ensures a constant speed at the tool edge and thus at the contour.

7.10 Spindle speed (S), direction of spindle rotation (M3, M4, M5)

Function

The functions described are used to

- switch the spindle on
- specify the required direction of spindle rotation, and
- define the counterspindle or an actuated tool as the master spindle, e.g., on turning machines.

The following programming commands are valid for the master spindle: G95, G96/G961, G97/G971, G33, G331 (see also Chapter "Main spindle, master spindle").

Machine manufacturer

Definition as master spindle is also possible via machine data (default).

Programming

`M3` or `M1=3`

or

`M4` or `M1=4`

or

`M5` or `M1=5`

Or

`S…`

Or

`Sn=…`

Or

`SETMS(n)` or `SETMS`

Parameters

M1=3 M1=4 M1=5	Spindle rotation clockwise/counterclockwise, spindle stop for spindle 1. Other spindles are defined according to M2=… M3=…
M3	Direction of spindle rotation clockwise for master spindle
M4	Direction of spindle rotation counterclockwise for master spindle
M5	Spindle stop for master spindle
S…	Spindle speed in rpm for the master spindle
Sn…=	Spindle speed in rpm for spindle n
SETMS(n)	Set spindle specified in n as master spindle
SETMS	Reset to the master spindle defined in machine data

Spindle speed S

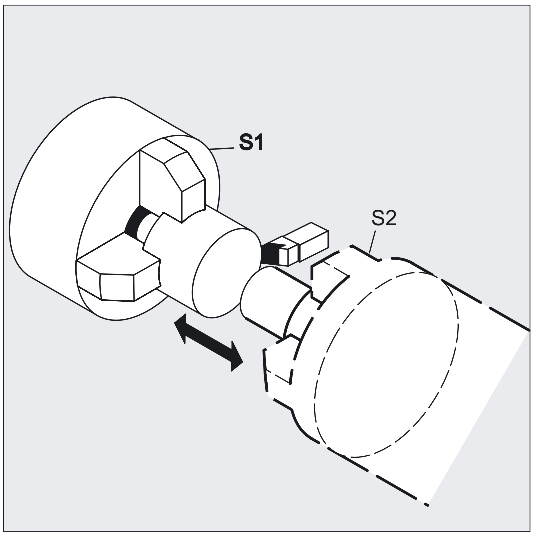
The speed specified with S… or S0=… applies to the master spindle. You specify the corresponding number for additional spindles: =…, S2=…

Note

Three S values can be programmed per NC block.

Example of master spindle with work spindle

S1 is the master spindle, S2 is the second spindle. The part is to be machined from two sides. To do this, it is necessary to divide the operations into steps. After the cut-off point, the synchronizing device (S2) takes over machining of the workpiece after the cut off. To do this, this spindle S2 is defined as the master spindle to which G95 then applies.



```
N10 S300 M3             ;Speed and direction of rotation
                        ;for drive spindle = preset master spindle
N20…N90                 ;Machining of right side of workpiece
N100 SETMS(2)           ;S2 is now master spindle
N110 S400 G95 F…        ;Speed for new master spindle
N120…N150               ;Machining of left side of workpiece
N160 SETMS              ;Switch back to master spindle S1
```

Preset M commands, M3, M4, M5

In a block with axis commands, the above mentioned functions are activated **before** the axis movements commence (basic settings on the control).

Example:

```
N10 G1 F500 X70 Y20 S270 M3          ;The spindle powers up to 270 rpm, then
                                     ;the movements are executed in
                                     ;X and Y.
N100 G0 Z150 M5                      ;Spindle stop before retraction motion in Z
```

Note

Machine data can be used to set when axis movements should be executed; either once the spindle has powered up to the setpoint speed, or immediately after the programmed switching operations have been traversed.

Working with multiple spindles

5 spindles – master spindle plus 4 additional spindles – can be available in one channel at the same time.

One of the spindles is defined in machine data as the **master spindle**. Special functions apply to this spindle, such as thread cutting, tapping, revolutional feed, dwell time.

The numbers must be specified with the speed and the direction of rotation/spindle stop for the other spindles, e.g., for a second spindle and actuated tool.

Example:

```
N10 S300 M3 S2=780 M2=4          ;Master spindle 300 rpm, U/min, clockwise,
                                 ;2nd spindle 780 rpm, counterclockwise
```

Deactivate SETMS

By issuing SETMS without spindle parameter you can switchback to the master spindle defined in the machine data.

Programmable switchover of master spindle, SETMS(n)

You can define any spindle as the master spindle with a command in the NC program.

Example:

```
N10 SETMS (2)                    ;SETMS must be located in a separate block,
                                 ;spindle 2 is now the master spindle
```

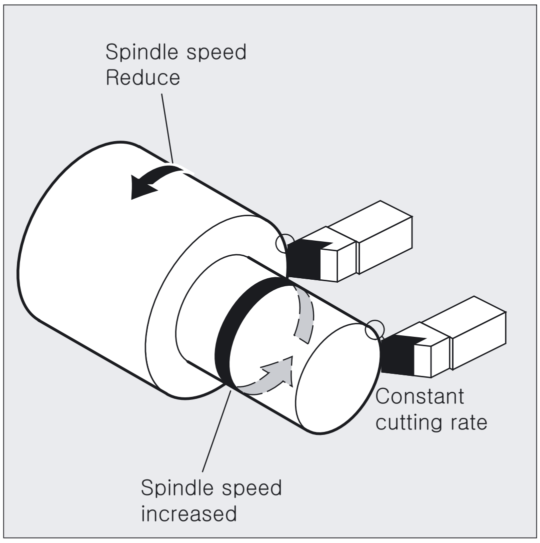
Note

The speed specified with S and M3, M4, M5 now apply.

7.11 Constant cutting rate (G96, G961, G97, G971, LIMS)

Function

When G96/G961 is active, the spindle speed – depending on the respective workpiece diameter – is modified in order that the cutting rate S in m/min or ft/min remains constant at the tool edge.



This increases the uniformity and thus the surface quality of turned parts, and also protects the tool.

The command LIMS specifies a maximum spindle speed limitation for the master spindle.

Programming

```
G96 S…
```

Or

```
G961
```

Or

```
G97
```

Or

```
G971
```

`LIMS=value or LIMS[1]=value up to LIMS[4]=value` in one block

LIMS can be expanded for machines with selectable master spindles by adding four limitations in the parts program for each of these master spindles. The speed limitation programmed with G26 or specified via setting data cannot be exceeded with LIMS and activates an alarm if not observed.

Parameters

G96	Activate constant cutting rate (as G95)
G961=	Activate constant cutting rate (as G94)
S...	Cutting rate in m/min, always applies to master spindle Range of values.
Range of values	The range of values for the cutting rate S can be between **0.1 m/min ... 9999 9999.9 m/min**. The precision can be set in the machine data. **Note:** At G70/G700: cutting rate is in feet/min.
G97	Deactivate constant cutting rate (as G95)
G971=	Deactivate constant cutting rate (as G94)
LIMS=	The speed limitation is active if G96, G961 and G97 are active for the master spindle (LIMS does not work with G971). LIMS applies to the master spindle.
LIMS[1 to 4]=…	Limitations of differing values can be programmed for up to four spindles within one block. Unless expansion is specified, LIMS will only remain effective for one master spindle.
Value	Spindle speed limitation in RPM

Example of speed limitation for the master spindle

```
N10 SETMS (3)
N20 G96 S100 LIMS=2500      ;Speed limitation at 2500 rpm
Or
N60 G96 G90 X0 Z10 F8 S100  ;Max. speed of the master spindle is 444 rpm
LIMS=444
```

Example of speed limitation for up to four spindles

Speed limitations are defined for spindle 1 (supposed master spindle) and spindles 2, 3 and 4:

```
N10 LIMS=300 LIMS[2]=450 LIMS[3]=800 LIMS[4]=1500
```

Adjust feedrate F

When G96 is active, G95 feed is automatically activated in mm/rev.

Caution

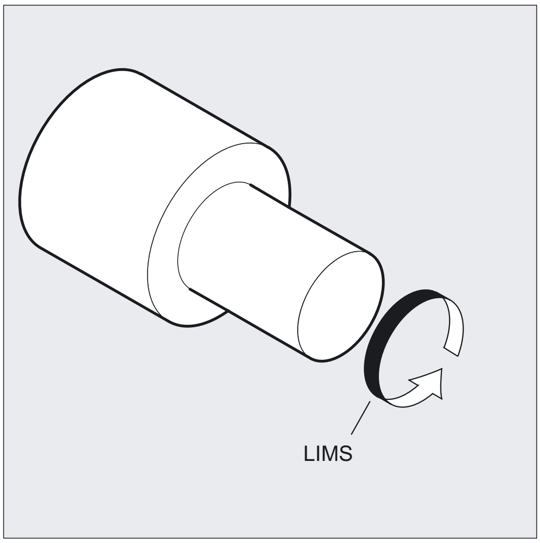
If G95 was not already active, you must specify a new feedrate F when you call G96 (e.g., convert F value from mm/min to mm/rev).

Activate constant cutting rate, G96/G961

When G96/G961 is first selected in the part program, a constant cutting rate must be entered in m/min or ft/min, when the command is reselected, a new cutting rate may be entered.

Upper speed limit LIMS

If you machine a workpiece that varies greatly in diameter, it is advisable to specify a speed limit for the spindle. This prevents excessively high speeds with small diameters. LIMS functions as a speed limiter with G96/G961 and G97.



Note

On loading the block into the main run, all programmed values are transferred into the setting data.

Deactivate constant cutting rate, G97/G971

After G97/G971 the control interprets an S word as a spindle speed in rpm again. If you do not specify a new spindle speed, the last speed set by G96/G961 is retained.

- The G96/G961 function can also be deactivated with G94 or G95. In this case, the last programmed speed S is used for further machining operations.
- G97 can be programmed without G96 beforehand. The function then has the same effect as G95; LIMS can also be programmed.
- With G961 and G971 the constant cutting rate can be activated/deactivated.

Note

The transverse axis must be defined in machine data.

Rapid traverse G0

With rapid traverse G0, there is no change in speed. Exception: if the contour is approached in rapid traverse and the next NC block contains a G1, G2, G3 … path command, the speed is adjusted in the G0 approach block for the next path command.

7.12 Constant grinding wheel peripheral speed (GWPSON, GWPSOF)

Function

With the function "Constant grinding wheel peripheral speed" (=GWPS), you can set the grinding wheel speed such that, taking account of the current radius, the grinding wheel peripheral speed remains constant.

Programming

```
GWPSON(T No.)
```

Or

```
GWPSOF(T No.)
S…
S1…
```

Parameters

GWPSON(T No.)	Select constant grinding wheel peripheral speed (GWPS) It is only necessary to specify the T number if the tool with this T number is not active.
GWPSOF(T No.)	Deselect GWPS; it is only necessary to specify the T number if the tool with this T number is not active.
S…	Program GWPS
S1…	S…: PWGS for master spindle; S1…: PWGS for spindle 1
GWPS	Peripheral speed value in m/s or ft/s The GWPS can only be selected for grinding tools (types 400-499).

Example of grinding tools with constant grinding wheel peripheral speed

A constant grinding wheel peripheral speed is to be used for grinding tools T1 and T5.

T1 is the active tool.

```
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              ;Selection of GWPS for active tool
N45 S1 = 60             ;Set GWPS for active tool to 60 m/s
…
N50 GWPSON(5)           ;GWPS selection for tool 5 (2nd spindle)
N55 S2 = 40             ;Set GWPS for spindle 2 to 40 m/s
…
N60 GWPSOF              ;Deactivate GWPS for active tool
N65 GWPSOF(5)           ;Switch off GWPS for tool 5 (spindle 2)
```

Tool-specific parameters

In order to activate the function "Constant peripheral speed", the tool-specific grinding data $TC_TPG1, $TC_TPG8 and $TC_TPG9 must be set accordingly. When the GWPS function is active, even online offset values (= wear parameters; cf. "Grinding-specific tool monitoring in the parts program TMON, TMOF" or PUTFTOC, PUTFTOCF) must be taken into account when changing speed.

Select GWPS: GWPSON, program GWPS

After selecting the GWPS with GWPSON, each subsequent S value for this spindle is interpreted as a grinding wheel peripheral speed.

Selection of grinding wheel peripheral speed with GWPSON does not cause the automatic activation of tool length compensation or tool monitoring.

The GWPS can be active for several spindles on a channel with different tool numbers.

If GWPS is to be selected for a new tool on a spindle where GWPS is already active, the active GWPS must first be deselected with GWPSOF.

Deactivate GWPS GWPSOF

When GWPS is deselected with GWPSOF, the last speed to be calculated remains valid as the setpoint.

GWPS programming is reset at the end of the parts program or on RESET.

Query active GWPS $P_GWPS[spindle no.]

This system variable can be used to query from the parts program whether the GWPS is active for a specific spindle.

TRUE: GWPS active.
FALSE: GWPS is inactive.

7.13 Programmable spindle speed limitation (G25, G26)

Function

You can use a command in the NC program to change the minimum and maximum spindle speeds defined in the machine data and setting data. It is possible to program spindle speed limitations for all spindles on the channel.

Programming

```
G25 S… S1=… S2=…
```

Or

```
G26 S… S1=… S2=…
```

A maximum of three spindle speed limitations can be programmed for each block.

Parameters

G25	Lower spindle speed limitation
G26	Upper spindle speed limitation
S S1 S2=…=…	Minimum or maximum spindle speed
Range of values	Value assignment for the spindle speed can be between **rpm ... 9999 9999.9 rpm.**

Caution

A spindle speed limitation programmed with G25 or G26 overwrites the speed limitations in the setting data and thus remains stored after the end of the program.

Examples

```
N10 G26 S1400 S2=350 S3=600        ;Upper speed limitation for master spindle,
                                   ;spindle 2 and spindle 3
```

Maximum possible spindle speed limitations within a block

```
LIMS[1]=500 LIMS[2]=600            ;Master spindle speed limitations
LIMS[3]=700 LIMS[3]=800            ;Maximum for 4 spindles within a block
G25 S1=1 S2=2 S3=3                 ;Lower and upper speed limit
G26 S1=1000 S2=2000 S3=3000        ;Maximum of 3 spindle limitations within a block
```

7.14 Multiple feedrate values in one block (F.., ST=.., SR=.., FMA.., STA=.., SRA=..)

Function

The "Several feedrates in one block" function can be used independent of external analog and/or digital inputs to activate

- Different feedrates of an NC block,
- dwell time, and
- Retraction

in synchronism with the movement.

The HW inputs signals are combined in one input byte, see description in: /FB1/, Description of Functions, A2, Various Interface Signals.

Programming

`F2=` to `F7=` Multiple path motions in 1 block

`ST=`

`SR=`

Or

`FMA[2,x]=` to `FMA[7,x]`=Multiple axial motions in 1 block

`STA=`

`SRA=`

Parameters

`F2=... to F7=...==`	In addition to the path feed, you can program up to 6 further feedrates in the block; non modal
`ST=...`	Dwell time (for grinding technology: sparking-out time); non modal
`SR=...`	Return path; non modal. The unit for the retraction path refers to the current valid unit of measurement (mm or inch).

`FMA[2,x]=... to`
`FMA[7,x]=...`
	In addition to the path feed, you can program up to 6 further feedrates per axis in the block; non modal

`STA=...`	Axial dwell time (for grinding technology: sparking-out time); non modal
`SRA=...`	Axial return path; non modal

FA, FMA and F value

The axial feedrate (FA or FMA value) or path feedrate (F value) corresponds to 100% feedrate. You can use this function to realize feedrates that are smaller than or equal to the axial feedrate or the path feedrate.

References

/FB1/Description of Functions, V1, Feedrates

Note

If feedrates, dwell time or return path are programmed for an axis on account of an external input, this axis must in this block must not be programmed as POSA axis (positioning axis over multiple blocks).

Look Ahead is also active for multiple feedrates in one block. In this way, the current feedrate is restricted by the Look Ahead value.

Example of programming path motion

The path feed is programmed under the address F and remains valid until an input signal is present. The numerical expansion indicates the bit number of the input that activates the feedrate when changed:

```
F7=1000                     ;7 corresponds to input bit 7
F2=20                       ;2 corresponds to input bit 2
ST=1                        ;Dwell time (s) input bit 1
SR=0.5                      ;Return path (mm) input bit 0
```

Example of programming axial motion

The axial path feed is programmed under the address FA and remains valid until an input signal is present.

FMA[7,x]= to FMA[2,x]= can be used to program up to 6 further feeds per axis in the block. The first expression in the square brackets indicates the bit number of the input; the second the axis for which the feedrate is to apply

```
FMA[3,y]=1000               ;Axial feedrate with the value 1000 for Y axis, 3
                            ;corresponds to input bit 3
```

Example of axial dwell time and return path

Dwell time and return path are programmed under the following additional addresses:

```
STA[x]=...                  ;Axial dwell time (s) input bit 1
SRA[x]=...                  ;Axial return path (mm) input bit 0
```

If input bit 1 is activated for the dwell time or bit 0 for the return path, the distance to go for the path axes or the relevant single axes is deleted and the dwell time or return started.

Example of several operations in one block

```
N20 T1 D1 F500 G0 X100      ;Initial setting
N25 G1 X105 F=20 F7=5       ;Normal feedrate with F, roughing with F7, finishing
F3=2.5 F2=0.5 ST=1.5 SR=    ;with
0.5                         ;F3, smooth-finishing with F2, dwell time 1.5 s
                            ;return path 0.5 mm
N30 ...
…
```

7.15 Blockwise feed (FB...)

Function

You can use the function "Blockwise feedrate" to define a separate feedrate for a single block.

The address FB is used to define the feedrate only for the current block. After this block, the previously active modal feedrate is active.

The feedrate is interpreted according to the active feedrate type:

G94: feedrate in mm/min or °/min

G95: feedrate in mm/rev or inch/rev

G96: constant cutting rate

References: Description of Functions /FB1/, V1, Feedrates

Programming

FB=<Wert> Feed motion only in one block

Parameters

`FB=...=`	`Instead of the modal feedrate active in the previous block, you can program a separate feedrate for this block; in the block that follows, the previously active modal feedrate applies.`
`<WERT>`	`The programmed value of FB=<Wert> must be greater than zero.`

Feed value

The address FB is used to define the feedrate only for the current block. After this block, the previously active modal feedrate is active.

The feedrate is interpreted according to the active feedrate type:

- G94: feedrate in mm/min or degrees/min
- G95: feedrate in mm/rev or inch/rev
- G96: constant cutting rate

References: /FB1/ Description of Functions, V1, Feedrates

Further information:

If no traversing motion is programmed in the block (e.g. computation block), the FB has no effect.

If no explicit feed for chamfering/rounding is programmed, then the value of FB also applies for any contour element chamfering/rounding in this block.

Feedrate interpolations FLIN, FCUB, etc. are also possible without restriction.

Simultaneous programming of FB and FD (handwheel travel with feed overlay) or F (modal path feedrate) is not possible.

Examples

```
N10 G0 X0 Y0 G17 F100     ;Initial setting
G94
N20 G1 X10                ;Feedrate 100 mm/min
N30 X20 FB=80             ;Feedrate 80 mm/min
N40 X30                   ;Feedrate is 100 mm/min again
N50 ...
…
```

Tool offsets 8

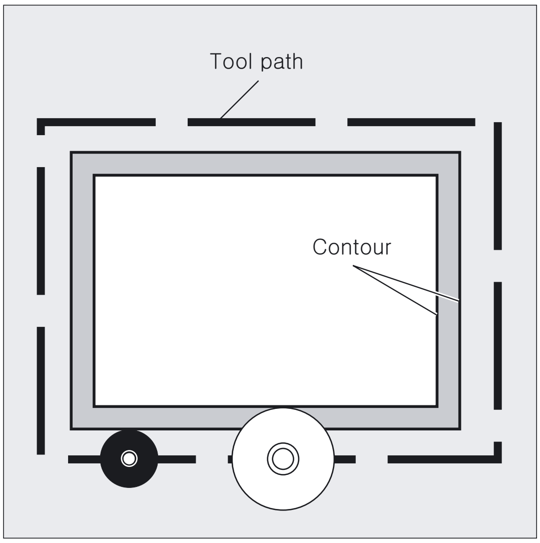
8.1 General notes

8.1.1 Tool offsets

When writing a program, it is not necessary to specify the cutter diameter, the tool point direction of the turning tool (left/right-handed turning tools) or tool length.

You program the workpiece dimensions directly, for example, following the production drawing.

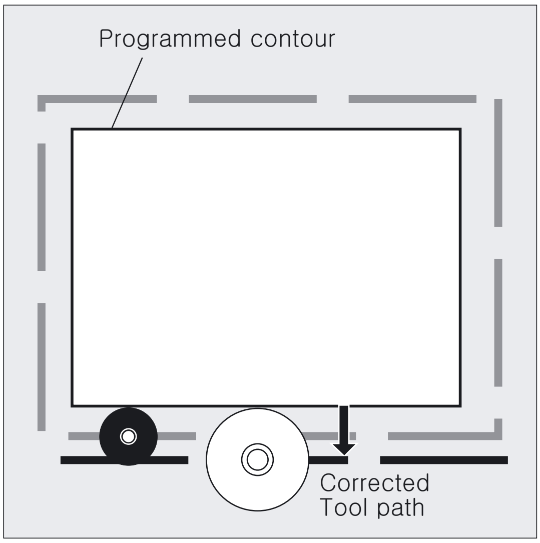
When machining a workpiece, the tool paths are controlled according to the tool geometry such that the programmed contour can be machined using any tool.



The control corrects the traverse path

You enter the tool data separately in the tool table on the control.

All you need to do is call the required tool with its offset data in the program.



During program execution, the control fetches the offset data from the tool files and corrects the tool path individually for different tools.

Enter tool offsets into the offset memory

In the offset memory enter the following:

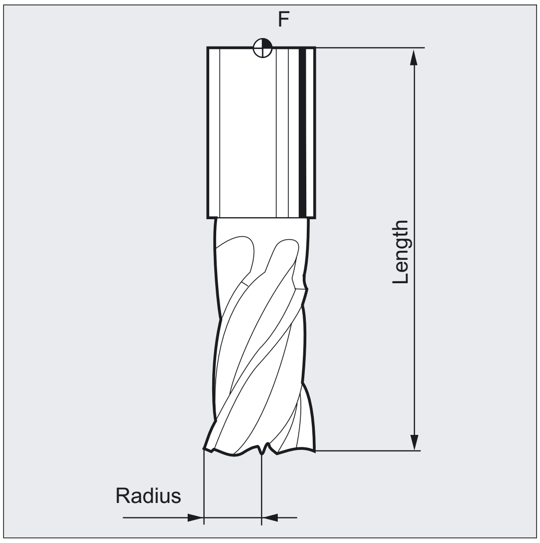
- Geometric dimensions based on wear: length, radius.
- Tool type with the tool parameters for drill, milling tool and grinding, turning or special tools
- Length of cutting edge

8.1.2 Tool offsets in the control's offset memory

Which tool offsets are stored in the control's offset memory?

In the offset memory enter the following:

- Geometrical dimensions: length, radius.



They consist of several components (geometry, wear). The control computes the components to a certain dimension (e.g., overall length 1, total radius). The respective overall dimension becomes active when the offset memory is activated.
The way in which these values are computed in the axes is determined by the tool type and the current plane G17, G18, G19.

- Tool type

The type determines which geometry data are needed and how they are calculated (drill or milling tool or turning tool).

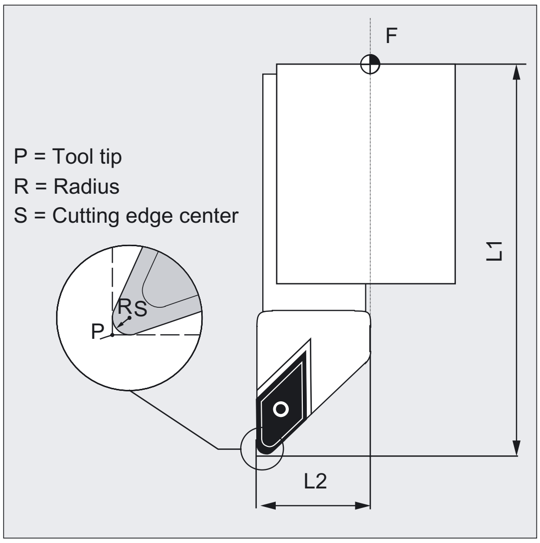
- Cutting-edge position

Tool parameters

The following section "List of tool types" describes the individual tool parameters on the display. The relevant tool parameters must be entered in the entry fields with "DP...".

Notice

Values that have been entered once in the offset memory are included in the processing for each tool called.



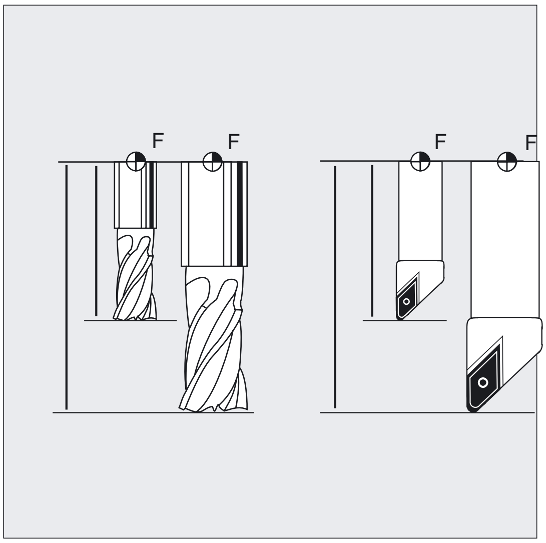
Any tool parameters that are not required must be set to "zero".

Description

Tool length compensation

This value compensates for the differences in length between the tools used.

The tool length is the distance between the toolholder reference point and the tip of the tool.



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

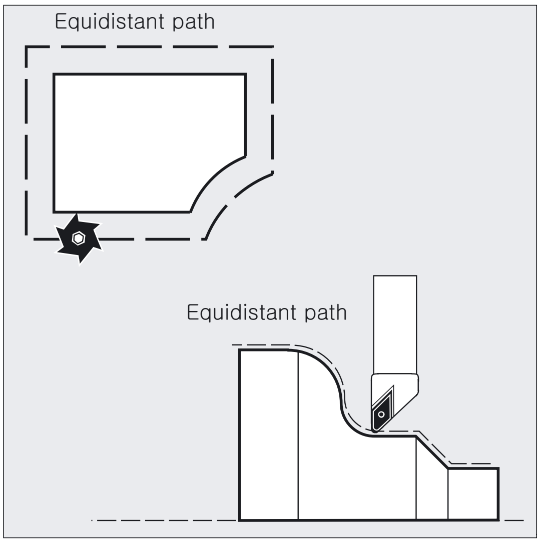
Notex

The compensation value of the tool length depends on the spatial orientation of the tool. See also chapter "Tool orientation and tool length compensation" for more information.

Tool radius compensation

The contour and tool path are not identical. The cutter or tool nose radius center must travel along a path that is equidistant from the contour. To do this, the programmed tool center point path is displaced by an amount that depends on the radius and the direction of machining and such that the tool nose travels exactly along the desired contour.

The control fetches the required radii during program execution and calculates the tool path from these values.



Notice

The tool radius compensation acts according to default setting CUT2D or CUT2DF. You will find more information later in this chapter.

8.2 List of tool types

Codings of tool types

The individually coded tool types are divided up into the following groups depending on the technology used:

1. Group with type 1xy milling tools
2. Group with type 2xy drills
3. Group with type 3xy reserved
4. Group with type 4xy grinding tools
5. Group with type 5xy turning tools
6. Group with type 6xy reserved
7. Group with type 7xy special tools such as slotting saw

Codings of tool types for milling tools

Group with type 1xy (milling tool):

100 Milling tool according to CLDATA

110 Ball end mill (cylindrical die sinker)

111 Ball end mill (tapered die sinker)

120 End mill (without corner rounding)

121 End mill (with corner rounding)

130 Angle head cutter (without corner rounding)

131 Angle head cutter (with corner rounding)

140 Face milling

145 Thread cutter

150 Side mill

151 Saw

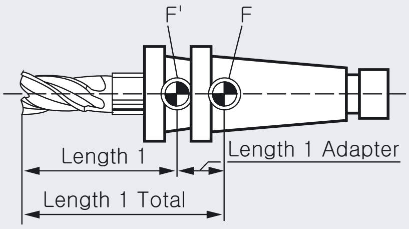
155 Bevel cutter (without corner rounding)

156 Bevel cutter (with corner rounding)

157 Conical die milling cutter

160 Drill and thread milling cutter

Entries in Tool parameters	
DP1	1xy
DP3	Geometry length 1
DP6	Geometry radius
DP21	Adapter length



F–Reference point adapter (with tool inserted=toolholder reference point)

Wear values as required

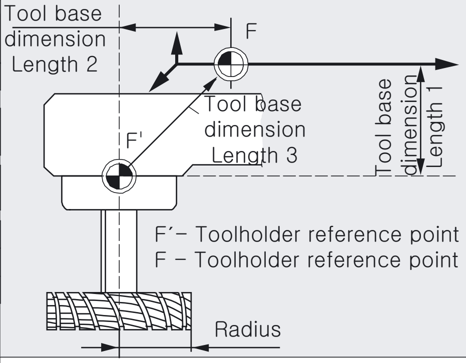
Set remaining values to 0

Effect	
G17:	Length 1 in Z Radius in X/Y
G18:	Length 1 in Y Radius in Z/X
G19:	Length 25.40 mm X Radius in Y/Z

F´–Toolholder Reference point

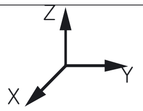
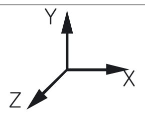
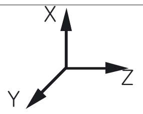
A fixed assignment is possible for G17, G18, G19
e.g., length1=X, length2=Z, length3=Y (see /FB1/W1 Tool compens.)

Entries in Tool parameters	
DP1	1xy
DP3	Geometry length 1
DP6	Geometry radius
DP21	Base length 1
DP22	Base length 2
DP23	Base length 3


Tool base dimension Length 2
F
Tool base dimension Length 3
Tool base dimension Length 1
F'
F´– Toolholder reference point
F – Toolholder reference point
Radius

Wear values acc. to requirements

Set remaining values to 0

Effect		
G17:	Length 25.40 mm Z Length 5.08 cm Y Length 3 in X Radius/TRC in X/Y	
G18:	Length 1 in Y Length 2 in X Length 3 in Z Radius/TRC in Z/X	
G19:	Length 25.40 mm X Length 2 in Z Length 3 in Y Radius/TRC in Y/Z	

A fixed assignment is possible for G17, G18, G19
e.g., length 1=X, length 2=Y, length 3=Z (see /FB1/W1 Tool compensation)

Coding of tool types for drills

Group type 2xy (drills):

200 Twist drill

205 Drill

210 Boring bar

220 Center drill

230 Countersink

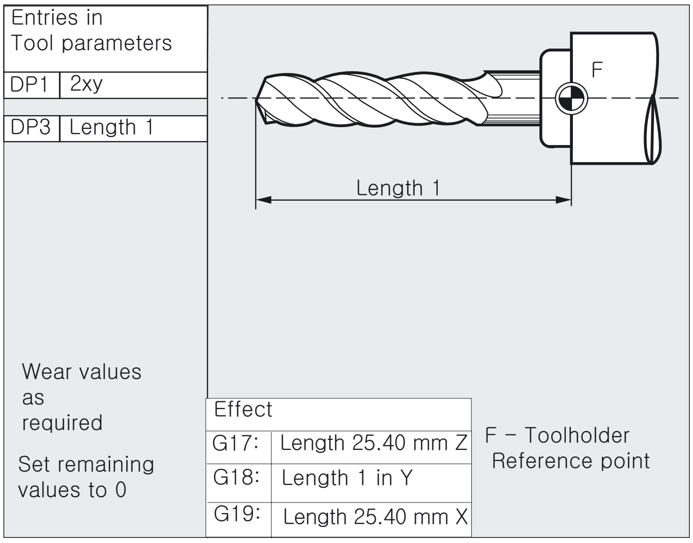
231 Counterbore

240 Regular thread tap

241 Fine thread tap

242 Whitworth-thread tap

250 Reamer



Entries in Tool parameters	
DP1	2xy
DP3	Length 1

Wear values as required

Set remaining values to 0

Effect	
G17:	Length 25.40 mm Z
G18:	Length 1 in Y
G19:	Length 25.40 mm X

F – Toolholder Reference point

Coding of tool types for grinding tools

Group type 4xy (grinding tools):

400 Surface grinding wheel

401 Surface grinding wheel with monitoring

402 Surface grinding wheel without monitoring without toolbase dimension (TOOLMAN)

403 Surface grinding wheel with monitoring/without tool base dimension
for grinding wheel surface speed (GWPS)

410 Facing wheel

411 Facing wheel (TOOLMAN) with monitoring

412 Facing wheel (TOOLMAN) without monitoring

413 Facing wheel with monitoring/without tool base dimension for
grinding wheel surface speed (GWPS)

490 Dresser

Entries in tool parameters	
DP1	403
DP2	Position *)
DP3	Length 1
DP4	Length 2
DP6	Radius

*) Tool nose position

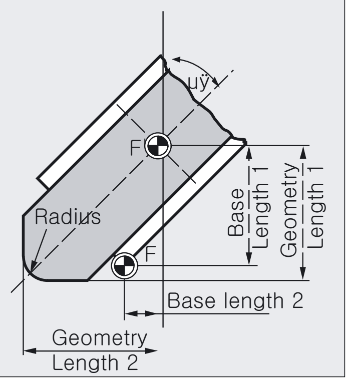
Wear values as required

Set remaining values to 0

TPG1	Spindle number
TPG2	Chaining rule
TPG3	Minimum wheel radius
TPG4	Minimum wheel width
TPG5	Current wheel width
TPG6	Maximum speed
TPG7	Max. peripheral speed
TPG8	Angle of inclined wheel
TPG9	Parameter no. for radius calculation

Effect	
G17:	Length 1 in Y Length 2 in X Radius in X/Y
G18:	Length 25.40 mm X Length 2 in Z Radius in Z/X
G19:	Length 25.40 mm Z Length 5.08 cm Y Radius in Y/Z

F – Toolholder reference point



Coding of tool types for turning tools

Group type 5xy (turning tools):

500 Roughing tool

510 Finishing tool

520 Plunge cutter

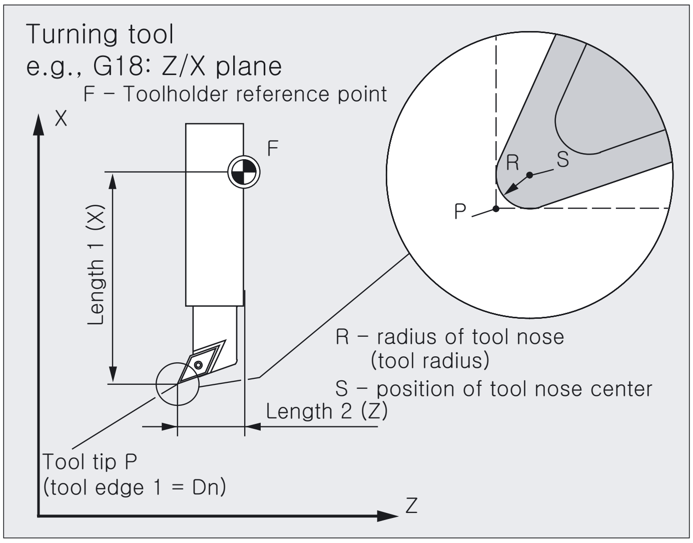
530 Parting tool

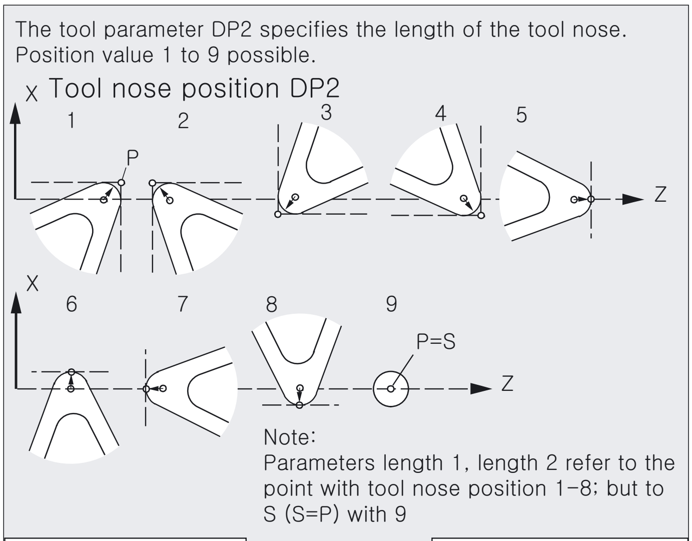
540 Threading tool

550 Mushroom tool/form tool (TOOLMAN)

560 Rotary drill (ECOCUT)

580 Probe with cutting edge position parameter




The tool parameter DP2 specifies the length of the tool nose.
Position value 1 to 9 possible.

Note:
Parameters length 1, length 2 refer to the point with tool nose position 1–8; but to S (S=P) with 9

Entries in Tool parameters	
DP1	5xy
DP2	1...9
DP3	Length 1
DP4	Length 2
DP6	Radius

Wear values as required

Set remaining values to 0

Effect	
G17:	Length 1 in Y Length 2 in X
G18:	Length 25.40 mm X Length 2 in Z
G19:	Length 25.40 mm Z Length 5.08 cm Y

Chaining rule

The tool length offsets

- Geometry,

- Wear and
- Tool base dimension

can be chained for the left and right wheel correction in each case, i.e., if the length offsets for the left tool edge are altered, the values for the right edge are automatically entered and vice versa. Please refer to
/FB 2/Description of Functions, W4 "Grinding".

Coding of tool types for special tools

Group type 7xy (special tools):

700 Slotting saw

710 3D probe

711 Edge probe

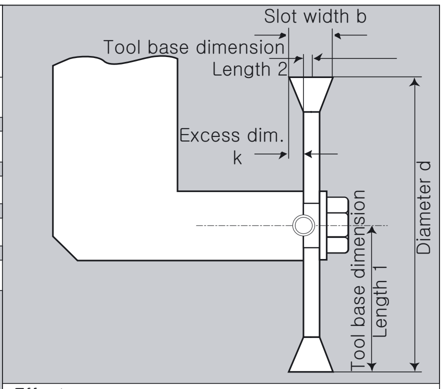
730 Stop

Slotting saw

Group with type:

700 Slotting saw

Entries in Tool parameters	
DP3	Base length 1
DP4	Base length 2
DP6	Geometry diameter
DP7	Geometry zero width
DP8	Geometry overshoot
Wear values as required Set remaining values to 0	



Effect		
G17	Half diameter (L1) in X Excess dim. in (L2) Y Saw blade in (R) X/Y	Plane selection 1st – 2nd axis (X–Y)
G18	Half diameter (L1) in Y Excess dim. in (L2) X Saw blade in (R) Z/X	Plane selection 1st – 2nd axis (X–Z)
G19	Half diameter (L1) in Z Excess dim. in (L2) Z Saw blade in (R) Y/Z	Plane selection 1st – 2nd axis (Y–Z)

Note

The parameters for the tool types are described in:

References: /FB1/Description of Functions W1 "Tool offsets" and in help images of control

8.3 Tool selection/tool call T

8.3.1 Tool change with T commands (turning)

Function

A direct tool change takes place when the T word is programmed.

Tool selection without tool management

Free selection of D No. (flat D No.) relative to cutting edges

Tabulated D No.: D1 ... D8

Tool selection with tool management

Free selection of D No. (flat D No.) relative to cutting edges

Fixed assignment of D No. to the cutting edges

Programming

`Tx` or `T=x` or `Ty=X`

Or

`T0=`

Parameters

`Tx or T=x or Ty=x`	`Tool selection with T No. including tool change (active tool), tool offset is active`
`x`	`x stands for T No.: 0-32000`
`T0=`	`Tool deselection`
	`Number of tools: 1200`
	`(depending on the machine manufacturer's configuration)`

Machine manufacturer

The effect of the T number call is defined in machine data. See machine manufacturer's configuration.

Important

It is important to ensure that the machine data for "Error response for programmed tool change" is expanded by bit 7.

- With the currently valid default setting, a check is performed **immediately** to ascertain whether the NC kernel is aware of the tool number. If not, an alarm is triggered immediately.

- A response that deviates from this (default setting for previous software versions) can be brought into line with bit 7.
  The programmed tool number will only be checked following D-selection. If the NC kernel is unaware of the tool number, an alarm is set during D-selection. This response is desirable if, for example, tool programming is also intended to achieve positioning and the tool data is not necessarily available (circular magazine).

8.3.2 Tool change with M06 (mill)

Function

Tool selection takes place when the T word is programmed.

1. Tool selection without tool management

- Free selection of D No. (flat D No.) relative to cutting edges

T...	[8-digit]

D1	D2	D3	• • •	D32000

- Tabulated D No.: D1 ... D8

T1	D	D	D	• • •	D
T2	D				
T3	D				
T6	D	D	D		
T9	D	D			
	D		D		
•					
•					
T...	D	D			

2. Tool selection with tool management

- Free selection of D No. (flat D No.) relative to cutting edges
- Fixed assignment of D No. to the cutting edges

The tool only becomes active with M06 (incl. corresponding D No.).

Programming

`Tx` or `T=x` or `Ty=X`

or

`T0=`

or

`M06F2=...` to `F7=...`

Parameter

`Tx or T=x or Ty=x`	Tool selection with T no.
`x`	x stands for T no.: 0-32000
`T0=`	Tool deselection
`M06`	Tool change, then tool T... and tool offset D are active Number of tools: 1200 (depending on the machine manufacturer's configuration)

Machine manufacturer

The effect of the T number call is defined in machine data. Please refer to the machine manufacturer's configuration.

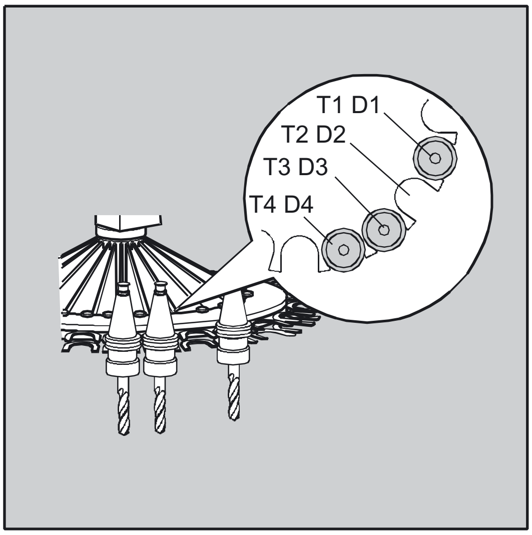
Explanation

The free selection of the D No., "Flat D numbers", is used when tool management is done outside the NC. In this case, the D numbers are created with the corresponding tool compensation blocks without assignment to tools.

T can continue to be programmed in the parts program. However, this T has no reference to the programmed D number.

Example:

Circular magazine with 12 locations and 12 single-edge tools.



Machine manufacturer

T can or cannot be programmed in the parts program, depending on the setting in MD 18102.

Creating a new D number

Creating a new D number with the associated tool compensation blocks is performed exactly as for the normal D number via tool parameters $TC_DP1 to $TC_DP25. The T number need not be entered any more.

Machine manufacturer

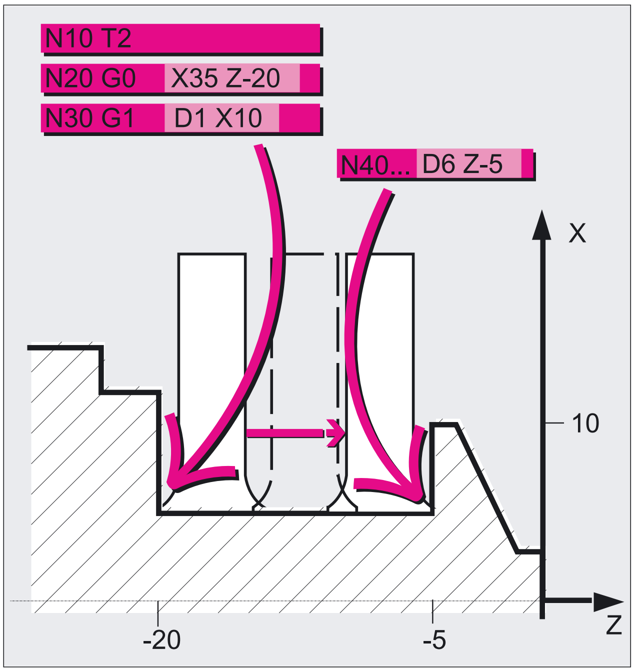
The type of D number management is defined in the machine data. There are two settings available for programming D numbers in the "flat D number structure":

- Flat D number structure with direct programming
- Flat D number structure with indirect programming

8.4 Tool offset D

Function

It is possible to assign between 1 and 8 (12) tool noses per tool with different tool offset blocks to a specific tool. This allows you to define various tool noses for **one** tool, which you can call as required in the NC program. Different offset values could be used, for example, for the left and right tool nose of a grooving tool.



When D is called, the tool length offset for a specific tool nose is activated. If D0 is programmed, the offsets for the tool are ineffective.

Tool length offsets take immediate effect when the D number is programmed. If no D word is programmed, the default setting from the machine data is valid for tool change. A tool radius offset must also be activated by G41/G42.

Programming

```
D...
```

or

```
D0=
```

Parameter

Dx	Tool offset number: Without WZV 1... 8 or with WZV 1...12
x	x stands for the D No.: 0-32000
D0=	Tool offset deselection, no offsets active. D0 is preset by default after control is powered up.

Note

If you do not enter a D number, you will be working without a tool offset.

Machine manufacturer

Machine manufacturer default setting (e.g., D1, which means no D programming) is activated/selected by means of tool change (M06) D1. The tools are activated with T programming (see machine manufacturer's specifications).

The offset is applied with the first programmed traverse of the respective length offset axis.

Caution

The modified values only become active the next time the T or D number is programmed.

The required D number must always be programmed before the tool length offset can be selected. The tool length offset is also effective if set in the machine data.

Example of turning

Tool change with T command

```
N10 T1 D1                         ;Tool T1 is changed and activated with associated
                                  ;D1
N11 G0 X... Z...                  ;The length offsets are traversed
N50 T4 D2                         ;Load tool T4, D2 from T4 is active
...
N70 G0 Z... D1                    ;Other cutting edge D1 is activated for tool T4
```

8.5 Tool selection T with tool management

Function

Tool selection T with tool management is illustrated in the sample magazine with 1 to 20 locations.

Initial conditions when calling the tool

Note

When calling the tool:

1. The tool offset values stored under a D number must be activated.

2. The appropriate working plane (system setting: G18) must be programmed. This ensures that the length compensation is assigned to the correct axis.

Machine manufacturer

Tool management: See machine manufacturer's configuration.

Important

It is important to ensure that the machine data for "Error response for programmed tool change" is expanded by bit 7.

Tool magazine

If the selected magazine location in a tool magazine is not occupied, the tool command has the same effect as T0. Selecting the empty magazine location can be used to position the empty location.

Sample magazine with 1 to 20 locations

A magazine has locations 1 to 20:

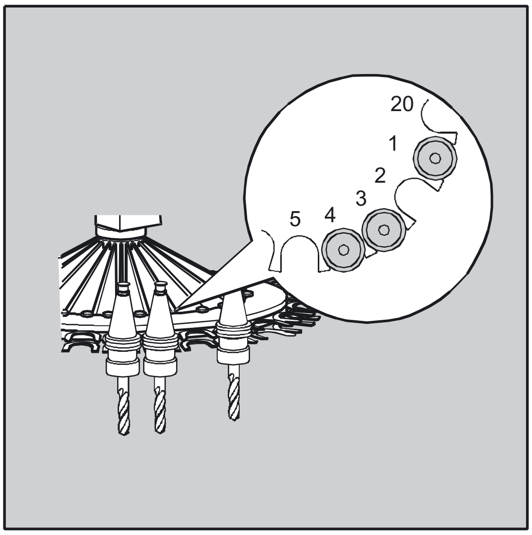
Location 1 occupied by drilling tool, duplo no.=1, T15, disabled

Location 2 is not occupied

Location 3 is occupied by drilling tool, duplo no.=2, T10, enabled

Location 4 is occupied by drilling tool, duplo no.=3, T1, active

Locations 5 to 20 are not occupied



Programming N10 T1 or T=1:

1. Magazine location 1 is scrutinized and the tool identifier is ascertained.
2. This tool is locked and therefore cannot be used.
3. A tool search for T="drill" is initiated in accordance with the search method set.
   Exception: "Find the active tool; or else, select the one with the next highest duplo number".
4. "Drill" tool with duplo number 3 (in magazine location 4) is identified as the tool that can be used.
   This completes the tool selection process and the tool change is initiated:
5. If the "Select the first available tool from the group" search method is employed, the sequence must first be defined within the tool group being loaded.
   Group T10 is loaded, as T15 is locked.
6. On completion of tool search strategy "Take the first tool with status "active" from group", T1 is loaded.

8.5.1 Turning machine with circular magazine (T selection)

Function

The tools must be assigned unique names and numbers for identification purposes. Below it will be demonstrated how to uniquely define the parameters for the tool management option on a turning machine with circular magazine.

Machine manufacturer

Tool management: See machine manufacturer's configuration.

Programming

The following sequence usually applies:

`T =` location

Or

`T =` identifier

`D...` Tool offset number: 1...32000 (max., see machine manufacturer's specifications)

Parameters

T = location or identifier	Location or identifier, T triggers the tool change. Extended address, tool for spindle 2
T2 = identifier	
T0	Magazine location not occupied
D = offset	1 to n (n ≤ 32000) If the relative D No. structure with internal reference to the associated tools is used, replacement tool management and monitoring function are possible.
D0	no offsets active!

8.5.2 Milling machine with chain magazine (T selection)

Function

The tools must be assigned unique names and numbers for identification purposes. Below it will be demonstrated how to uniquely define the parameters for the tool management option on a milling machine with chain magazine.

Machine manufacturer

Tool management: See machine manufacturer's configuration.

Programming

The following sequence usually applies:

`T` = identifier or `T` = number	
	M06 triggers the tool change
`D` = offset	Tool edge number 1 to n (n ≤ 12)

Tool selection

With integrated tool management (within the NC)

Relative D no. structure	**With** internal reference made to the associated tools (e.g., tool management and monitoring function)

Without integrated tool management (external to NC)

Flat D no. structure	**Without** internal reference made to the associated tools

Selection

- With integrated tool management (inside NC)
  relative D no. structure **with** internal reference to the associated tools
  (e.g., replacement tool management and monitoring function possible)
- Without integrated tool management (outside NC)
  flat D no. structure **without** internal reference to associated tools.

Note

When calling the tool:

1. The tool offset values stored under a D number must be activated.

2. The appropriate working plane (system setting: G17) must be programmed. This ensures that the length compensation is assigned to the correct axis.

If the selected magazine location in a tool magazine is not occupied, the tool command has the same effect as T0. Selecting the empty magazine location can be used to position the empty location.

Machine manufacturer

Tool management: See machine manufacturer's configuration.

Tool magazine

If the selected magazine location in a tool magazine is not occupied, the tool command has the same effect as T0. Selecting the empty magazine location can be used to position the empty location.

8.6 Tool offset call D with tool management

8.6.1 Turning machine with circular magazine (D call)

Programming

The following programming sequence usually applies:

`T` = location

Or

`T` = identifier T triggers the tool change

`D...` Tool offset number: 1...32000 (max., see machine manufacturer's specifications)
`D0:` no offsets active!

Direct, absolute programming

Programming is performed with the D number structure. The compensation blocks to be used are called directly via their D number.

Assignment of the D number to a specific tool does not take place in the NC kernel.

Machine manufacturer

Direct programming is defined by machine data.

Example of turning machine with circular magazine

```
$MC_TOOL_CHANGE_MODE=0          ;MD20270 CUTTING_EDGE_DEFAULT = 1
...
D92                             ;Traverse with tool offsets from D92
...
T17                             ;Select T17, traverse with tool offsets from D92
...
D16                             ;Traverse with tool offsets from D16
...
D32000                          ;Traverse with tool offsets from D32000
...
T29000500                       ;Select T29000500, traverse with tool offsets from
                                ;D32000
...
D1                              ;Traverse with tool offsets from D1
```

8.6.2 Milling machine with chain magazine (D call)

Function

It is possible to assign between 1 and 12 tool noses with different tool compensation blocks to a specific tool. When D is called, the tool length compensation for a specific tool nose is activated. When D0 is programmed, offsets for the tool are ineffective. If no D word is programmed, the default setting from the machine data is valid for tool change.

Tool length compensations take immediate effect when the D number is programmed.

A tool radius offset must also be activated by G41/G42.

Machine manufacturer

Tool management: See machine manufacturer's specifications.

Programming

The following sequence usually applies:

`T` = "Ident" or `T` = number or `T` = duplo number	
	M06 triggers the tool change
`D` = offset	Tool edge number 1 to n (n ≤ 12)

Tool selection

With integrated tool management (within the NC)

Relative D no. structure	**With** internal reference made to the associated tools (e.g., tool management and monitoring function)

Without integrated tool management (external to NC)

Flat D no. structure	**Without** internal reference made to the associated tools

Selection

- With integrated tool management (inside NC)
  relative D no. structure **with** internal reference to the associated tools
  (e.g., replacement tool management and monitoring function possible)
- Without integrated tool management (outside NC)
  flat D no. structure **without** internal reference to associated tools

Machine manufacturer

Tool management: See machine manufacturer's specifications.

8.7 Activating the active tool offset immediately

Function

MD $MM_ACTIVATE_SEL_USER_DATA can be used to define that the active tool offset can be activated immediately if the parts program is in "stop" mode. See /FB1/, Description of Functions, Fundamentals, K2 Axes, "Coordinate Systems..."

⚠

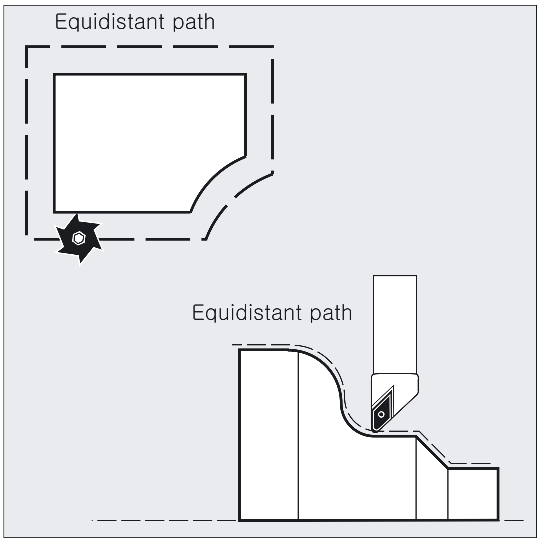
Danger

The offset is applied the next time the parts program is started.

8.8 Tool radius compensation (G40, G41, G42)

Function

When tool radius compensation is active, the control automatically calculates the equidistant tool paths for different tools.



You can generate equidistant paths with OFFN, e.g., for rough-finishing.

Programming

```
G40
```

Or

```
G41
```

Or

```
G42
```

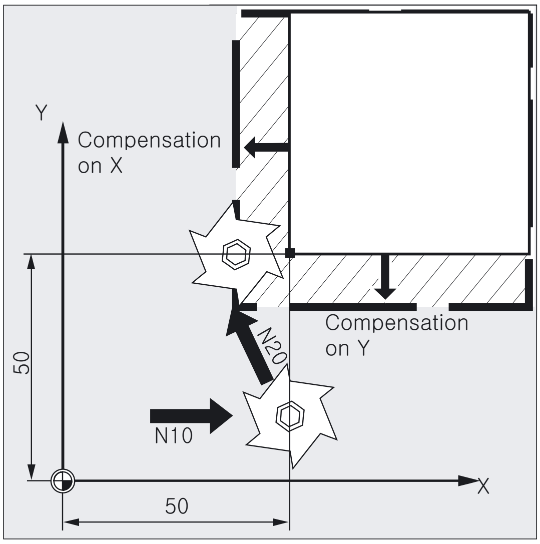
Or

```
OFFN=
```

Parameters

G40	Deactivate tool radius compensation.
G41	Activate tool radius compensation; tool operates in machining direction to the **left** of the contour.
G42	Activate tool radius compensation, tool operates in machining direction to the **right** of the contour.
OFFN=	Allowance on the programmed contour (normal contour offset).

Example 1 milling



```
N10 G0 X50 T1 D1
N20 G1 G41 Y50 F200
N30 Y100
```

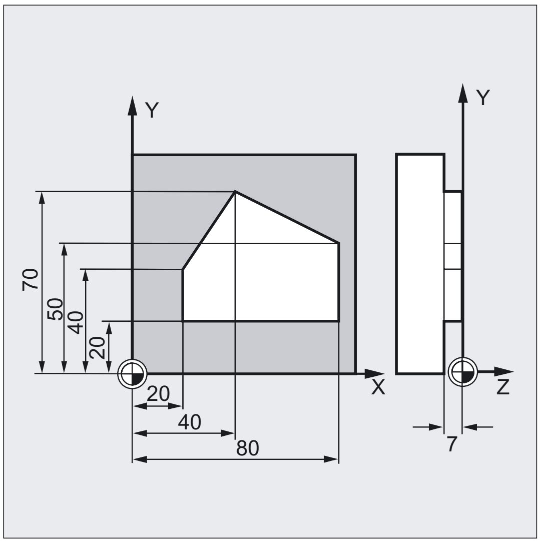
Only tool length compensation is activated in block N10. X50 is approached without compensation.

In block N20, the radius compensation is activated, point X50/Y50 is approached with compensation.

Example 2 milling

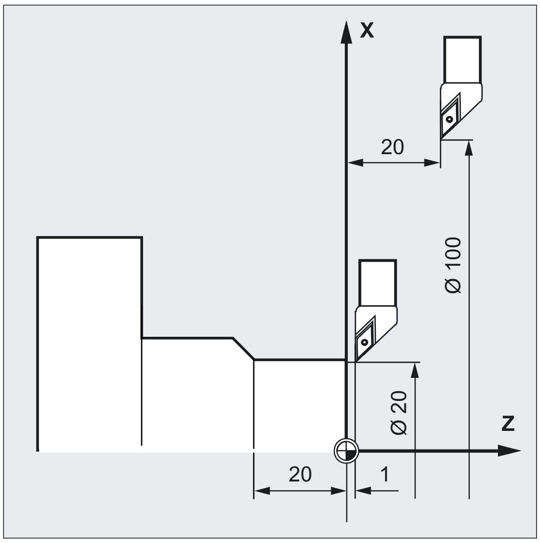
The "conventional" approach:

Call tool, load tool, activate working plane and tool radius compensation.



```
N10 G0 Z100                       ;Retract to tool change point
N20 G17 T1 M6                     ;Change tool
N30 G0 X0 Y0 Z1 M3 S300 D1        ;Call tool offset values, select length
                                  ;compensation
N40 Z-7 F500                      ;Tool infeed
N50 G41 X20 Y20                   ;Activate tool radius compensation, tool
                                  ;operates on left-hand side of 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 1 turning

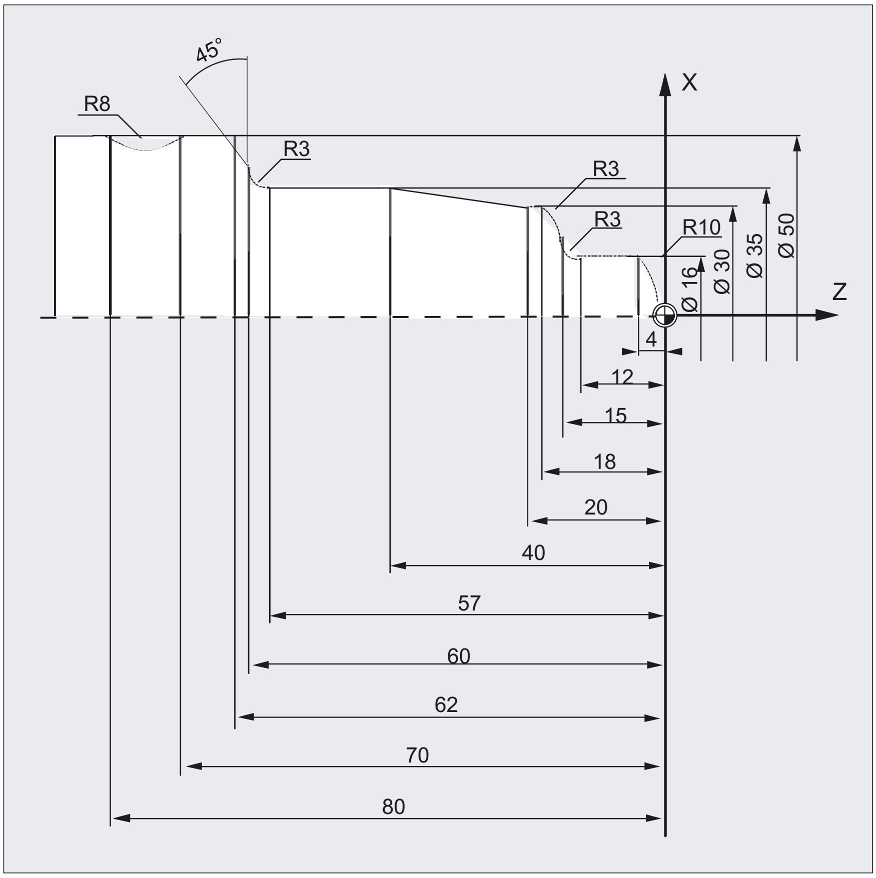


```
N20 T1 D1
N30 G0 X100 Z20
N40 G42 X20 Z1
N50 G1 Z-20 F0.2
```

Only tool length compensation is activated in block N20. X100 Z20 is approached without compensation in block N30.

In block N40, the radius compensation is activated, point X20/Z1 is approached with compensation.

Example 2 turning



```
%_N_1001_MPF                   ;Program name
N5 G0 G53 X280 Z380 D0         ;Start point
N10 TRANS X0 Z250              ;Zero offset
N15 LIMS=4000                  ;Speed limitation (G96)
N20 G96 S250 M3                ;Select constant feed
N25 G90 T1 D1 M8               ;Select tool and offset
N30 G0 G42 X-1.5 Z1            ;Activate tool with tool radius compensation
N35 G1 X0 Z0 F0.25
N40 G3 X16 Z-4 I0 K-10         ;Rotate radius 10
N45 G1 Z-12
N50 G2 X22 Z-15 CR=3           ;Rotate radius 3
N55 G1 X24
N60 G3 X30 Z-18 I0 K-3         ;Rotate radius 3
N65 G1 Z-20
N70 X35 Z-40
N75 Z-57
N80 G2 X41 Z-60 CR=3           ;Rotate 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 up tool and select offset
N105 G96 S210 M3               ;Select constant cutting speed
N110 G0 G42 X50 Z-60 M8        ;Activate tool with tool radius compensation
N115 G1 Z-70 F0.12             ;Rotate diameter 50
N120 G2 X50 Z-80 I6.245 K-5    ;Rotate radius 8
N125 G0 G40 X100 Z50 M9        ;Retract tool and deselect tool radius
                               ;compensation
N130 G0 G53 X280 Z380 D0 M5    ;Move to tool change location
N135 M30                       ;Program end
```

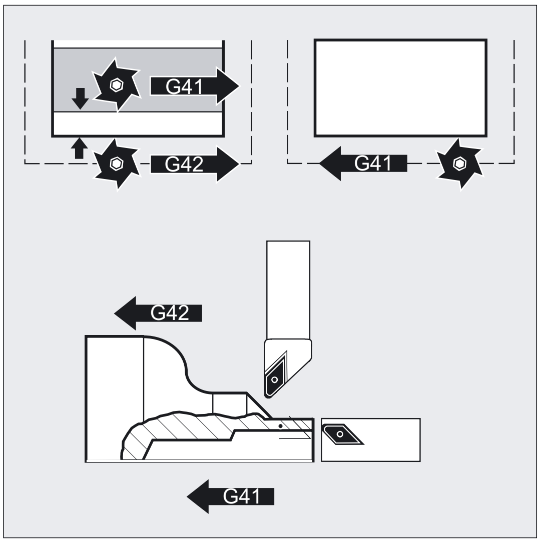
Description

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

- Tool no. T/edge no. D
- Direction of machining G41, G42
- Working plane G17 to G19

Tool no. T/edge no. D

Where appropriate, a tool offset number D is also required. The distance between the tool path and the workpiece contour is calculated from the cutter and tool edge radii and the tool point direction parameters.



With flat D number structure it is only necessary to program the D number.

Direction of machining G41, G42

From this information, the control detects the direction, in which the tool path is to be displaced.

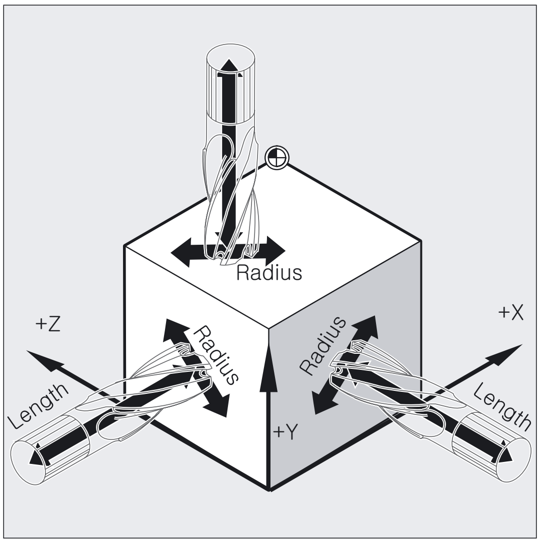
Note

A negative offset value is the same as a change of offset side (G41, G42).

You can generate equidistant paths with OFFN, e.g., for rough-finishing.

Working plane G17 toG19

From this information, the control detects the plane and therefore the axis directions for compensation.



Example of milling cutters

```
N10 G17 G41 …
```

The tool radius compensation is performed in the X/Y plane, the tool length compensation is performed in the Z direction.

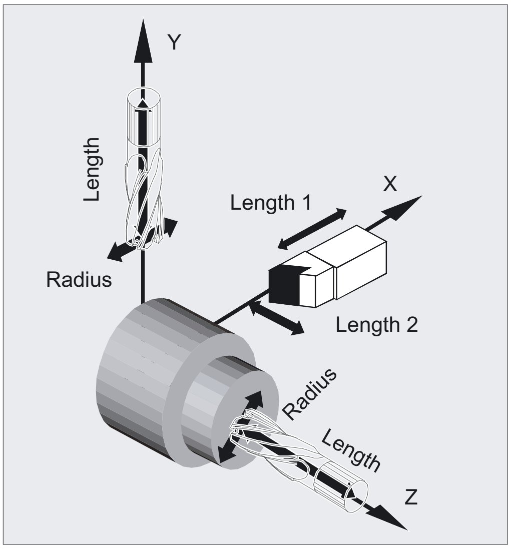
Note

On 2-axis machines, the tool radius compensation is only possible in "real" planes, in general with G18 (see tool length compensation table).

Tool length compensation

The wear parameter assigned to the diameter axis on tool selection can be defined as the diameter value (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 has been changed.

Turning:



Using NORM and KONT you can determine the tool path for activation/deactivation of compensation mode (see chapter "Contour approach and retraction", NORM, KONT, G450, G451).

Caution

Activation/deactivation of tool radius compensation

A travel command must be programmed with G0 or G1 in an NC block with G40, G41 or G42. This travel command must specify at least one axis in the selected working plane.

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

The two axes must be active as GEOAX in the channel. This can be achieved by programming them with GEOAX.

Intersection

Select intersection with SD 42496: CUTCOM_CLSD_CONT

FALSE:

If two intersections appear on the inside when offsetting an (virtually) closed contour, which consists of two circle blocks following on from one another, or from one circle block and one linear block, the intersection positioned closest to the end of block on the first partial contour is selected, in accordance with standard procedure.

A contour is deemed to be (virtually) 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 (equals 1 mm with three decimal places).

TRUE:

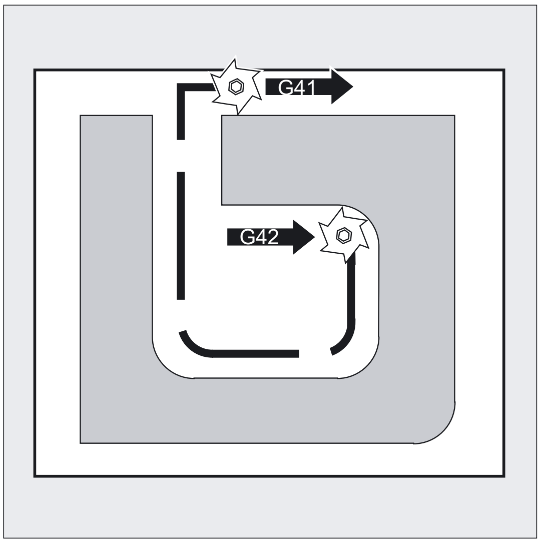
In the same situation as described above, the intersection positioned on the first partial contour closer to the block start is selected.

Changing the direction of compensation

G41/G42, G42/G41 can be programmed without an intermediate G40.

Changing the working plane

It is not possible to change the working plane G17 to G19 when G41/G42 is active.



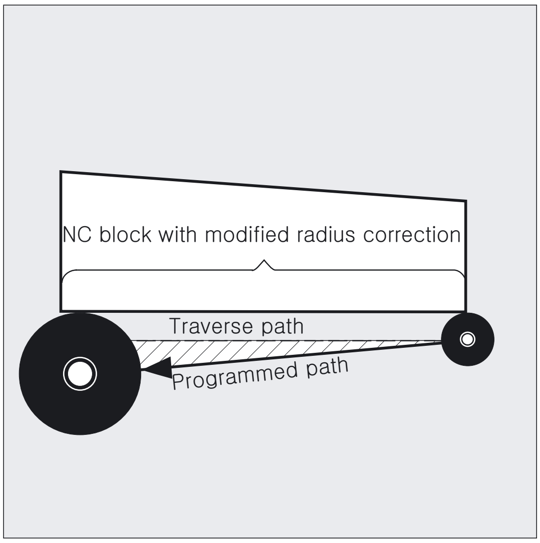
Changing the offset number D

The offset number D can be changed in compensation mode.

A modified tool radius is active with effect from the block, in which the new D number is programmed.

Caution

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



With linear movements, the tool travels along an inclined path between the starting point and end point; with circular interpolation spiral movements are produced.

Changing the tool radius

This can be achieved, for example, using system variables. The execution is the same as for changes in the D number.

Caution

The modified values only take effect the next time T or D is programmed. The change only applies with effect from the next block.

Note

Compensation mode

Compensation mode may only be interrupted by a certain number of consecutive blocks or M commands, which do not contain any travel commands or positional parameters in the compensation plane: Standard 3.

Machine manufacturer

The number of consecutive blocks or M commands can be set in machine data 20250 (see machine manufacturer).

Note

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

8.9 Contour approach and retraction (NORM, KONT, KONTC, KONTT)

Function

You can use these functions to adapt the approach and retraction paths, for example, according to the desired contour or shape of the blanks.

Only G1 blocks are permitted as original approach/retraction blocks for the two functions KONTC and KONTT. The control replaces these by polynomials for the relevant approach/retraction path.

Programming

```
NORM
```

Or

```
KONT
```

Or

```
KONTC
```

Or

```
KONTT
```

Parameters

NORM	The tool travels directly in a straight line and is positioned perpendicular to the contour point.
KONT	The tool traverses the contour point according to the programmed corner behavior G450 or G451.
KONTC	Tool reaches/leaves contour point with continuous curvature. Continuous curvature includes continuous tangent. See below. With continuous curvature means at constant acceleration.
KONTT	Tool reaches/leaves contour point along continuous tangent. Generally, continuous tangent is not continuous acceleration.

KONTC

The contour point is approached/exited with a continuous curvature. No acceleration step change occurs at the contour point. The path from the starting point to the contour point is interpolated as a polynomial.

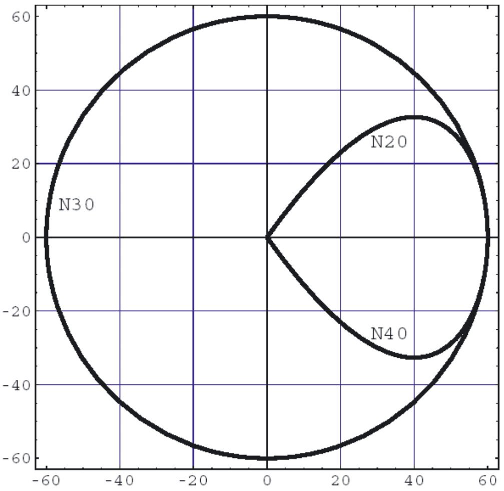
KONTT

The contour point is approached/exited along a continuous tangent. An acceleration step change might occur at the contour point. The path from the starting point to the contour point is interpolated as a polynomial.

Example of KONTC

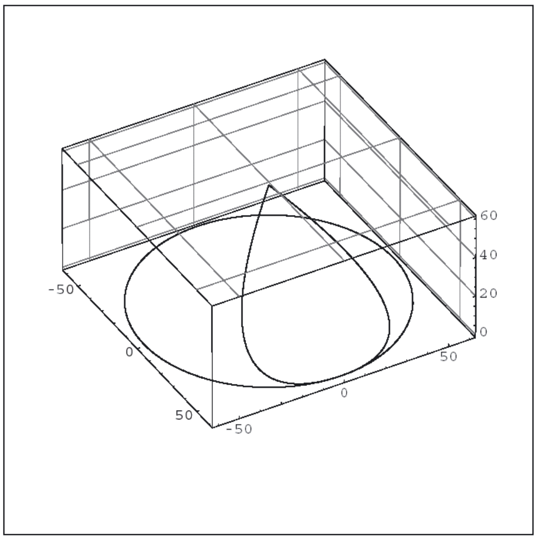
The full circle is approached beginning at the circle center point. The direction and curvature radius of the approach circle at the block end point are identical to the values of the next circle. Infeed takes place in the Z direction in both approach/retraction blocks simultaneously. The adjacent diagram shows the vertical projection of the path.

The associated NC program segment is as follows:



```
$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
```

3D representation: At the same time the curvature is adjusted to the circular path of the full circle, the axes moves from Z60 to the plane of circle Z0.

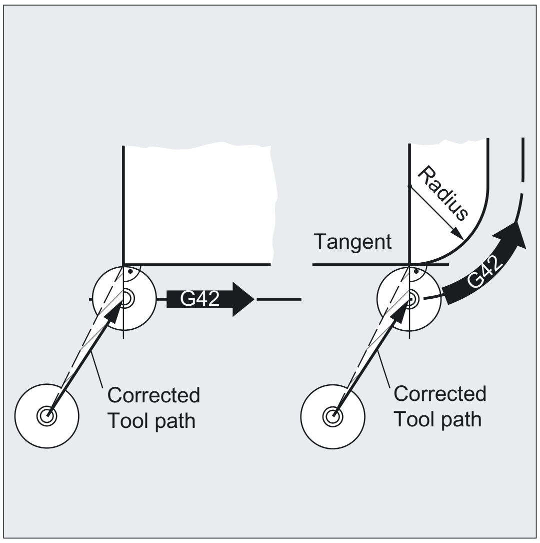


Direct approach to perpendicular position, G41, G42, NORM

The tool travels in a straight line directly to the contour and is positioned perpendicular to the path tangent at the starting point.

Selection of the approach point

When NORM is active, the tool travels directly to the compensated starting position irrespective of the approach angle programmed for the travel movement (see diagram).

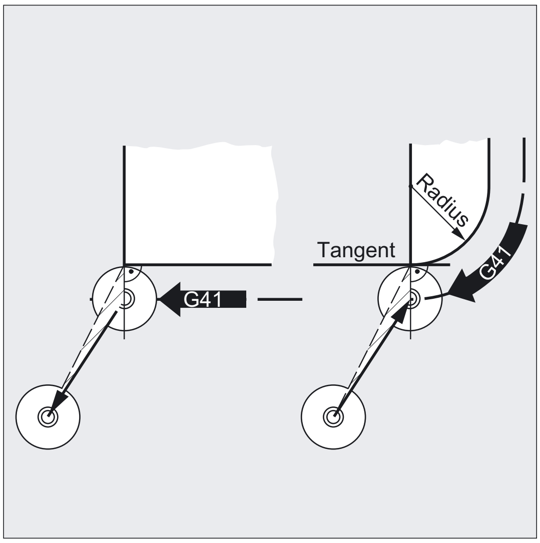


Deactivate compensation mode, G40, NORM

The tool is positioned perpendicular to the last compensated path end point and then travels directly in a straight line to the next uncompensated position, e.g., to the tool change location.

Choosing the retraction point

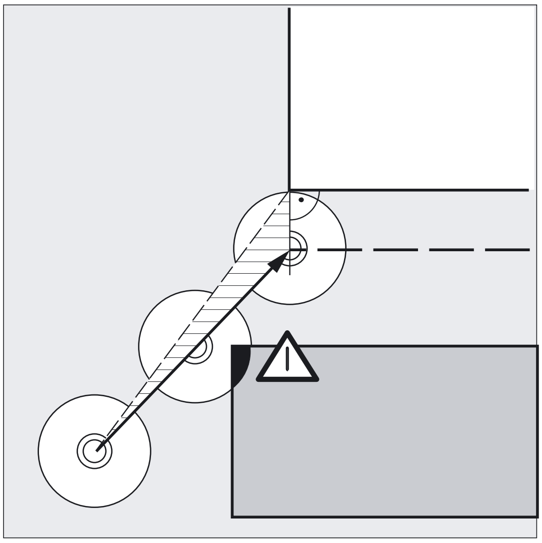
When NORM is active, the tool travels directly to the uncompensated position irrespective of the approach angle programmed for the travel movement (see diagram).



Warning

The following applies to approach and retraction movements:

You should make allowance for the modified angles of travel when programming in order to avoid collisions.



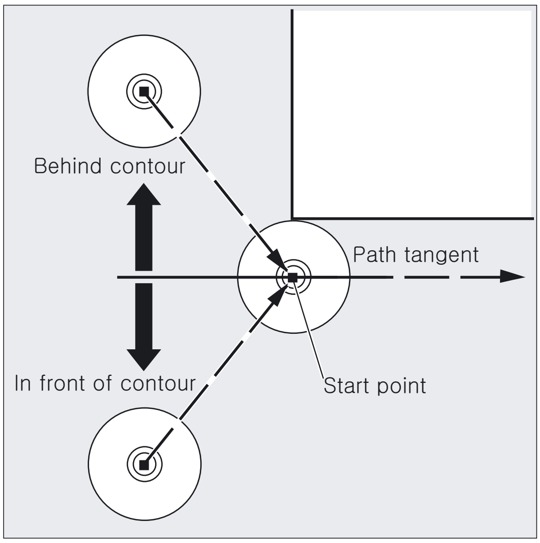
Travel round contour at starting point, G41, G42, KONT

Two cases are distinguished here:

1. Starting point lies in front of the contour

The approach strategy is the same as with NORM.

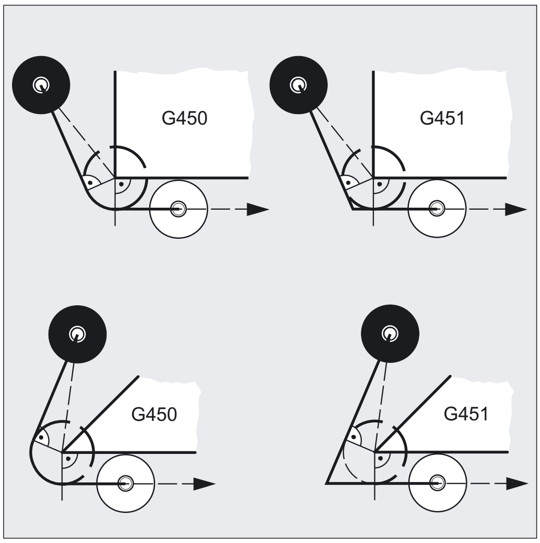
The path tangent at the starting point serves as a dividing line between the **front** and **rear** of the contour.



1. Starting point lies behind the contour

The tool travels round the starting point either along a circular path or over the intersection of the equidistant paths depending on the programmed corner behavior G450/G451.

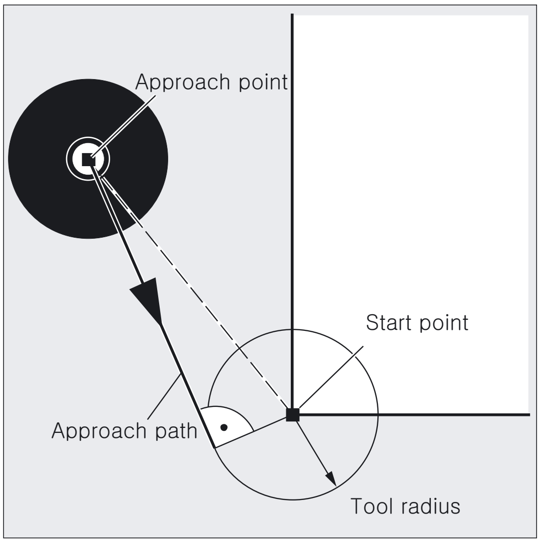
The commands G450/G451 apply to the transition from the current block to the next block.



Generation of the approach path

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.



Deactivate compensation mode, G40, KONT

If the retraction point is located in front of the contour, the same retraction movement as for NORM applies.

If the retraction point is located behind the contour, the retraction movement is the reverse of the approach movement.

Precondition for KONTC and KONTT

The two functions KONTC and KONTT are available only if polynomial interpolation is enabled in the control.

Description of KONTC and KONTT

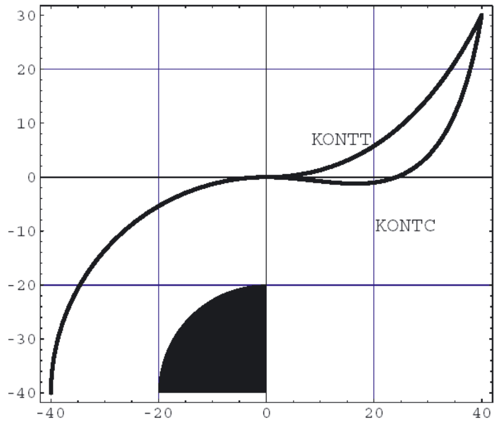
The continuity conditions are observed in all three axes. It is therefore permissible to program a path component perpendicular to the offset plane simultaneously.

Exception:

KONTT and KONTC are not available in the 3D variants of the tool radius compensation (CUT3DC, CUT3DCC, CUT3DF).

If they are programmed, the control switches internally to NORM without an error message.

Difference between KONTC and KONTT



This diagram 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 Y-40 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 blocks 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 retraction block, the contour is exactly the same, but machined in the opposite direction.

8.10 Compensation at the outside corners (G450, G451)

Function

G450/G451 defines the following:

On the one hand, the approach path for active KONT and the approach point behind the contour (see section "Contour approach and retraction").

On the other hand, the corrected tool path when traveling around outside corners.

Programming

```
G450 DISC=…
```

Or

```
G451
```

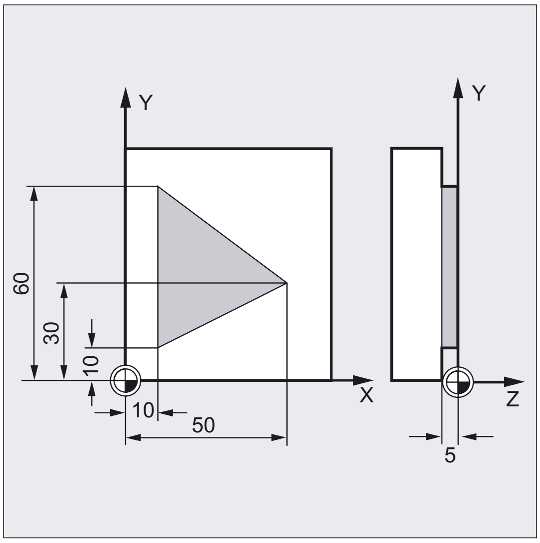
Parameters

G450	Transition circle: the tool travels around workpiece corners on a circular path with tool radius
DISC=	Flexible programming of the approach and retraction instruction. In steps of 1 from DISC=0 circle to DISC=100 intersection
G451	Intersection, the tool backs off from the workpiece corner

DISC=... is effective only when G450 is called, but can be programmed in a preceding block without G450. Both commands are modal.

Example

In this example a transition radius is added to all outside corners (progr. in block N30). This prevents the tool from having to stop and free cut when changing direction.

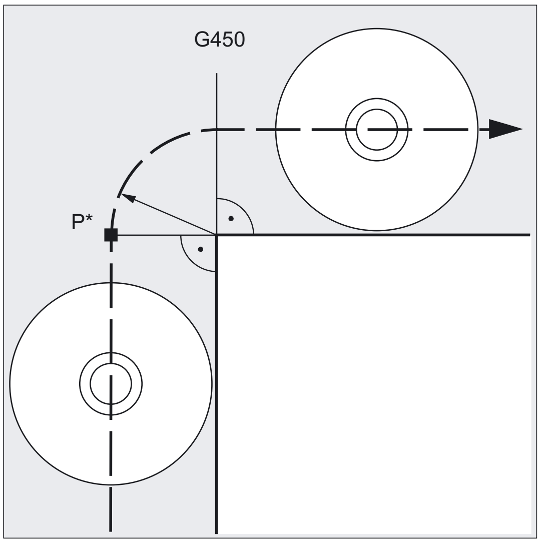


```
N10 G17 T1 G0 X35 Y0 Z0 F500      ;Start conditions
N20 G1 Z-5                        ;Tool infeed
N30 G41 KONT G450 X10 Y10         ;Activate compensation mode
N40 Y60                           ;Cut contour
N50 X50 Y30
N60 X10 Y10
N80 G40 X-20 Y50                  ;Deactivate compensation mode, retract on
                                  ;transition circle
N90 G0 Y100
N100 X200 M30
```

Corner behavior, transition circle, G41, G42, G450

The tool center point travels around the workpiece corner across an arc with tool radius.

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



The transition circle belongs to the next travel command with respect to the data.

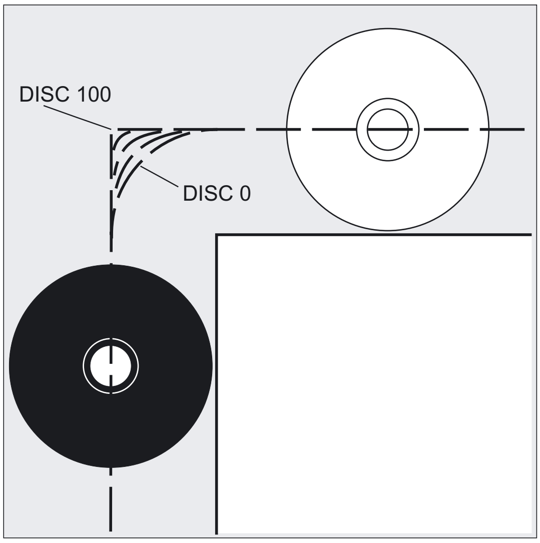
Corner behavior, selectable transitions G41, G42, G450 DISC=…

DISC distorts the transition circle, thus creating sharp contour corners.

The values have the following meanings:

DISC=0 transition circle

DISC=100 intersection of the equidistant paths (theoretical value)



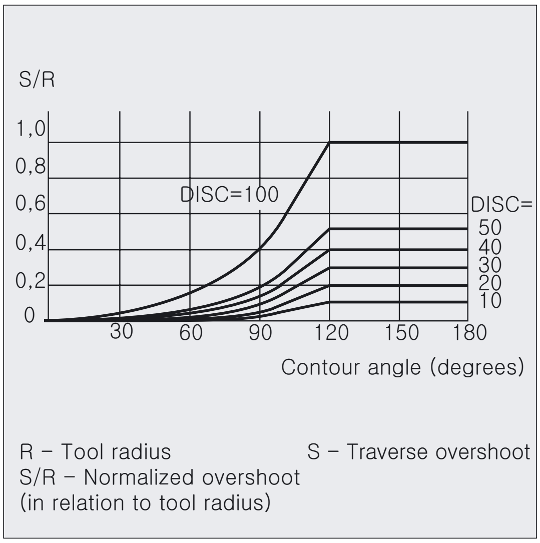
DISC is programmed in steps of 1.

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

An upper limit can be defined in machine data – generally DISC=50.

Path action, depending on DISC values and contour angle

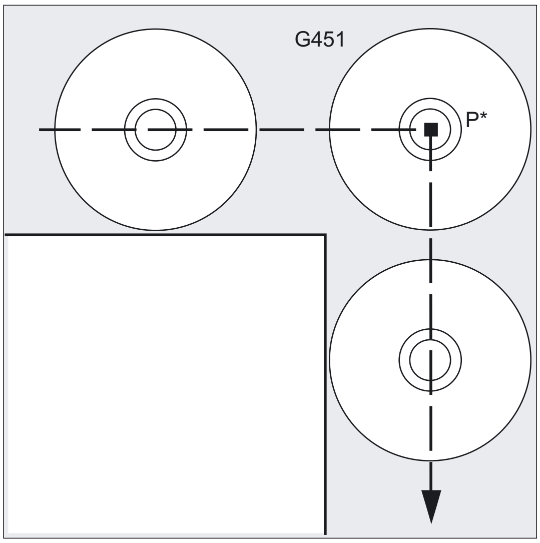
Depending on the angle of the contour that is traversed, with acute contour angles and high DISC values the tool is lifted off the contour at the corners. With angles of 120° and more, the contour is traversed evenly (see adjacent table).



Corner behavior, intersection, G41, G42, G451

The tool approaches the intersection of the two equidistants, which lie in the distance between the tool radius and the programmed contour. G451 applies only to circles and straight lines.

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



Note

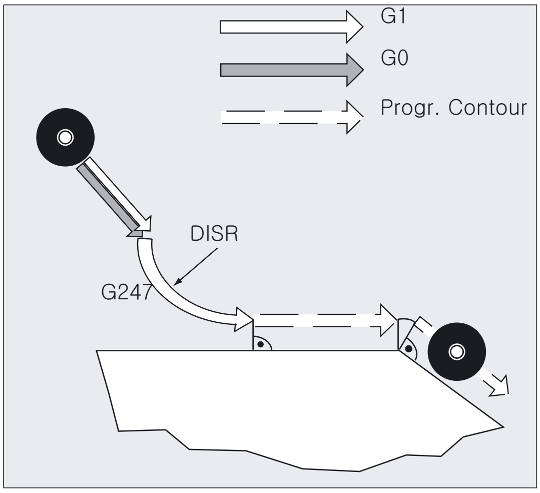
Superfluous non-cutting tool paths can result from liftoff movements at acute contour angles. A parameter can be used in the machine data to define automatic switchover to transition circle in such cases.

8.11 Smooth approach and retraction

8.11.1 Approach and retraction (G140 to G143, G147, G148, G247, G248, G347, G348, G340, G341)

Function

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



The function is mainly used in conjunction with the tool radius offset, but is not mandatory.

The approach and retraction motion consists of a maximum of 4 sub-movements:

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

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

Programming

G140

Or

G141 to G143

Or

G147, G148

Or

G247, G248

Or

G347, G348

Or

G340, G341

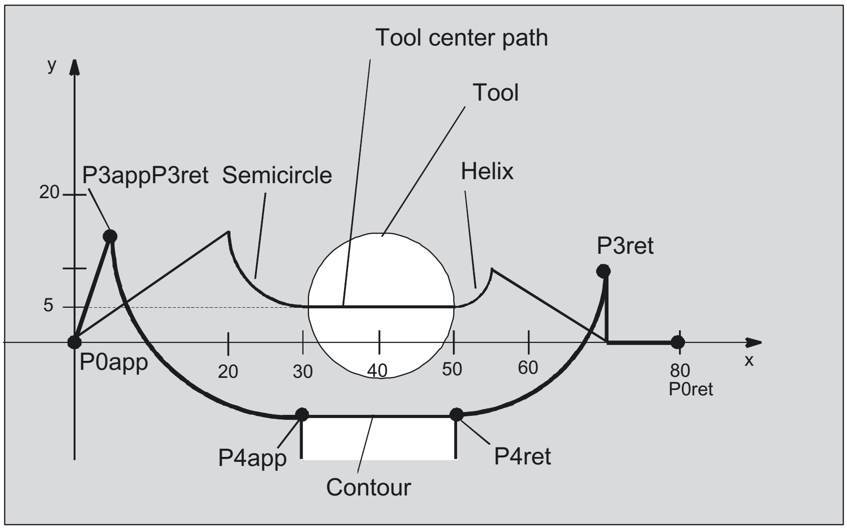
Or

DISR=..., DISCL=..., FAD=...

Parameters

G140	Approach and retraction direction independent of the current compensation side (basic setting)
G141	Approach from the left or retraction to the left
G142	Approach from the right or retraction to the right
G143	Approach and retraction direction depends on the relative position of the start and end point with respect to the tangent direction
G147	Approach with a straight line
G148	Retraction with a straight line
G247	Approach with a quadrant
G248	Retraction with a quadrant
G347	Approach with a semicircle
G348	Retraction with a semicircle
G340	Approach and retraction in space (basic setting)
G341	Approach and retraction in the plane
DISR	**Approach and retraction with straight line (G147/G148)** Distance from the mill edge to the start point of the contour
	Approach and retraction with circles (G247, G347/G248, G348) Radius of the tool center point path Caution: In the case of REPOS with a semicircle, DISR is the diameter of the circle
DISCL	DISCL=... Distance from the end point of the fast infeed motion to the machining plane DISCL=AC(...) Specifies the absolute position of the end point of the fast infeed motion
FAD	Speed of the slow infeed motion
	FAD=... the programmed value applies according to the G code of group 15 (feedrate; G93, G94, etc.) FAD=PM(...) the programmed value is interpreted as a linear feedrate (as G94), independently of the active G code, group 15 FAD=PR(...) the programmed value is interpreted as a revolutional feedrate (as G95), independently of the active G code, group 15.

Example



- Smooth approach (block N20 activated)
- Approach motion with quadrant (G247)
- Approach direction not programmed, G140 is operative, i.e., TRC is active (G41)
- Contour offset OFFN=5 (N10)
- Current tool radius=10; thus the effective offset radius for TRC=15, the radius of the SAR contour=25, so that the radius of the tool center point path is then 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 for this speed to become active in the following blocks, the G0 active in N30 must be overwritten by G1, otherwise the contour would continue to be machined with G0)
- Smooth retraction (block N60 active)
- Retraction motion 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

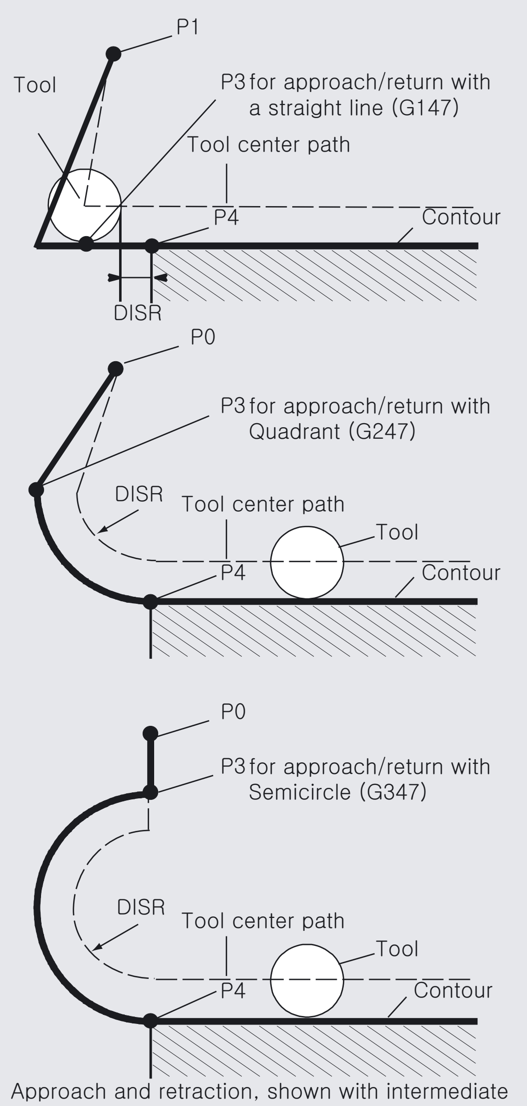
Retraction motions from Z8 to Z20 and the motion parallel to X-Y plane to X70 Y0

```
$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   ;Approach (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   ;Retract (P3ret)
N70 X80 Y0                                     ;(P0ret)
N80 M30
```

Selecting the approach and retraction contour

The appropriate G command can be used

- to approach or retract with a straight line (G147, G148),
- a quadrant (G247, G248) or
- a semicircle (G347, G348).


P1
Tool
P3 for approach/return with a straight line (G147)
Tool center path
P4
Contour
DISR
P0
P3 for approach/return with Quadrant (G247)
DISR
Tool center path
Tool
P4
Contour
P0
P3 for approach/return with Semicircle (G347)
DISR
Tool center path
Tool
P4
Contour

Approach and retraction, shown with intermediate point P4 (with simultaneous activation of tool radius compensation)

Selecting the approach and retraction direction

Use the tool radius compensation (G140, basic setting) to determine the approach and retraction direction with positive tool radius:

- G41 active → approach from left
- G42 active → approach from right

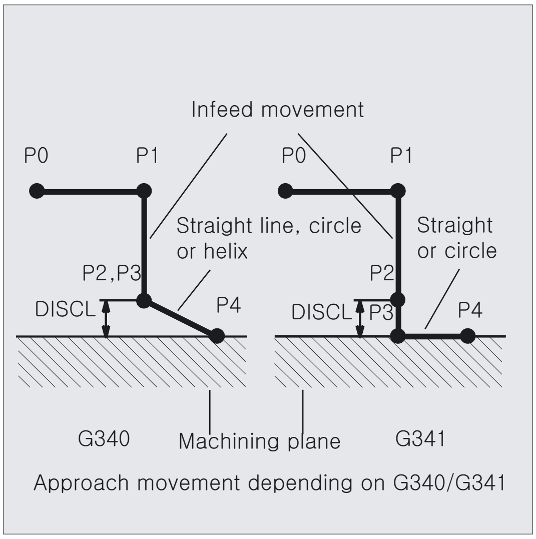
G141, G142 and G143 provide further approach options.

Description

The G codes are only significant when the approach contour is a quadrant or a semicircle.

Motion steps between start point and end point (G340 and G341)

The approach characteristic from P_0 to P_4 is shown in the adjacent image.



In cases which include the position of the active plane G17 to G19 (circular plane, helical axis, infeed motion perpendicular to the active plane), any active rotating FRAME is taken into account.

Length of the approach straight line or radius for approach circles (DISR) (see figure when selecting approach/retraction contour)

- Approach/retract with straight line
  DISR specifies the distance of the cutter edge from the starting point of the contour, i.e., the length of the straight line when TRC is active is the sum of the tool radius and the programmed value of DISR. The tool radius is only taken into account if it is positive. The resultant line length must be positive, i.e., negative values for DISR are allowed provided that the absolute value of DISR is less than the tool radius.

- Approach/retraction with circles
  DISR indicates the radius of the tool center point path. If TRC is activated, a circle is produced with a radius that results in the tool center point path with the programmed radius.

Distance of the point from the machining plane (DISCL) (see figure when selecting approach/retraction contour)

If the position of point P_2 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(...).

The following applies for DISCL=0:

- With G340: The whole of the approach motion now only consists of two blocks (P_1, P_2 and P_3 are combined). The approach contour is formed by P_1 to P_4.
- With G341: The whole approach contour consists of three blocks (P_2 and P_3 are combined). If P_0 and P_4 are on the same plane, only two blocks result (infeed movement from P_1 to P_3 is omitted).
- The point defined by DISCL is monitored to ensure that it is located between P_1 and P_3, i.e., the sign must be identical for the component perpendicular to the machining plane in all motions that possess such a component.
- On detection of a direction reversal, a tolerance defined by the machine data SAR_CLEARANCE_TOLERANCE is permitted.

Programming the end point P4 for approach or P0 for retraction

The end point is generally programmed with X... Y... Z...

- **Programming during approach**
  - P_4 in SAR block
  - P_4 is defined by the end point of the next traversing block.
    Other blocks can be inserted between the SAR block and the next traversing block without moving the geometry axes.

Example:

```
$TC_DP1[1.1]= 120                  ;Milling tool T1/D1
$TC_DP6[1.1]=7                     ;Tool with 7 mm 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:

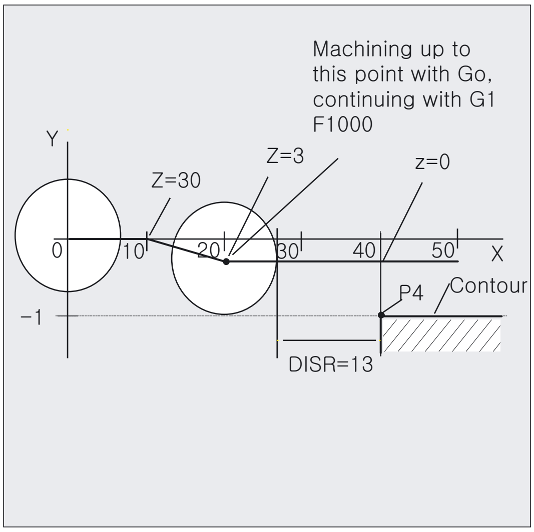
1.

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

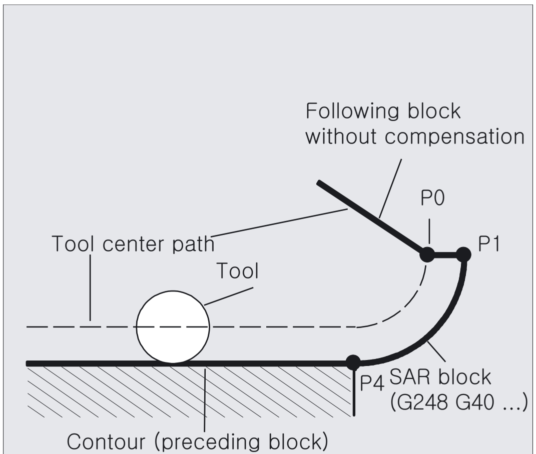
Or

2.

```
N30 G41 G147 DISCL=3 DISR=13
F1000
N40 G1 X40 Y-10 Z0
```


Machining up to this point with Go, continuing with G1 F1000
Y
Z=30
Z=3
z=0
0
10
20
30
40
50
X
P4
Contour
−1
DISR=13

- **Programming during retraction**
  - For an SAR block without programmed geometry axis, the contour ends in P_2. The position in the axes that form the machining plane are obtained from the retraction contour. The axis component perpendicular to this is defined by DISCL. If DISCL=0, movement runs fully in the plane.
  - If only the axis is programmed perpendicular to the machining plane in the SAR block, the contour ends in P_1. The position of the other axes is obtained as described previously. If the SAR block is also the deactivation block of the TRC, an additional path is inserted from P_1 to P_0, so that there is no movement at the end of the contour when the TRC is deactivated.
  - If only one axis on the machining plane is programmed, the missing 2nd axis is modally added from its last position in the previous block.
  - For an SAR block without programmed geometry axis, the contour ends in P_2. The position in the axes that form the machining plane are obtained from the retraction contour. The axis component perpendicular to this is defined by DISCL. If DISCL=0, movement runs fully in the plane.
  - If only the axis is programmed perpendicular to the machining plane in the SAR block, the contour ends in P_1. The position of the other axes is obtained as described previously. If the SAR block is also the deactivation block of the TRC, an additional path is inserted from P_1 to P_0, so that there is no movement at the end of the contour when the TRC is deactivated.
  - If only one axis on the machining plane is programmed, the missing 2nd axis is modally added from its last position in the previous block.



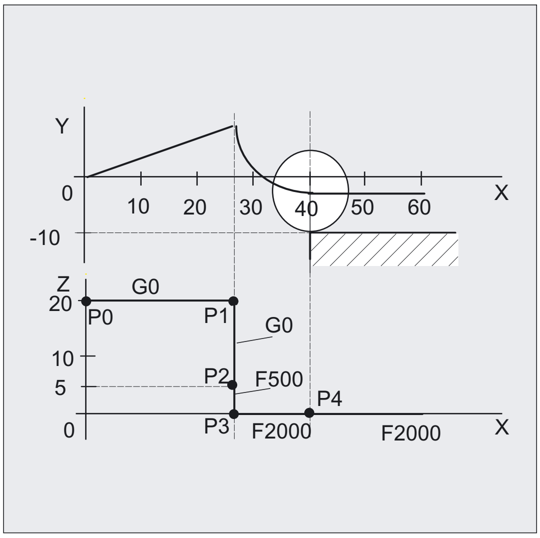
Retraction with SAR and simultaneous deactivation of TRC

Approach and retraction velocities

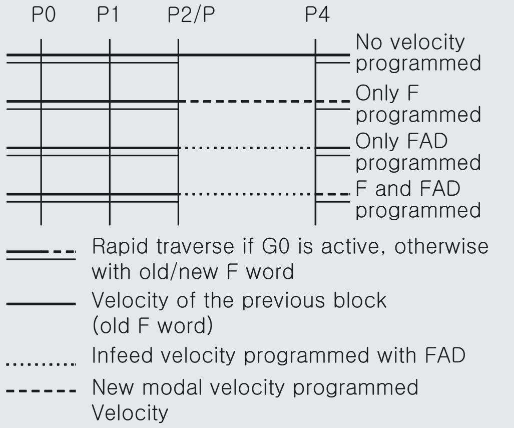
- Velocity of the previous block (G0):
  All motions from $_0$ up to P_2 are executed at this velocity, i.e., the motion parallel to the machining plane and the part of the infeed motion up to the safety clearance.
- Programming with FAD:
  Specify the feedrate for
  - G341: infeed movement perpendicular to the machining plane from P_2 to P_3
  - G340: from point P_2 or P_3 to P_4
    If FAD is not programmed, this part of the contour is also traversed at the modally active speed of the previous block, if no F word is programmed in the SAR block.
- Programmed feedrate F:
  This feedrate value is effective as of $_3$ or P_2 if FAD is not programmed. If no F word is programmed in the SAR block, the speed of the previous block is active.

Example:

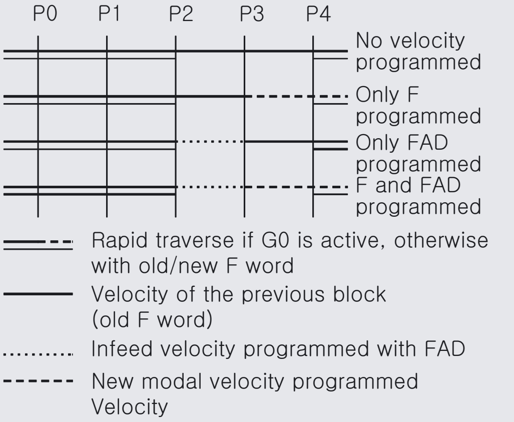
```
$TC_DP1[1.1]= 120                 ;Milling tool T1/D1
$TC_DP6[1.1]=7                    ;Tool with 7 mm radius
N10 G90 G0 X0 Y0 Z20 D1 T1
N20 G41 G341 G247 DISCL=AC(5)
DISR=13
FAD 500 X40 Y-10 Z=0 F200
N30 X50
N40 X60
...
```



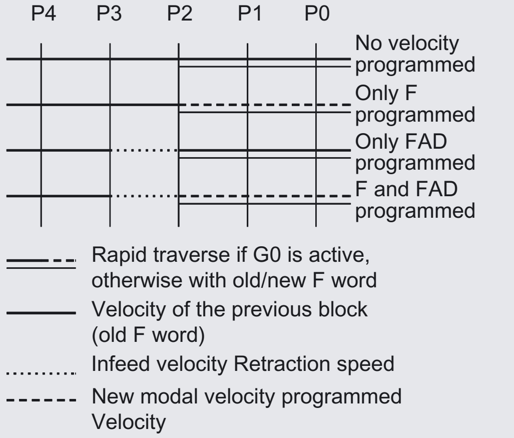
During retraction, the rolls of the modally active feedrate from the previous block and the programmed feedrate value in the SAR block are changed round, i.e., the actual retraction contour is traversed with the old feedrate value and a new speed programmed with the F word applies from P_2 up to P_0.


P0
P1
P2/P
P4
No velocity programmed
Only F programmed
Only FAD programmed
F and FAD programmed
Rapid traverse if G0 is active, otherwise with old/new F word
Velocity of the previous block (old F word)
Infeed velocity programmed with FAD
New modal velocity programmed Velocity

Velocities in the SAR subblocks
on retraction with G340


P0
P1
P2
P3
P4
No velocity programmed
Only F programmed
Only FAD programmed
F and FAD programmed
Rapid traverse if G0 is active, otherwise with old/new F word
Velocity of the previous block (old F word)
Infeed velocity programmed with FAD
New modal velocity programmed Velocity

Velocities in the SAR subblocks
on retraction with G341


P4
P3
P2
P1
P0
No velocity programmed
Only F programmed
Only FAD programmed
F and FAD programmed
Rapid traverse if G0 is active, otherwise with old/new F word
Velocity of the previous block (old F word)
Infeed velocity Retraction speed
New modal velocity programmed Velocity

Velocities in the SAR subblocks
on retraction

Reading positions

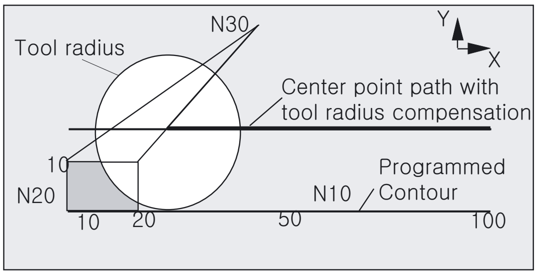
Points P_3 and P_4 can be read in the WCS as a system variable during approach.

- $P_APR: reading P
- $_3$ (initial point)
- $P_AEP: reading P
- $_4$ (contour starting point)
- $P_APDV: read whether $P_APR and $P_AEP contain valid data

8.11.2 Approach and retraction with enhanced retraction strategies (G460, G461, G462)

Function

In certain special geometrical situations, enhanced approach and retraction strategies, compared with the previous implementation with activated collision monitoring for approach and retraction block, are required in order to activate or deactivate tool radius compensation. For instance, collision monitoring can lead to a section of the contour not being fully machined. See following image.



Retraction behavior with G460

Programming

```
G460
```

or

```
G461
```

or

```
G462
```

Parameters

G460 As before (activation of collision monitoring for approach and retraction block)

G461 Insertion of a circle in the TRC block, if no intersection point is possible, whose center point is at the end point of the uncorrected block and whose radius is equal to the tool radius.

Auxiliary circuit is used to machine around the contour end point (i.e. up to the end of the countour) up to the intersection.

G462 Insertion of a straight line in the TRC block if no intersection point is possible; the block is extended by its end tangent (default setting)

Machining is done up to the **extension** of the last contour element (i.e. up to just before the contour end).

Further information

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 of retraction behavior with G460

The following example describes only the situation for deactivation of tool radius compensation. The behavior for approach is the same.

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

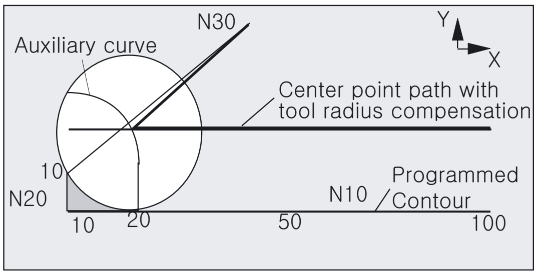
Example of approach with G461

```
N10 $TC_DP1[1,1]=120              ;Tool type: milling tool
N20 $TC_DP6[1,1]=10               ;Tool 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
```

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.

The control attempts to cut this circle with one of the preceding blocks.



Retraction behavior with G461 (see example)

Collision monitoring CDON, CDOF

If CDOF is active (see section Collision monitoring, CDON, CDOF), the search is aborted when an intersection is found, i.e., the system does not check whether further intersections with previous blocks exist.

If CDON is active, the search continues for further intersections 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.

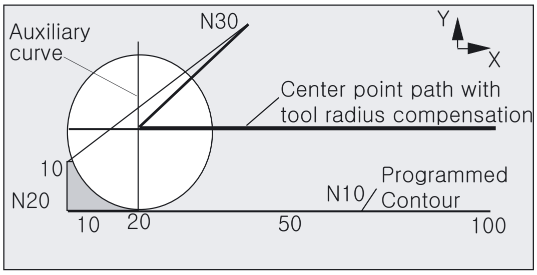
Note

If no intersection is found, alarm 10751 (collision danger) is output.

G462

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

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



Retraction behavior with G462 (see example)

With G462, the corner generated by N10 and N20 in the example 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.

Corner behavior with KONT

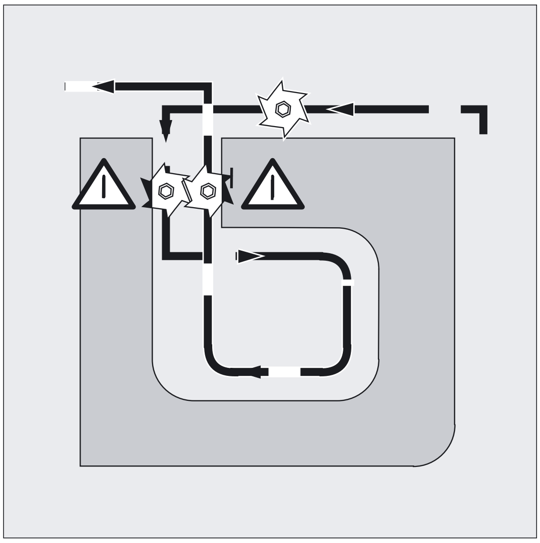
If KONT is active (travel round contour at start or end point), the behavior differs according to whether the end point is in front of or behind the contour.

- End point in front of contour
  If the end point is in front of the contour, the retraction behavior is the same as with NORM. This property does not change even if the last contour block for 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.
- Endpoint behind the contour
  If the end point is behind the contour, a circle or straight line is always inserted depending on G450/G451. In this case, G460-462 has no effect. If the last traversing block in this situation has no intersection with a preceding block, an intersection with the inserted contour element or with the straight line of the end point of the bypass circle to the programmed endpoint can result.
  If the inserted contour element is a circle (G450), and this forms an interface with the preceding block, this is equal to the interface that would occur with NORM and G461. In general, however, a remaining section of the circle still has to be traversed. For the linear part of the retraction block, no further calculation of intersection is required.
  In the second case, if no interface of the inserted contour element with the preceding blocks is found, the intersection between the retraction straight line and a preceding block is traversed.
  Therefore, a behavior that deviates from G460 can only occur with active G461 or G462 either if NORM is active or the behavior with KONT is geometrically identical to that with NORM.

8.12 Collision monitoring (CDON, CDOF, CDOF2)

Function

When CDON (Collision Detection ON) and tool radius compensation are active, the control monitors the tool paths with Look Ahead contour calculation. This Look Ahead function allows possible collisions to be detected in advance and permits the control to actively avoid them.



When collision detection is off (CDOF), a search is made at inside corners in the previous traversing block (and if necessary in blocks further back) for a common intersection point for the current block. If no intersection is found with this method, an error is generated.

Programming

```
CDON
```

Or

```
CDOF
```

Or

```
CDOF2
```

Parameters

CDON	Activate bottleneck detection
CDOF	Deactivate bottleneck detection
CDOF2	Determine tool offset direction from adjacent block parts. CDOF2 is active only for 3D circumferential milling.

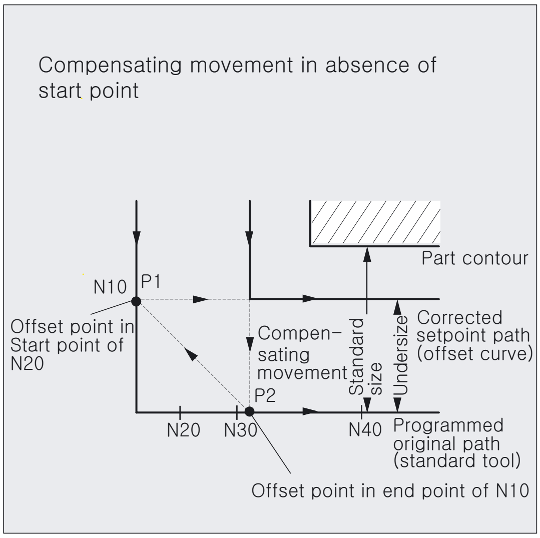
CDOF helps prevent the incorrect detection of bottlenecks, e.g., due to missing information, which is not available in the NC program.

Machine manufacturer

The number of NC blocks monitored can be defined in the machine data (see machine manufacturer).

Example of milling on the center-point path with standard tools

The NC program defines the center-point path of a standard tool. The contour for a tool currently in use results in an undersize, which is shown on an unrealistically large scale solely to illustrate the geometric conditions. The following description is based on the simple assumption that the control can only "see" three blocks.


Compensating movement in absence of start point
Part contour
N10
P1
Offset point in Start point of N20
Compen-sating movement
Standard size
Undersize
Corrected setpoint path (offset curve)
P2
N20
N30
N40
Programmed original path (standard tool)
Offset point in end point of N10

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 this instance, the control is still not aware of block N40 if N10 must be processed subsequently. Only a single block can therefore be omitted.

When CDOF2 is active, the compensatory motion shown in the diagram is executed and not halted. In this situation, an active CDOF or CDON would generate an alarm.

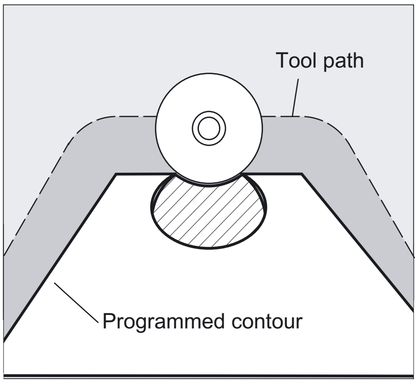
Identifying critical machining situations

The following are some examples of critical machining situations, which can be detected by the control and compensated for by modifying the tool paths.

In order to prevent program stops, you should always select the tool with the widest radius from all of the tools used when testing the program.

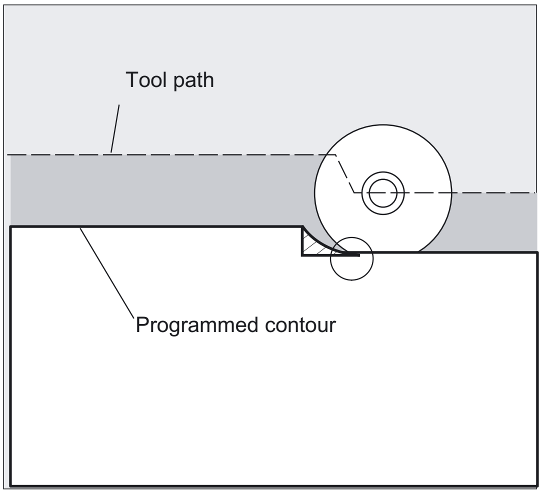
In each of the following examples a tool with too wide a radius was selected for machining the contour.

Bottleneck detection



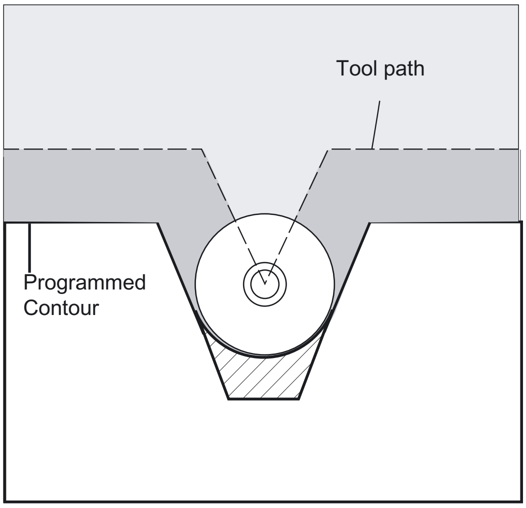
Since the tool radius selected is too wide to machine this inside contour, the "bottleneck" is bypassed.
An alarm is output.

Contour path shorter than tool radius



The tool travels round the workpiece corner on a transition circle and then continues to follow the programmed contour exactly.

Tool radius too wide for inside machining



In such cases, the contours are machined only to the extent possible without damaging the contour.

8.13 2 ½ D tool offset (CUT2D, CUT2DF)

Function

With CUT2D or CUT2DF you define how the tool radius compensation is to act or to be interpreted when machining in inclined planes.

Tool length compensation

The tool length compensation generally always refers to the fixed, non-rotated working plane.

Programming

```
CUT2D
```

Or

```
CUT2DF
```

Parameters

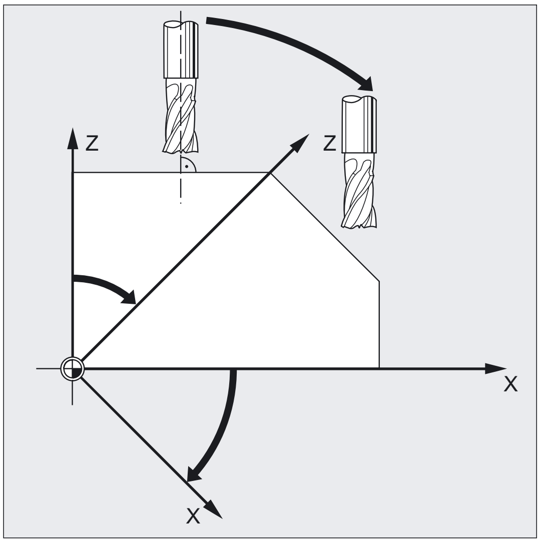
CUT2D	Activate 2 1/2 D radius compensation (default)
CUT2DF	Activate 2 1/2 D radius compensation, tool radius compensation relative to the current frame or to inclined planes

CUT2D is used when the orientation of the tool cannot be changed and the workpiece is rotated for machining on inclined surfaces.

CUT2D is generally the standard setting and does not, therefore, have to be specified explicitly.

Tool radius compensation, CUT2D

As for many applications, tool length compensation and tool radius compensation are calculated in the **fixed** working plane specified with G17 to G19.



Example of G17 (X/Y plane):

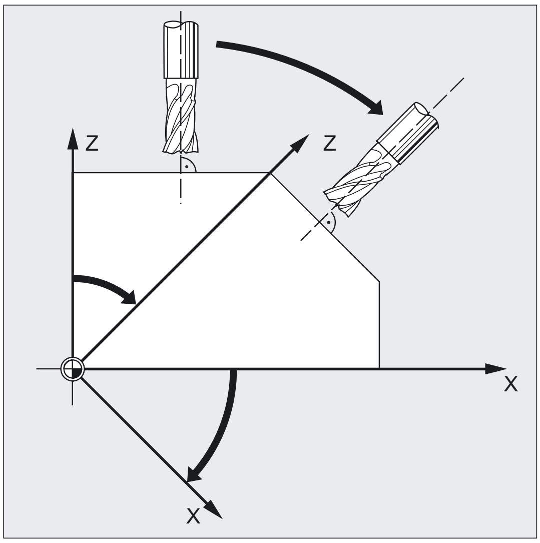
Tool radius compensation is active in the non-rotated X/Y plane, tool length compensation in the Z direction.

Tool offset values

For machining on inclined surfaces, the tool compensation values have to be defined accordingly, or be calculated using the functions for "Tool length compensation for orientable tools". For more information on this calculation method, see chapter "Tool orientation and tool length compensation".

Tool radius compensation, CUT2DF

In this case, it is possible to arrange the tool orientation perpendicular to the inclined working plane on the machine.



If a frame containing a rotation is programmed, the compensation plane is also rotated with CUT2DF. The tool radius compensation is calculated in the rotated machining plane.

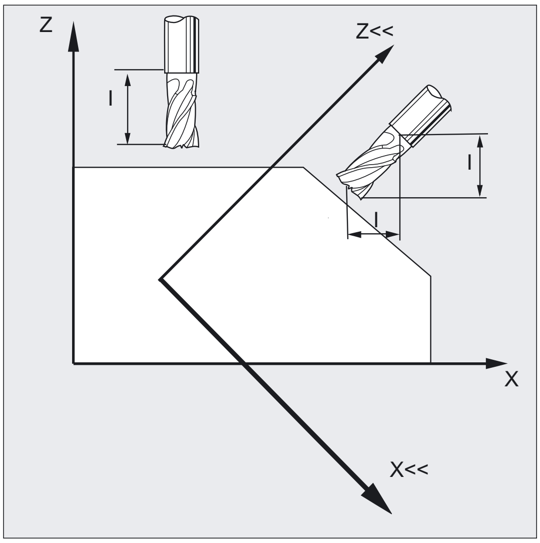
Note

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

8.14 Tool length compensation for orientable toolholders (TCARR, TCOABS, TCOFR)

Function

When the spatial orientation of the tool changes, its tool length components also change.



After a reset, e.g., through manual setting or change of the toolholder with a fixed spatial orientation, the tool length components also have to be determined again. This is performed using the TCOABS and TCOFR path commands.

It is possible for an orientable toolholder of an active frame to determine the direction of the tool when a tool is selected with TCOFRZ, TCOFRY, and TCOFRX.

Programming

```
TCARR=[m]
```
or

```
TCOABS
```
or

```
TCOFR
```
or

```
TCOFRZ, TCOFRY, TCOFRX
```

Parameters

TCARR=[m]	Request toolholder with the number "m"
TCOABS	Determine tool length components from the orientation of the current toolholder
TCOFR	Determine tool length components from the orientation of the active frame
TCOFRZ	Orientable toolholder from active frame with a tool pointing in the Z direction
TCOFRY	Orientable toolholder from active frame with a tool pointing in the Y direction
TCOFRX	Orientable toolholder from active frame with a tool pointing in the X direction

Determine tool length compensation from the orientation of the toolholder, TCOABS

TCOABS calculates the tool length compensation from the current orientation angles of the toolholder; stored in system variables $TC_CARR13 and $TC_CARR14.

For a definition of toolholder kinematics with system variables, see
References: /PGA/Programming Guide, Advanced, Chapter "Toolholder kinematics"

In order to make a new calculation of the tool length compensation when frames are changed, the tool has to be selected again.

Tool direction from active frame

It is possible to set the orientating toolholder such that command

- TCOFR or TCOFRZ points in Z direction.
- TCOFRY points in Y direction.
- TCOFRX points in X direction.

If there is a switchover between TCOFR and TCABS, the tool length compensation is calculated again.

Request toolholder, TCARR

With TCARR, the toolholder number m is requested with its geometry data (offset memory). With m=0, the active toolholder is deselected.

The geometry data of the toolholder only become active after a tool is called. The selected tool remains active after a toolholder change has taken place.

The current geometry data for the toolholder can also be defined in the parts program via the corresponding system variables.

Recalculation of tool length compensation, TCOABS with frame change

In order to make a new calculation of the tool length compensation when frames are changed, the tool has to be selected again.

Note

The tool orientation must be manually adapted to the active frame.

When the tool length compensation is calculated, the angle of rotation of the toolholder is calculated in an intermediate step. With toolholders with two rotary axes, there are generally two sets of rotation angles, which can be used to adapt the tool orientation to the active frame; therefore, the rotation angle values stored in the system variables must at least correspond approximately to the mechanically set rotation angles.

Note

Tool orientation

It is not possible for the control to check whether the rotation angles calculated by means of the frame orientation are settable on the machine.

If the rotary axes of the toolholder are arranged such that the tool orientation calculated by means of the frame orientation cannot be reached, then an alarm is output.

The combination of tool precision compensation and the functions for tool length compensation on movable toolholders is not permissible. If both functions are called simultaneously, an error message is issued.

The **TOFRAME** function allows a frame to be defined on the basis of the direction of orientation of the selected toolholder. For more information please refer to chapter "Frames".

When orientation transformation is active (3, 4 or 5-axis transformation), it is possible to select a toolholder with an orientation deviating from the zero position without causing output of an alarm.

8.15 Grinding-specific tool monitoring in parts programs (TMON, TMOF)

Function

The command TMON is used to activate geometry and speed monitoring for grinding tools (types 400-499) in the NC parts program. Monitoring remains active until it is deactivated in the parts program with TMOF.

Machine manufacturer

See machine manufacturer's specifications

Programming

```
TMON (T no.)
```

Or

```
TMOF (T no.)
```

Parameters

TMON (T no.)	Activate tool monitoring	It is only necessary to specify the T number if the tool with this T number is not active.
TMOF (T no.)	Deselect tool monitoring T No. = 0: Deactivate monitoring for all tools	

Assignment of tool-specific parameters

Further tool-specific parameters can be set up in the machine data and assigned by the user.

Parameters	Meaning	Data type
Tool-specific parameters		
$TC_TPG1	Spindle number	Integer
$TC_TPG2	Chaining rule The parameters are automatically kept identical for the left and right side of the wheel.	Integer
$TC_TPG3	Minimum wheel radius	Real
$TC_TPG4	Minimum wheel width	Real
$TC_TPG5	Current wheel width	Real
$TC_TPG6	Maximum speed	Real
$TC_TPG7	Maximum peripheral speed	Real
$TC_TPG8	Angle of inclined wheel	Real
$TC_TPG9	Parameter number for radius calculation	Integer

Requirements

You can only activate tool monitoring if the tool-specific grinding data $TC_TPG1 to $TC_TPG9 (see /PGA/Programming Guide "Advanced") are set.

According to the machine data settings, tool monitoring for the grinding tools (types 400-499) can be automatically activated when the tool selection is activated.

Only **one** monitoring routine can be active at any one time for each spindle.

Geometry monitoring

The current wheel radius and the current width are monitored.

The set speed is monitored against the speed limitation cyclically with allowance for the spindle override.

The speed limit is the smaller value resulting from a comparison of the maximum speed with the speed calculated from the maximum wheel peripheral speed and the current wheel radius.

Working without a T or D number

In the machine data, a default T and D number

- T number, and
- D number can be set,

that do not have to be reprogrammed and are effective after Power ON/Reset.

Example

All machining is performed with the same grinding wheel.

Machine data can be set to keep the current tool active after Reset; see /PGA/Programming Guide Advanced.

8.16 Additive offsets

Function

Additive offsets are essentially process offsets which can be programmed during machining. They refer to the geometrical data of a tool edge and are thus components of the tool edge data.

The data of an additive offset are addressed via a DL number (DL: location-dependent; offsets relative to relevant location) and enter tool offset via parameter display in the Parameter area.

References: /BAD, BEM/ "Operator's Guide HMI Advanced, HMI Embedded" chapter "Parameters"

8.16.1 Select offsets (via DL numbers)

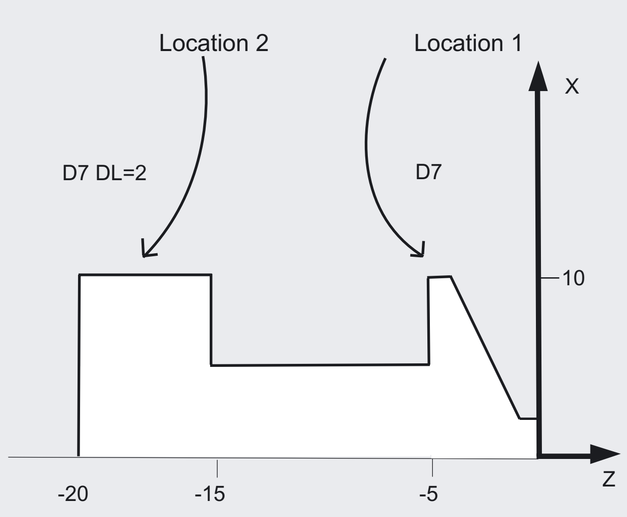
Function

Setup value:

The setup value is defined optionally by the machine manufacturer in MD.

Same tool edge:

The same tool edge is used for 2 bearing seats (see example). Compensation can be made for a location-dependent measurement error occurring as a result of machining forces, etc.



Fine offset:

Location-dependent allowances can be made for over/under-dimensioning.

Parameters

Machine data are used to activate and define the number of additive offsets. Please refer to the machine manufacturer's instructions.

Example

```
N110 T7 D7                        ;The tool turret is positioned at location 7
                                  ;D7 and DL=1 are activated and traversed in the
                                  ;next block
N120 G0 X10 Z1                    ;N120
N130 G1 Z-6
N140 G0 DL=2 Z-14                 ;DL=2 is activated in addition to D7 and traversed
                                  ;in the next block
N150 G1 Z-21
N160 G0 X200 Z200                 ;Approach tool change point
...
```

8.16.2 Specify wear and setup values ($TC_SCPxy[t,d], $TC_ECPxy[t,d])

Function

Wear and setup values can be read and written via system variables and the corresponding OPI services.

The logic is based on the logic of the corresponding system variables for tools and tool noses.

Programming

`$TC_SCPxy [t,d]` Wear values

or

`$TC_ECPxy [t,d]` Setup values

Parameter

$TC_SCPxy	Wear values are assigned to the corresponding geometry parameters via xy, where x is the number of the wear value and y is the reference to the geometry parameter.
$TC_ECPxy	Setup values are assigned to the corresponding geometry parameters via xy, where x is the number of the setup value and y is the reference to the geometry parameter.
t	T number of the tool
d	D number of the tool nose

Note

The defined wear and setup values are added to the geometry parameters and the other offset parameters (D numbers).

Example

The wear value of length 1 is set to the value 1.0 for the tool nose (D number d) of the tool (t).

Parameter: $TC_DP3 (length 1, with turning tools)

Wear values: $TC_SCP13 to $TC_SCP63

Setup values: $TC_ECP13 to $TC_ECP63

$TC_SCP43 [t, d] = 1.0

8.16.3 Delete additive offsets (DELDL)

Function

DELDL is used to delete the additive offsets for the tool edge of a tool (in order to release memory). Both the defined wear values and the setup values are deleted.

Programming

```
status = DELDL [t,d]
```

Parameters

DELDL [t,d]	All additive offsets of the tool edge with D number d of tool t are deleted
DELDL[t]	All additive offsets of all tool edges of tool t are deleted
DELDL	All additive offsets of the tool edges of all tools of the TO- unit are deleted (for the channel, in which the command is programmed)
status	0: Offsets have been successfully deleted. -: Offsets have not been deleted (if the parameter settings specify exactly one tool edge), or not deleted completely (if the parameter settings identify several tool edges).

Note

The wear and setup values of active tools cannot be deleted (behave similar to deletion of D or tool data).

8.17 Special handling of tool offsets

Function

Setting data SD 42900 - SD 42960 can be used to control the evaluation of the sign for tool length and wear.

The same applies to the behavior of the wear components when mirroring geometry axes or changing the machining plane, and also to temperature compensation in tool direction.

Parameters

Where reference is made below to wear values, this means in each case the sum of the actual wear values ($TC_DP12 to $TC_DP20) and the total offsets with the wear ($SCPX3 to $SCPX11) and setup values ($ECPX3 to $ECPX11).

For more details on total offsets, see:

References:/FBW/, Description of Functions, Tool Management.

SD42900 MIRROR_TOOL_LENGTH	Mirroring of tool length components and components of the tool base dimension
D42910 MIRROR_TOOL_WEAR	Mirroring of wear values of the tool length components
SD42920 WEAR_SIGN_CUTPOS	Sign evaluation of the wear components depending on the tool point direction
SD42930 WEAR_SIGN	Inverts the sign of the wear dimensions
SD42935 WEAR_TRANSFORM	Transformation of wear values
SD42940 TOOL_LENGTH_CONST	Assignment of tool length components to geometry axes
SD42950 TOOL_LENGTH_TYPE	Assignment of tool length components irrespective of tool type
SD42960 TOOL_TEMP_COMP	Temperature compensation value in tool direction. Also operative when tool orientation is programmed.

References:

/PGA/, Programming Guide Advanced, Chapter 8

/FB1/, Description of Functions, Basic Machine (Part 1), Tool Compensation (W1)

Description

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.

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.

Orientatable toolholders and new setting data

Setting data SD 42900-42940 have no effect on the components of an active orientatable toolholder. However, the calculation with an orientatable toolholder always allows for a tool with its total resulting length (tool length + wear + tool base dimension). All modifications initiated by the setting data are included in the calculation of the resulting total length; i.e. vectors of the orientatable toolholder are independent of the machining plane.

Further information

When orientatable toolholders are used, it is frequently practical 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 toolholder 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.

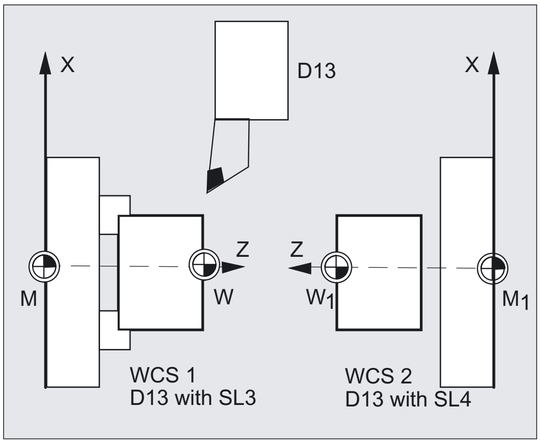
Further application options

The use of orientatable toolholder functionality can also be useful if there is no physical option of turning tools on the machine, even though tools with different orientations are permanently installed. 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 toolholder.

8.17.1 Mirroring of tool lengths

Function

Set setting data SD 42900 MIRROR_TOOL_LENGTH and SD 42910 MIRROR_TOOL_WEAR not equal to zero can be used to mirror tool length components and components of the tool base dimensions with wear values of the corresponding axes.



Parameters

`SD 42900 MIRROR_TOOL_LENGTH`

Setting data **not equal to** zero:

The tool length components ($TC_DP3, $TC_DP4 and $TC_DP5) and the components of the tool base dimensions ($TC_DP21, $TC_DP22 and $TC_DP23), whose associated axes are mirrored, are also mirrored – through sign inversion.

The wear values are **not** mirrored. If these are also to be mirrored, setting data $SC_MIRROR_TOOL_WEAR must be enabled.

`SD 42910 MIRROR_TOOL_WEAR`

Setting data **not equal** to zero:
The wear values of the tool length components, whose associated axes are mirrored, are also mirrored - by sign inversion.

8.17.2 Wear sign evaluation

Function

Set setting data SD 42920 WEAR_SIGN_CUTPOS und SD 42930 WEAR_SIGN not equal to zero can be used to invert the sign evaluation of the wear components.

Parameters

`SD 42920 WEAR_SIGN_CUTPOS`

Setting data **not equal** to zero:
In the case of tools with a relevant tool point direction (turning and grinding tools, tool types 400), the sign evaluation of the wear components depends on the tool point direction in the machining plane. This setting data has no effect with tool types, which do not have a relevant tool point direction.

In the following table, the dimensions whose sign is inverted by SD 42920 (not equal to 0) are marked with an X:

Length of cutting edge	Length 1	Length 2
1		
2		X
3	X	X
4	X	
5		
6		
7		X
8	X	
9		

Note

The sign settings of SD 42920 and 42910 are independent. For example, if the sign of a dimensional parameter is changed by both setting data, the resulting sign is unchanged.

`SD 42930 WEAR_SIGN`

Setting data **not equal to** zero:
the sign of all wear dimensions is inverted. This affects both the tool length and other variables such as tool radius, rounding radius, etc.

If a positive wear dimension is entered, the tool becomes "shorter" and "thinner", see chapter "Tool offsets, Special handling, Activation of changed setting data".

8.17.3 Coordinate system of the active machining operation (TOWSTD/TOWMCS/TOWWCS/TOWBCS/TOWTCS/TOWKCS)

Function

Depending on the kinematics of the machine or the availability of an orientable toolholder, the wear values measured in one of these coordinate systems are converted or transformed to a suitable coordinate system.

Programming

```
TOWSTD
```

Or

```
TOWMCS
```

Or

```
TOWWCS
```

Or

```
TOWBCS
```

Or

```
TOWTCS
```

Or

```
TOWKCS
```

Parameters

Coordinate systems of active machining operation

The following coordinate systems can produce tool length offsets that can be used to incorporate the tool length component "wear" into an active tool via the corresponding G code of Group 56.

1. Machine coordinate system (MCS)
2. Basic coordinate system (BCS)
3. Workpiece coordinate system (WCS)
4. Tool coordinate system (TCS)
5. Tool coordinate system of kinematic transformation (KCS)

TOWSTD	Initial setting value for offsets in tool length wear value
TOWMCS	Offsets in tool length in MCS
TOWWCS	Offsets in tool length in WCS
TOWBCS	Offsets in tool length in BCS
TOWTCS	Offsets of tool length at toolholder reference point (orientable toolholder)
TOWKCS	Offsets of tool length at tool head (kinematic transformation)

Description

The most important distinguishing features are shown in the following table:

G code	Wear value	Active orientable toolholder
TOWSTD	Initial value, tool length	Wear values are subject to rotation
TOWMCS	Wear value in MCS. TOWMCS is identical to TOWSTD if no orientable toolholder is active.	Only the vector of the resultant tool length rotates. Wear is ignored
TOWWCS	The wear value is converted to the MCS in the WCS	The tool vector is calculated as with TOWMCS, i.e., the wear value is ignored
TOWBCS	The wear value is converted to the MCS in the BCS	The tool vector is calculated as with TOWMCS, i.e., the wear value is ignored
TOWTCS	The wear value is converted to the MCS in the tool coordinate system	The tool vector is calculated as with TOWMCS, i.e., the wear value is ignored

TOWWCS, TOWBCS, TOWTCS: The wear vector is added to the tool vector.

Linear transformation

The tool length can be defined meaningfully in the MCS only if the MCS is generated by linear transformation from the BCS.

Non-linear transformation

If, for example, a non-linear transformation is active with TRANSMIT, the BCS is automatically applied when the MCS is specified as the chosen coordinate system.

No kinematic transformation and no orientable toolholder

If neither a kinematic transformation nor an orientable toolholder is active, then all the other four coordinate systems (except for the WCS) are combined. It is then only the WCS, which is different to the other systems. Since only tool lengths need to be evaluated, translations between the coordinate systems are irrelevant.

References

For further information about tool offsets, please see:

/FB1/, Description of Functions, Basic Machine (Part 1), Tool Compensation (W1)

Inclusion of wear values in calculation

The setting data SD 42935 WEAR_TRANSFORM defines, which of the following three wear components

1. Wear
2. Total offsets fine
3. Total offsets coarse

are to be made subject to a rotation by way of an adapter transformation or orientable toolholder if one of the following G codes is active.

- TOWSTD Initial setting
  for offsets in tool length
- TOWMCS Wear values
  in the machine coordinate system (MCS)
- TOWWCS Wear values
  in workpiece coordinate system (WCS)
- TOWBCS Wear values (BCS)
  in basic coordinate system
- TOWTCS Wear values in tool coordinate system at the toolholder (T toolholder reference)
- TOWKCS Wear values in coordinate system of tool head with kinematic transformation

Note

Evaluation of individual wear components (assignment to geometry axes, sign evaluation) is influenced by the

- Active plane

- Adapter transformation and

- The following setting data:
SD 42910: MIRROW_TOOL_WEAR
SD 42920: WEAR_SIGN_CUTPOS
SD 42930: WEAR_SIGN
SD 42940: TOOL_LENGTH_CONST
SD 42950: TOOL_LENGTH_TYPE

8.17.4 Tool length and plane change

Function

With the set setting data SD 42940 TOOL_LENGTH_CONST not equal to zero, you can assign tool length components such as length, wear and base dimension to the geometry axes for turning and grinding tools at a plane change.

Parameters

`SD 42940 TOOL_LENGTH_CONST`

Setting data **not equal** to zero:
The assignment between the tool length components (length, wear and tool base dimension) and the geometry axes is not modified when the machining plane is changed (G17-19).

The following table shows the assignment between the tool length components and the geometry axes for turning and grinding tools (tool types 400 to 599):

Table of Contents	Length 1	Length 2	Length 3
17	Y	X	Z
*)	X	Z	Y
19	Z	Y	X
-17	X	Y	Z
-18	Z	X	Y
-19	Y	Z	X

*) Each value not equal to 0 which is not equal to one of the six listed values is evaluated as value 18.

The following table shows the assignment between the tool length components and the geometry axes for all other tools (tool types < 400 or > 599):

Machining plane	Length 1	Length 2	Length 3
*)	Z	Y	X
18	Y	X	Z
19	X	Z	Y
-17	Z	X	Y
-18	Y	Z	X
-19	X	Y	Z

*) Each value not equal to 0 which is not equal to one of the six listed values is evaluated as value 17.

Further information

The representation in the tables assumes that the geometry axes to 3 are labeled X, Y, Z. The axis order and not the axis identifier determines the assignment between an offset and an axis.

8.18 Tools with a relevant cutting edge length

Function

In the case of tools with a relevant tool point direction (turning and grinding tools – tool types 400–599; see chapter "Sign evaluation wear"), a change from G40 to G41/G42 or vice-versa is treated as a tool change. If a transformation is active (e.g., TRANSMIT), this leads to a preprocessor stop (decoding stop) and hence possibly to deviations from the intended part contour.

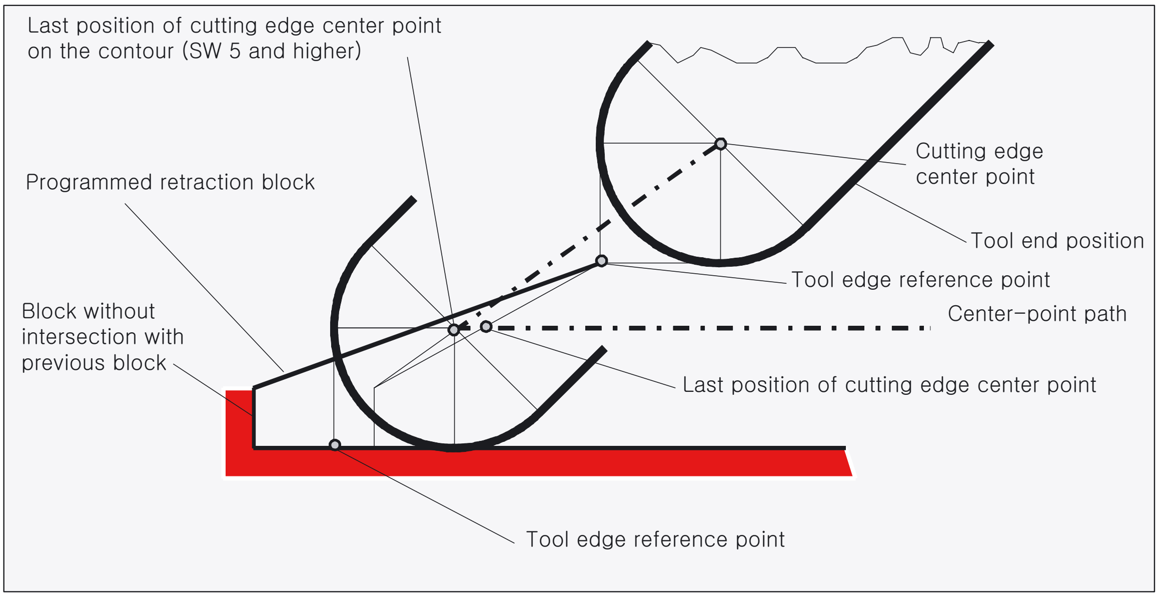
This original functionality changes with regard to:

1. Preprocessor stop on TRANSMIT
2. Calculation of intersection points at approach and retraction with KONT
3. Tool change with active tool radius compensation
4. Tool radius compensation with variable tool orientation at transformation

Description

This original functionality was changed as follows:

- A change from G40 to G41/G42 and vice-versa is no longer treated as a tool change. A preprocessor stop therefore no longer occurs with Transmit.
- 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.
  On approach and retraction with KONT (tool circumnavigates the contour point, see above subsection "Contour approach and retraction"), superimposition takes place in the linear part block of the approach or retraction motion. The geometric conditions are therefore identical for tools with and without a relevant tool point direction. Differences from the previous behavior occur only in relatively rare situations where the approach or retraction block does not intersect with an adjacent motion block, see diagram below.


Last position of cutting edge center point on the contour (SW 5 and higher)
Programmed retraction block
Block without intersection with previous block
Cutting edge center point
Tool end position
Tool edge reference point
Center−point path
Last position of cutting edge center point
Tool edge reference point

- 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 tool point direction 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 tool point direction 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 tool point direction parameter is ignored).

Special functions 9

9.1 Auxiliary function outputs

Function

The auxiliary function output sends information to the PLC indicating when the NC program needs the PLC to perform specific switching operations on the machine tool. The auxiliary functions are output, together with their parameters, to the PLC interface. The values and signals must be processed by the PLC user program.

Function outputs

The following functions can be downloaded to the programmable controller:

- Select tool T
- Tool compensation D, DL
- Feed F/FA
- Spindle speed S
- H functions
- M functions

For the above-mentioned functions it is possible to define whether they are to be transferred during the machining sequence, and which reactions are to be activated.

For each function group or individual function, machine data define whether the output is initiated

- Before the traversing movement,
- With the traversing movement, or
- After the traversing movement.

The PLC can be programmed to acknowledge auxiliary function outputs in various ways.

Programming

Letter[address extension]=Value

The letters, which can be used for auxiliary functions, are:

`M`

Or

`S`

Or

`H`

Or

`T`

Or

`D`

Or

`DL`

Or

`F`

Parameters

In the following table you will find information about the meaning and value ranges for the address extension and the value in auxiliary function outputs. The maximum number of auxiliary functions of the same type per block is also specified.

Overview of auxiliary functions, programming							
Function	Address extension (integer)		Value			Explanation	Number per block
	Meaning	Area	Area	Type	Meaning		
M	-	implicit 0	0 - 99	INT	Function	The address extension is 0 for the value range between 00 and 99. M0, M1, M2, M17, M30 must be used without an address extension.	5
	Spindle no.	1 - 12	1 - 99		Function	M3, M4, M5, M19, M70 with address extension spindle no., e.g., M5 for spindle 2: M2=5. Master spindle is used if no spindle is specified.	
	Any	0 - 99	100 - (max. INT value)		Function	User M function	

S	Spindle no.	1 - 12	0 - ±3.4028 ex 38	REAL	Spindle speed	Without spindle no. for master spindle	3
H	Any	0 - 99	±(max. INT value) ±3.4028 ex 38	INT REAL	Any	Functions have no effect in the NCK; only to be implemented on the PLC	3
T	Spindle no. (for active tool manage ment)	1 - 12	0 - 32000 (or tool names with active tool managem ent)	INT	Tool selection	Tool names are not passed to the PLC interface.	1
D			0 - 9	INT	Tool offset selection	D0 selection, default D1	1
DL	Location-depende nt offset	1 - 6	±3.4028 ex 38	REAL	See tool fine offset selection /FBW/	Refers to previously selected D number	1
F	Path feedrate	0	0.001 - 999 999,999	REAL	Path feedrates		6
(FA)	Axis No.	1 - 31	0.001 - 999 999,999		Axis feedrates		

The highest number for a type specified in the table must not be exceeded.

Example

```
M=QU(…)
H=QU(…)
```

```
N10 H=QU(735)                                    ;High-speed output for H735
N10 G1 F300 X10 Y20 G64
N20 X8 Y90 M=QU(7)
```

M7 was programmed as a high-speed output, so continuous-path mode (G64) is not interrupted.

Note

You should only use this function in individual cases, because it can affect the time synchronization as a result of interaction with other function outputs.

Description

Number of function outputs per NC block

Up to 10 function outputs can be programmed in one NC block. Auxiliary functions can also be output from the action component of **synchronized actions**. See /FBSY/.

Grouping

The functions described can be grouped together. Group assignment is predefined for some M commands. The acknowledgment behavior can be defined by the grouping.

Acknowledgements, high-speed function outputs, QU

Functions, which have not been programmed as high-speed outputs, can be defined as high-speed outputs for individual outputs with the keyword QU. Program execution continues without waiting for the acknowledgment of the miscellaneous function (the program waits for the transport acknowledgment). This helps avoid unnecessary hold points and interruptions to traversing movements.

Machine manufacturer

The appropriate MD must be set for "high-speed function outputs", see /FB1/ Description of Functions, H2, "Auxiliary Function Output"

Function outputs for travel commands

Time is needed to transfer information and wait for a corresponding response, and this has an impact on the travel movements.

High-speed acknowledgment without block change delay

Block change behavior can be influenced by machine data. When the "without block change delay" setting is selected, the system response with respect to high-speed auxiliary functions is as follows:

Auxiliary function output	Response
Before the movement	The block transition between blocks with highspeed auxiliary functions occurs **without** interruption and **without** a reduction in velocity. The auxiliary function output takes place in the first interpolation cycle of the block. The following block is executed with no acknowledgment delay.
During the movement	The block transition between blocks with highspeed auxiliary functions occurs **without** interruption and **without** a reduction in velocity. The auxiliary function output takes place during the block. The following block is executed with no acknowledgment delay.
After the movement	The movement stops at the end of the block. The auxiliary function output takes place at the end of the block. The following block is executed with no acknowledgment delay.

Caution

Function outputs in continuous-path mode

Function outputs **before** the traversing movements interrupt continuous-path mode (G64/G641) and generate an exact stop for the previous block.

Function outputs **after** the traversing movements interrupt continuous-path mode (G64/G641) and generate an exact stop for the current block.

Important: A wait for an outstanding acknowledgment signal from the PLC can also cause an interruption to continuous-path mode, e.g., M instruction sequences in blocks with extremely short path lengths.

9.1.1 M functions

Function

M functions can trigger, for example, switching operations such as "Coolant ON/OFF" and other operations on the machine. Permanent functions have already been assigned to some of the M functions by the control manufacturer (see list of predefined M functions).

Programming

`M...` Possible values. 0 to 9999 9999 (max. INT value), integer

Parameters

List of predefined M functions

M0*	Programmed stop 1
M1*	Optional stop
M2*	End of main program with return to beginning of program
M30*	End of program, same effect as M2
M17*	End of subprogram
M3	Spindle clockwise
M4	Spindle counterclockwise
M5	Spindle stop
M6	Tool change (default setting)
M70	Spindle is switched to axis mode
M40	Automatic gear change
M41	Gear stage 1
M42	Gear stage 2
M43	Gear stage 3
M44	Gear stage 4
M45	Gear stage 5

Notice

Extended address notation cannot be used for the functions marked with *.

The commands M0, M1, M2, M17 and M30 are always initiated **after** the traversing movement.

Machine manufacturer

All free M function numbers can be assigned by the machine manufacturer, e.g., with switching functions for controlling clamping fixtures or for activating/deactivating other machine functions, etc.

See machine manufacturer's specifications.

Example

```
N10 S...
N20 X... M3                              ;M function in the block with axis
                                         ;movement, spindle accelerates before
                                         ;the X axis movement
N180 M789 M1767 M100 M102 M376           ;Max. of 5 M functions in the block
```

Predefined M commands

Certain important M functions for program execution are supplied as standard with the control:

Programmed stop, M0

Machining stops in the NC block with M0. You can now, for example, remove swarf, remeasure, etc.

Programmed stop 1 - optional stop, M1

M1 can be set with

- HMI/dialog "Program Control" or
- the VDI interface.

Program execution on the NC is stopped at each of the programmed blocks.

Programmed stop 2 - an auxiliary function associated with M1 with stop in program execution

Programmed halt 2 can be set via HMI/dialog "Program control" and permits technological processes to be interrupted at any time when the part has finished machining. It therefore allows the operator to intervene in the production process in order to remove chips, for example.

End of program, M2, M17, M30

A program is terminated with M2, M17 or M30 and reset to the beginning of the program. If the main program is called from another program (as a subprogram), M2/M30 has the same effect as M17 and vice versa, i.e., M17 has the same effect in the main program as M2/M30.

Spindle functions, M3, M4, M5, M19, M70

The extended address notation with spindle number is used for all spindle functions.

Example: M2=3 means CW spindle rotation for the second spindle. If no address extension is programmed, the function applies to the master spindle.

9.1.2 H functions

Function

H functions are used to transfer information to the PLC (programmable logic controller), in order to activate specific switching operations. H functions are REAL values.

Up to three H functions can be programmed in one NC block.

Machine manufacturer

The meaning of the functions is determined by the manufacturer.

Programming

```
N10 G0 X20 Y50 H3=–11.3
```

Arithmetic Parameters and Program Jumps 10

10.1 Arithmetic parameter (R)

Function

The arithmetic parameters are used, for example, if an NC program is not only to be valid for values assigned once, or if you need to calculate values. The required values can be set or calculated by the control during program execution. Another possibility consists of setting the arithmetic parameter values by operator inputs. If values have been assigned to the arithmetic parameters, they can be assigned to other variable-setting NC addresses in the program.

Programming

```
Rn=...
```

Parameters

R	Arithmetic variables
n	Number of the arithmetic parameter, n= 0 to max. See machine data or machine manufacturer for max.; default setting: max = 0-99
Range of values	±(0.000 0001 ... 9999 9999) (8 decimal places and leading sign and decimal point) can be assigned to the arithmetic parameters.

Machine manufacturer

The number of R parameters is set in the machine data, or see specifications of machine manufacturer.

Example of R parameters

N10 R1= R1+1	;The new R1 is calculated from the ;old R1 plus 1
N20 R1=R2+R3 R4=R5-R6 R7=R8* R9 R10=R11/R12	
N30 R13=SIN(25.3)	;R13 equals sine of 25.3 degrees
N40 R14=R1*R2+R3	;Multiplication or division takes ;precedence over addition or ;subtraction R14=(R1*R2)+R3
N50 R14=R3+R2*R1	;Result, the same as block N40
N60 R15=SQRT(R1*R1+R2*R2)	;Meaning: R15=square root of $R1^2+R2^2$

Example of assignment of axis values

```
N10 G1 G91 X=R1 Z=R2 F300
N20 Z=R3
N30 X=-R4
N40 Z=-R5
...
```

Value assignment

You can assign values in the following range to the arithmetic parameters:

±(0.000 0001 ... 9999 9999) (8 decimal places and sign and decimal point)

- The decimal point can be omitted for integer values.
- A plus sign can always be omitted.

Example:

R0=3.5678 R1=-37.3 R2=2 R3=-7

R4=-45678.1234

It is possible to assign an extended numerical range by using exponential notation:

Example:

± (10^{-300} ... 10^{+300})

The value of the exponent is written after the EX characters; maximum total number of characters: 10 (including signs and decimal point)

Range of values for EX: -300 to +300

Example:

R0=-0.1EX-5 ;Meaning: R0 = -0.000 001

R1=1.874EX8 ;Meaning: R1 = 187 400 000

Note

There can be several assignments in one block incl. assignments of arithmetic expressions.

Value assignment must be in a separate block.

Assignments to other addresses

The flexibility of an NC program comes down to being able to assign these arithmetic parameters or expressions with arithmetic parameters to other NC addresses. Values, arithmetic expressions and arithmetic parameters can be assigned to all addresses; Exception: addresses N, G, and L.

When assigning, write the character " = " after the address character. It is also possible to have an assignment with a minus sign. A separate block is required for assignments to axis addresses (traversing instructions).

Example:

N10 G0 X=R2 ;Assignment to the X axis

Arithmetic operations and functions

When operators/arithmetic functions are used, it is necessary to use conventional mathematical notation. Machining priorities are set by parentheses. Otherwise, multiplication and division take precedence over addition and subtraction. Degrees are used for the trigonometrical functions.

10.2 Unconditional program jumps

Function

By default, main programs, subprograms, cycles and interrupt routines execute the blocks in the sequence in which they were programmed. Program jumps can be used to modify this sequence.

Programming

`GOTOB` <destination>

`GOTOF` <destination>

`GOTO/GOTOC` <destination variable>

Parameter

GOTOB	"Jump instruction" with backward jump destination (towards beginning of program)
GOTOF	Jump instruction with forward jump destination (towards end of program)
GOTO	Jump instruction with forward, then backward destination search (first towards end of program and then towards beginning of program)
GOTOC	Suppress Alarm 14080 "Branch destination not found". Jump instruction with forward, then backward destination search (first towards end of program and then towards beginning of program)
<destination>	Branch destination parameters for label, block number, or string variable
Label	Destination for a jump instruction
Label:	Labeling of branch destination within the program
Block number	Branch destination as main block or subblock number (e.g., 200, N300)
String variable	Variable of type string containing a label or block number

Example

Axis U: Pallet storage, transporting the pallet to the working area

Axis V: Transfer line to a measuring station, where sampling controls are carried out:

```
N10 …
N20 GOTOF LABEL_0                         ;Jump forward to LABEL_0
N30 …
N40 LABEL_1: R1=R2+R3                     ;Branch destination LABEL_1
N50 …
N60 LABEL_0:                              ;Branch destination LABEL_0
N70 …
N80 GOTOB LABEL_1                         ;Jump backwards to LABEL_1
N90 …
```

Description

Jump destinations with user-defined names can be programmed in a routine. The command GOTOF or GOTOB can be used to branch to a jump destination from any other point within the same program. The program then resumes execution at the instruction immediately following the jump destination.

Destination not found

If the destination is not found, program execution is terminated with Alarm 14080 "Destination not found". Command GOTOC suppresses this alarm. Program execution is resumed at the line following the GOTOC command.

Destination backward

Jump with label

```
Label_1: ;destination
....
GOTOB Label_1
```

Jump forward

Jump with block number

```
GOTOF N100
....
N100 ;destination
```

Indirect jumps

Jump to block number

```
N5 R10=100
N10 GOTOF "N"<<R10 ;jump to block whose number is in R10
N90
N100 ;destination
N110
```

Jump to labels

```
DEF STRING[20] DESTINATION
DESTINATION = "label2" ;jump with variable jump destination
GOTOF DESTINATION
Label1: T="Drill1"
....
Label2: T="drill2" ;destination
```

Further information:

The unconditional jump must be programmed in a separate block.

In programs with unconditional jumps, the end of program M2/M30 does not have to appear at the end of the program.

10.3 Conditional program jumps (IF, GOTOB, GOTOF, GOTO, GOTOC)

Function

Jump conditions can be formulated using IF instructions. The jump to the programmed jump destination is only performed if the jump condition is fulfilled.

Programming

`IF expression GOTOB` <destination>

Or

`IF expression GOTOF` <destination>

Or

`IF expression GOTO/GOTOC` <destination>

Parameters

IF	Keyword for condition
GOTOB	"Jump instruction" with backward jump destination (towards beginning of program)
GOTOF	Jump instruction with forward jump destination (toward program end)
GOTO	Jump instruction with destination search first forward then backward (first toward end of program and then toward beginning of program)
GOTOC	Suppress Alarm 14080 "Destination not found". Jump instruction with destination search first forward then backward (first toward end of program and then toward beginning of program)
<destination>	Destination parameters for label, block number, or string variable
Label	Destination for a jump command
Label:	Labeling of destination within the program
Block number	Jump destination as main block or subblock number (e.g., 200, N300)
String variable	Variable of type string containing a label or block number

Comparison and logical operands

The jump condition can be programmed with any comparison or logic operation (result: TRUE or FALSE). The program jump is executed if the result of the operation is TRUE.

The jump destination can only be a block with a label or block number that appears within the program.

Note

Several conditional jumps can be formulated in the same block.

==	Equal to
<>	Not equal to
>	Greater than
<	Less than
>=	Greater than or equal to
<=	Less than or equal to

Note

For more information, see
/PGA/Programming Guide Advanced, chapter "Flexible NC Programming".

Example

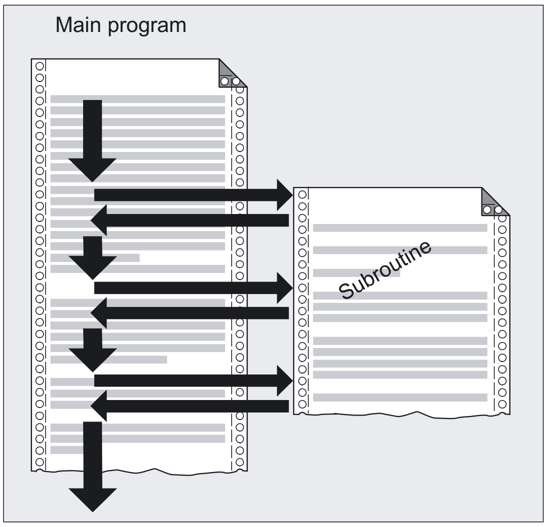
```
N40 R1=30 R2=60 R3=10 R4=11 R5=50 R6=20           ;Assignment of initial values
N41 MA1: G0 X=R2*COS(R1)+R5 ->                    ;Calculation and assignment to
-> Y=R2*SIN(R1)+R6                                ;axis address
N42 R1=R1+R3 R4=R4-1                              ;Specification of variable
N43 IF R4>0 GOTOB MA1                             ;Jump instruction with label
N44 M30                                           ;End of program
```

Subprograms and Repetition of Program Sections 11

11.1 Use of subprograms

Function

In principle, a subprogram has the same structure as a parts program. It consists of NC blocks with traversing and switching commands. Basically, there is no difference between a main program and a subprogram. The subprogram contains either machining operations or sequences of operations that are to be performed several times.



Parameters

M17	End of subprogram, return to the calling program.
RET	Subprogram end if continuous-path mode should **not** be interrupted by retraction. RET must be programmed in a separate block.
M30	The program end M30 set in the main program is evaluated like M17 in the subprogram. This, therefore, enables a main program to be called as a subprogram.

Note

Observe functional differences of the machine, such as spindle stop at M30!

Example

```
N10 POCKET1
```

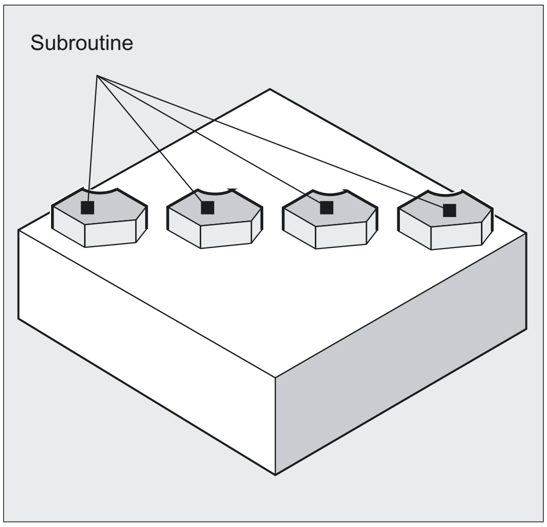
It is also possible to use the address word L... in subprograms The value can have 7 decimal places (integers only). Note: With address L, leading zeros are meaningful for differentiation.

```
N10 L123
N20 L0123
N30 L00123
```

There are three different subprograms in this example.

Use of subprograms

Recurring machining sequences are programmed once in a subprogram. Examples include certain recurring contour shapes and machining cycles. The subprogram can be called and executed in any main program.



Structure of a subprogram

The structure of the subprogram is identical to that of the main program (see chapter on structure and content of an NC program). Subprograms are provided with M17 end of program. This means a return to the program level where the subprogram was called from. It is possible to suppress this end of program M17 in the machine data (e.g., to achieve a better running time).

Note

A program header with parameter definitions can also be programmed in the subprogram. You will find a more detailed description in the Programming Guide "Advanced".

End of subprogram with RET

The instruction RET can also be used in subprograms as a substitute for the backward jump with M17. RET must be programmed in a separate block. The RET statement is used when G64 (G641, G642, G643) continuous-path mode is not to be interrupted by a return. This is only possible if the subprogram has no SAVE attribute.

If M17 is programmed in a separate block, G64 is interrupted and an exact stop generated. Remedy: Do not write M17 in a subprogram block on its own, instead use it with, for example, a traverse path: G1 X=YY M17. The following must be set in the machine data: "No M17 from PLC".

Subprogram name

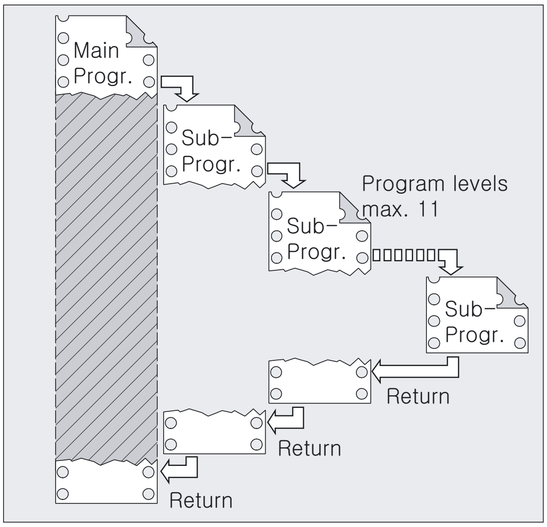
The subprogram is given a name allowing it to be selected from among the others. The name can be freely chosen during program creation, taking the following conventions into account:

- The first two characters must be letters.
- The others can be letters, numerals or underscore characters.
- A maximum of 31 characters can be used.
- No separators are to be used (see the section on "Language elements of the programming language").

The same rules apply as to the names of main programs.

Nesting depth

Subprograms can also be called from a subprogram, not only from a main program. In total, up to 12 program levels are available for this type of nested call, including the main program level. This means: Up to 11 nested subprogram calls can be issued from the main program.



Note

If you are working with SIEMENS machining and measuring cycles, then three levels are required. If a cycle is to be called from a subprogram, this call cannot be issued from beyond level 9.

11.2 Subprogram call

Function

You call the subprogram in the main program either with address L and the subprogram number or by specifying the subprogram name.

Example

```
...
N120 L100                                       ;Subprogram call
                                                ;"L100.SPF":
                                                N10 MSG (DIN subprogram")
                                                N20 G1 G91...
                                                ...
                                                N60 M17 ;End of subprogram

N160 M30                                        ;End of main program
```

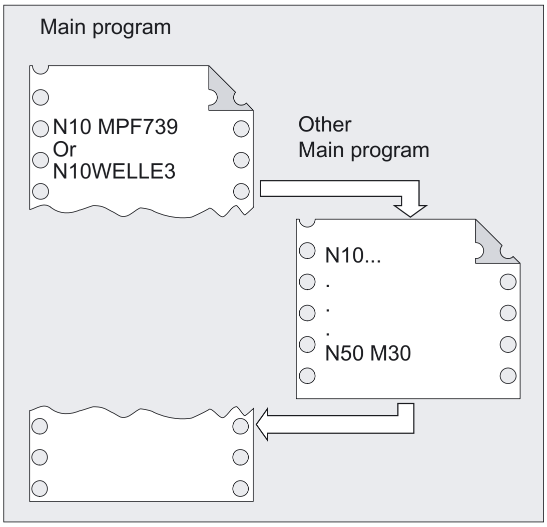
Example of R parameter transfer

```
N10 G0 X0 Y0 G90 T1                             ;Tool T1 in rapid traverse to the
                                                ;first position, absolute dimension
N20 R10=10 R11=20                               ;Describe arithmetic parameters R10
                                                ;and R11
N30 RECTANGLE                                   ;Call the rectangle subprogram
                                                ;"RECTANGLE.SPF" with
                                                ;transfer of R parameters:
                                                N15 G1 X=R10 G91 F500
                                                N25 Y=R11
                                                N35 X=-R10
                                                N45 Y=-R11
                                                N55 M17 ;End of subprogram


N40 G0 X50 Y50 G90                              ;Position tool at the next
                                                ;machining position
N50 RECTANGLE                                   ;Call the rectangle subprogram
                                                ;"RECTANGLE.SPF" with
                                                ;transfer of R parameters
N60 M30                                         ;End of main program
```

Calling main program as subprogram

A main program can also be called as a subprogram. The end of program set in the main program M30 is evaluated as M17 in this case (end of program with return to calling program). You program the call specifying the program name. Example: N10 MPF739 or N10 WELLE3



A subprogram can also be started as a main program.

Note

Search strategy of the control:

Are there any *_MPF?

Are there any *_SPF?

This means: if the name of the subprogram to be called is identical to the name of the main program, the main program that issues the call is called again. This is generally an undesirable effect and must be avoided by assigning unique names to subprograms and main programs.

Call the subprograms with the INI file

Subprograms that do not require parameter assignment can be called from an initialization file:

Example:

`N10 MYINISUB1` ;Subprogram call without parameters

11.3 Subprogram with program repetition

Function

If a subroutine is to be executed several times in succession, the desired number of program repetitions can be entered at address P in the block with the subroutine call.

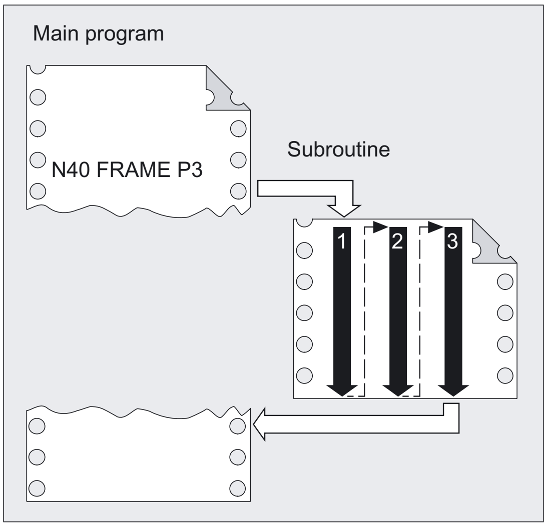
Parameter

P	Number of program runs or program repetitions Range of values P: 1…9999

Example

```
N40 FRAME P3
```

The subprogram FRAME must be executed 3 times in succession.



Caution

The following applies to **every subprogram** call: The subroutine call must always be programmed in a separate NC block.

Subprogram call with program repetition and parameter transfer: Parameters are transferred only when the program is called, i.e. on the first run. The parameters remain unchanged for the remaining repetitions. If you want to change the parameters during program repetitions, you must make the appropriate provision in the subroutine.

11.4 Program section repetition

Function

Instead of using subprograms, you can repeat existing program sections in any order. The block or program sections to be repeated are identified by labels. For more information on labels, please see:

References:
/PG/Programming Guide Fundamentals, chapter "Language elements of the programming language"
/PGA/Programming Guide Advanced, chapter "CASE instruction", "Control structures"

Programming repeat block

```
LABEL: xxx
yyy
REPEATB LABEL P=n
Zzz
```

The program line identified by a label is repeated P=n times. If P is not specified, the program section is repeated exactly once. After the last repetition, the program is continued at the line zzz following the REPEATB line.

The block identified by the label can appear before or after the REPEATB statement. The search initially commences toward the start of the program. If the label is not found in this direction, the search continues toward the end of the program.

Programming repeat area starting at label

```
LABEL: xxx
yyy
REPEAT LABEL P=n
zzz
```

The program section between the label with any name and the REPEAT statement is repeated P=n times. If the block with the label contains further statements, these are executed again on each repetition. If P is not specified, the program section is repeated exactly once. After the last repetition, the program is continued at the line zzz following the REPEAT line.

Note

The label must appear before the REPEAT statement. The search is performed toward the start of the program only.

Programming repeat area between two labels

```
START_LABEL: xxx
ooo
END_LABEL: yyy
ppp
REPEAT START_LABEL END_LABEL P=n
zzz
```

The area between the two labels is repeated P=n times. User-defined names can be assigned to the labels. The first line of the repetition contains the start label, the last line contains the end label. If the line containing the start or end label contains further statements, these are executed again on each run. If P is not specified, the program section is repeated once. After the last repetition, the program is continued at the line zzz following the REPEAT line.

Note

The program section to be repeated can appear before or after the REPEAT statement. The search initially commences toward the start of the program. If the start label is not found in this direction, the search resumes from the REPEAT statement toward the end of the program.

It is not possible to nest the REPEAT statement with the two labels within parentheses. If the start label is found before the REPEAT statement and the end label is not reached before the REPEAT statement, the repetition is performed on the section between the start label and the REPEAT statement.

Programming repeat an area between a label and the end label

```
LABEL: xxx
ooo
ENDLABEL: yyy
REPEAT LABEL P=n
zzz
```

ENDLABEL is a predefined label with a fixed name. ENDLABEL marks the end of a program section and can be used multiple times in the program. The block marked by ENDLABEL can contain further statements.

The area between a label and the following ENDLABEL is repeated P=n times. Any name can be used to define the start label. If the block with the start label or ENDLABEL contains further statements, these are executed on each repetition.

Note

If no ENDLABEL is found between the start label and the block with the REPEAT call, the loop ends before the REPEAT line. The construct therefore has the same effect as described above in "repeat area from label".

If P is not specified, the program section is repeated once.

After the last repetition, the program is continued at the line zzz following the REPEAT line.

Parameters

LABEL:	Jump destination; the name of the jump destination is followed by a colon
REPEAT	Repeat (repeat several lines)
REPEATB	Repeat block (repeat one line only)

Example of repetition of positions

```
N10 POSITION1: X10 Y20
N20 POSITION2: CYCLE(0,,9,8)                  ;Position cycle
N30 ...
N40 REPEATB POSITION1 P=5                     ;Execute BLOCK N10 five times
N50 REPEATB POSITION2                         ;Execute block N20 once
N60 ...
N70 M30
```

Example 5 squares with increasing width are to be produced

```
N5 R10=15
N10 Begin: R10=R10+1                   ;Width
N20 Z=10-R10
N30 G1 X=R10 F200
N40 Y=R10
N50 X=-R10
N60 Y=-R10
N70 Z=10+R10
N80 REPEAT BEGIN P=4                   ;Execute area from N10 to N70 four times
N90 Z10
N100 M30
```

Example repeat program section from BEGIN to END

```
N5 R10=15
N10 Begin: R10=R10+1                    ;Width
N20 Z=10-R10
N30 G1 X=R10 F200
N40 Y=R10
N50 X=-R10
N60 Y=-R10
N70 END: Z=10
N80 Z10
N90 CYCLE(10,20,30)
N100 REPEAT BEGIN END P=3               ;Execute area from N10 to N70 three times
N110 Z10
N120 M30
```

Example of ENDLABEL

```
N10 G1 F300 Z-10
N20 BEGIN1:
N30 X10
N40 Y10
N50 BEGIN2:
N60 X20
N70 Y30
N80 ENDLABEL: Z10
N90 X0 Y0 Z0
N100 Z-10
N110 BEGIN3: X20
N120 Y30
N130 REPEAT BEGIN3 P=3                  ;Execute area from N110 to N120 three times
N140 REPEAT BEGIN2 P=2                  ;Execute area from N50 to N80 twice
N150 M100
N160 REPEAT BEGIN1 P=2                  ;Execute area from N20 to N80 twice
N170 Z10
N180 X0 Y0
N190 M30
```

Example of milling: Machine drill position with different technologies

```
N10 CENTER DRILL()                             ;Load center drill
N20 POS_1:                                     ;Drilling positions 1
N30 X1 Y1
N40 X2
N50 Y2
N60 X3 Y3
N70 ENDLABEL:
N80 POS_2:                                     ;Drilling positions 2
N90 X10 Y5
N100 X9 Y-5
N110 X3 Y3
N120 ENDLABEL:
N130 DRILL()                                   ;Change drill and drilling cycle
N140 THREAD(6)                                 ;Load tap M6 and ;threading cycle
N150 REPEAT POS_1                              ;Repeat program section once from
                                               ;POS_1 up to ENDLABEL
N160 DRILL()                                   ;Change drill and drilling cycle
N170 THREAD(8)                                 ;Load tap M8 and ;threading cycle
N180 REPEAT POS_2                              ;Repeat program section once from
                                               ;POS_2 up to ENDLABEL
N190 M30
```

Supplementary conditions

- Program section repetitions can be nested. Each call uses a subprogram level.
- If M17 or RET is programmed during processing of a program section repetition, the repetition is aborted. The program is resumed at the block following the REPEAT line.
- In the actual program display, the program section repetition is displayed as a separate subprogram level.
- If the level is canceled during the program section repetition, the program resumes at the point after the program section repetition call.

Example:

```
N5 R10=15
N10 Begin: R10=R10+1             ;Width
N20 Z=10-R10
N30 G1 X=R10 F200
N40 Y=R10                        ;Level cancellation
N50 X=-R10
N60 Y=-R10
N70 END: Z10
N80 Z10
N90 CYCLE(10,20,30)
N100 REPEAT BEGIN END P=3
N120 Z10                         ;Resume program processing
N130 M30
```

- Control structures and program section repetitions can be used in combination. There should be no overlap between the two, however. A program section repetition should appear within a control structure branch or a control structure should appear within a program section repetition.
- If jumps and program section repetitions are mixed, the blocks are executed purely sequentially. For example, if a jump is performed from a program section repetition, processing continues until the programmed end of the program section is found.

Example:

```
N10 G1 F300 Z-10
N20 BEGIN1:
N30 X=10
N40 Y=10
N50 GOTOF BEGIN2
N60 ENDLABEL:
N70 BEGIN2:
N80 X20
N90 Y30
N100 ENDLABEL: Z10
N110 X0 Y0 Z0
N120 Z-10
N130 REPEAT BEGIN1 P=2
N140 Z10
N150 X0 Y0
N160 M30
```

Note

Program section repetition is activated by programming.

The REPEAT instruction should be placed behind the traveling blocks.

Tables 12

12.1 List of statements

The list of instructions summarizes all basic programming commands.

Legend:
[1] Default setting at beginning of program (factory settings of the control, if nothing else programmed).
[2] The groups are numbered according to the table in section "List of G functions/preparatory functions"
[3] Absolute end points: modal; incremental end points: non-modal; otherwise modal/non-modal (m, n), depending on syntax of G function.
[4] As arc centers, IPO parameters act incrementally. They can be programmed in absolute mode with AC. The address modification is ignored when the parameters have other meanings (e.g., thread pitch).
[5] The vocabulary word is not valid for SINUMERIK 810D
[6] The vocabulary word is not valid for SINUMERIK 810D/NCU571
[7] The vocabulary word is only valid for SINUMERIK FM-NC
[8] The OEM can add two extra interpolation types. The names can be changed by the OEM.
[9] Extended address notation cannot be used for these functions.

Name	Meaning	Value assign-ment	Description, comment	Syntax	Modal/ non-modal	Group [2]
:	Block number - main block (see N)	0 ... 9999 9999 integers only, without leading signs	special identification of blocks, instead of N... ;this block should contain all instructions required for a complete subsequent machining section	e.g.,20		

A	Axis	Real			m,n [3]	
A2 [5]	Tool orientation: Euler angles	Real			s	
A3 [5]	Tool orientation: Direction vector component	Real			s	
A4 [5]	Tool orientation for start of block	Real			s	
A5 [5]	Tool orientation for end of block; normal vector component	Real			s	
AC	Input of absolute dimensions	0, ..., 359.9999°		X=AC(100)	s	
ACC [5]	Axial acceleration	Real, w/o signs			m	
ACCLIMA [5]	Reduction or overshoot of maximum axial acceleration	0, ..., 200	Valid range is 1 to 200%	ACCLIMA[X]= ...[%]	m	
ACN	Absolute dimensions for rotary axes, approach position in negative direction			A=ACN(...) B=ACN(...) C=ACN(...)	s	
ACP	Absolute dimensions for rotary axes, approach position in positive direction			A=ACP(...) B=ACP(...) C=ACP(...)	s	
ADIS	Rounding clearance for path functions G1, G2, G3, ...	Real, w/o signs			m	
ADISPOS	Approximate distance for rapid traverse G0	Real, w/o signs			m	
ALF	Angle tilt fast	Integer, w/o signs			m	
AMIRROR	Programmable mirroring (additive mirror)			AMIRROR X0 Y0 Z0AMIRROR ;separate block	s	3
ANG	Contour angle				s	
AP	Angle polar	0, ..., ± 360°			m,n [3]	
AR	Opening angle (angle circular)	0, ..., 360°			m,n [3]	
AROT	Programmable rotation (additive rotation)	Rotation about 1st geom. axis: -180º .. 180° 2nd geom. axis: -89.999° ... 90° 3rd geom. axis: -180° .. 180°		AROT X... Y... Z... ;separate AROT RPL= block	s	3

AROTS	Programmable frame rotations with solid angles (additive rotation)			AROTS X... Y... AROTS Z... X... AROTS Y... Z... ;separate AROTS RPL= block	s	3
ASCALE	Programmable scaling (additive scale)			ASCALE X... Y... Z... ;separate block	s	3
ASPLINE	Akima spline				m	1
ATRANS	Additive programmable shift (additive translation)			ATRANS X... Y... Z... ;separate block	s	3
AX	Variable axis identifier	Real			m,n [3]	
AXCTSWE	Advance container axis			AXCTSWE(CT_i)		25
B	Axis	Real			m,n [3]	
B2 [5]	Tool orientation: Euler angles	Real			s	
B3 [5]	Tool orientation: Direction vector component	Real			s	
B4 [5]	Tool orientation for start of block	Real			s	
B5 [5]	Tool orientation for end of block; normal vector component	Real			s	
BAUTO	Definition of first spline segment by the following 3 points (begin not a node)				m	19
BNAT[1]	Natural transition to first spline block (begin natural)				m	19
BRISK[1]	Fast non-smoothed path acceleration				m	21
BRISKA	Switch on brisk path acceleration for the programmed axes					
BSPLINE	B-spline				m	1
BTAN	Tangential transition to first spline block (begin tangential)				m	19
C	Axis	Real			m,n [3]	
C2 [5]	Tool orientation: Euler angles	Real			s	
C3 [5]	Tool orientation: Direction vector component	Real			s	
C4 [5]	Tool orientation for start of block	Real			s	
C5 [5]	Tool orientation for end of block; normal vector component	Real			s	
CDOF [1]	Collision detection OFF				m	23
CDON	Collision detection ON				m	23
CDOF2	Collision detection OFF		For CUT3DC only		m	23
CFC [1]	Constant feed at contour				m	16

CFTCP	Constant feed in tool-center- point (center-point path)				m	16
CFIN	Constant feed at internal radius only, not at external radius				m	16
CHF	Chamfer; value = length of chamfer	Real, w/o signs			S	
CHR	Chamfer; value = width of chamfer in direction of movement (chamfer)					
CHKDNO	Check for unique D numbers					
CIP	Circular interpolation through intermediate point			CIP X... Y... Z... I1=... J1=... K1=...	m	1
COMPOF[1,6]	Compressor OFF				m	30
COMPON[6]	Compressor ON				m	30
COMPCURV	Compressor ON: Polynomials with constant curvature				m	30
COMPCAD	Compressor ON: Surface quality CAD program				m	30
CP	continuous path; path motion				m	49
CPRECOF[1,6]	Programmable contour precision OFF				m	39
CPRECON [6]	Programmable contour precision ON				m	39
CR	Circle radius	Real, w/o signs			s	
CROTS	Programmable frame rotations with solid angles (rotation in the specified axes)			CROTS X... Y... CROTS Z... X... CROTS Y... Z... ;separate CROTS RPL= block	s	
CSPLINE	Cubic spline				m	1
CT	Circle with tangential transition			CT X... Y.... Z...	m	1
CUT2D [1]	2½D cutter compensation (cutter compensation type 2 dimensional)				m	22
CUT2DF	2½D tool offset (cutter compensation type 2 dimensional frame); The tool offset acts in relation to the current frame (inclined plane)				m	22
CUT3DC [5]	3D cutter compensation type 3-dimensional circumference milling				m	22
CUT3DCC [5]	Cutter compensation type 3-dimensional circumference milling with limit surfaces				m	22
CUT3DCCD [5]	Cutter compensation type 3-dimensional circumference milling with limit surfaces with differential tool				m	22
CUT3DF [5]	3D cutter compensation type 3-dimensional face milling				m	22
CUT3DFF [5]	3D cutter compensation type 3-dimensional face milling with constant tool orientation				m	22

	dependent on the current frame					
CUT3DFS [5]	3D cutter compensation type 3-dimensional face milling with constant tool orientation independent of the current frame				m	22
CUTCONOF[1]	Constant radius compensation OFF				m	40
CUTCONON	Constant radius compensation ON				m	40
D	Tool offset number	1, ... 32 000	contains offset data for a specific tool T... ;D0 → offset values for a tool	D...		
DC	Absolute dimensions for rotary axes, approach position directly			A=DC(...) B=DC(...) C=DC(...) SPOS=DC(...)	s	
DIAMCYCOF	Radius programming for G90/G91: ON. The G-code of this group that was last active remains active for display		Radius programming of most recent G-code		m	29
DIAMOF[1]	Diameter programming: OFF (Diametral programming OFF)		Radius programming for G90/G91		m	29
DIAMON	Diametral programming: ON (Diametral programming ON)		Diametral programming for G90/G91		m	29
DIAM90	Diametral programming: for G90, radius progr. for G91				m	29
DILF	Length for lift fast				m	
DISC	Transition circle overshoot - radius compensation	0, ..., 100			m	
DISPR	Distance for repositioning	Real, w/o signs			s	
DISR	Distance for repositioning	Real, w/o signs			s	
DITE	Thread run-out path	Real			m	
DITS	Thread run-in path	Real			m	
DL	Total tool offset	INT			m	
DRFOF	Deactivate the handwheel offsets (DRF)				m	
DRIVE[7]	Velocity-dependent path acceleration				m	21
EAUTO	Definition of last spline section by the last 3 points (end not a node)				m	20

DYNFINISH	Dynamic for smooth-finishing		Technology G group	DYNFINISH G1 X10 Y20 Z30 F1000	m	59
DYNNORM	Standard dynamic, as previously			DYNORM G1 X10	m	59
DYNPOS	Dynamic for positioning mode, tapping			DYNPOS G1 X10 Y20 Z30 F...	m	59
DYNROUGH	Dynamic for roughing			DYNROUGH G1 X10 Y20 Z30 F10000	m	59
DYNSEMIFIN	Dynamic for finishing			DYNSEMIFIN G1 X10 Y20 Z30 F2000	m	59
ENAT [1]	Natural transition to next traversing block (end natural)				m	20
ETAN	Tangential transition to next traversing block at spline end (end tangential)				m	20
F	Feed value (in conjunction with G4 the dwell time is also programmed in F)	0.001, ..., 99999.999	Path feedrate tool/workpiece; unit of measurement in mm/min or mm/rev dependent on G94 or G95	F=100 G1 ...		
FA	Axial feed (feed axial)	0.001, ..., 999999.999 mm/min, degrees/ min; 0.001, ..., 39999.9999 inch/min		FA[X]=100	m	
FCUB [6]	Feedrate variable according to cubic spline (feed cubic)		Acts on feed with G93 and G94		m	37
FD	Path feed for handwheel override (feed DRF)	Real, w/o signs			s	
FDA	Axial feed for handwheel override (feed DRF axial)	Real, w/o signs			s	
FENDNORM	Corner deceleration OFF				m	57
FFWOF [1]	Feedforward control OFF (feed forward OFF)				m	24
FFWON	Feedforward control ON (feed forward ON)				m	24
FGREF	Reference radius of rotary axis or path reference factors of orientation axes (vector interpolation)		Reference size effective value		m	
FGROUP	Definition of axis/axes with path feed		F applies to all axes programmed under FGROUP	FGROUP (axis1, [axis2], ...)		

FIFOCTRL	Control of preprocessing buffer				m	4
FL	Speed limit for synchronized axes (feed limit)	Real, w/o signs	The unit set with G93, G94, G95 is applicable (max. rapid traverse)	FL[axis] =…	m	
FLIN [6]	Feed linear variable (feed linear)		Acts on feed with G93 and G94		m	37
FMA	Feed multiple axial	Real, w/o signs			m	
FNORM [1,6]	Feed normal to DIN 66025				m	37
FP	Fixed point: number of fixed points to be approached	Integer, w/o signs		G75 FP=1	s	
FPR	Identification for rotary axis	0.001, ..., 999999.999		FPR (rotary axis)		
FPRAOF	Deactivate revolutional feedrate					
FPRAON	Activate revolutional feedrate					
FRC	Feed for radius and chamfer				s	
FRCM	Feed for radius and chamfer, modal				m	
FTOCOF [1,6]	Online fine tool offset OFF				m	33
FTOCON [6]	Online fine tool offset ON				m	33
FXS	Travel to fixed stop ON	Integer, w/o signs	1 = select, 0 = deselect		m	
FXST	Torque limit for travel to fixed stop (fixed stop torque)	%	parameter optional		m	
FXSW	Monitoring window for travel to fixed stop (fixed stop window)	mm, inch or degrees	parameter optional			

G Functions						
G	G function (preparatory function) The G functions are divided into G groups. Only one G function from one group can be written in one block. A G function can either be modal (until canceled by another function from the same group), or non-modal (only effective for the block it is written in).	Only integer, predefined values		G...		

G0	Linear interpolation with rapid traverse (rapid traverse motion)		Motion commands	G0 X... Z...	m	1
G1 [1]	Linear interpolation with feedrate (linear interpolation)			G1 X... Z... F...	m	1
G2	Circular interpolation clockwise			G2 X... Z... I... K... F... ;Center point and end point G2 X... Z... CR=... F... ;radius and end point G2 AR=... I... K... F... ;opening angle and ;center point G2 AR=... X... Z... F. ;opening angle and ;end point	m	1
G3	Circular interpolation counter-clockwise			G3 ...; otherwise as for G2	m	1
G4	Dwell time preset		Special motion	G4 F... ; Dwell time in seconds or G4 S... ;Dwell time in ;spindle revolution ;separate block	s	2
G5	Oblique plunge-cut grinding		Oblique plunge-cutting		s	2
G7	Compensatory motion during oblique plunge-cut grinding		Start position		s	2
G9	Exact stop - deceleration				s	11
G17 [1]	Selection of working plane X/Y		Infeed direction Z		m	6
G18	Selection of working plane Z/X		Infeed direction Y		m	6
G19	Selection of working plane Y/Z		Infeed direction X		m	6
G25	Lower working area limitation		Value assignment in channel axes	G25 X.. Y.. Z.. ;separate block	s	3
G26	Upper working area limitation			G26 X.. Y.. Z.. ;separate block	s	3
G33	Thread interpolation with constant pitch	0.001, ..., 2000.00 mm/rev	Motion command	G33 Z... K... SF=... ;cylindrical thread G33 X... I... SF=... ;face thread G33 Z... X... K... SF=... ;taper thread ;(bigger in Z-axis path ;than in X-axis) ;G33 Z... X... I... SF=... ;taper thread ;(bigger in X-axis path ;than in Z-axis)	m	1
G34	Linear degressive speed change [mm/rev^2]		Motion command	G34 X.. Y.. Z.. I.. J.. K.. F..	m	1
G35	Linear progressive speed change [mm/rev^2]		Motion command	G35 X.. Y.. Z.. I.. J.. K.. F..	m	1
G40 [1]	Tool radius compensation OFF				m	7
G41	Tool radius compensation left of contour				m	7
G42	Tool radius compensation right of contour				m	7

G53	Suppression of current zero offset (non-modal)	including programmed offsets		s	9
G54	1st settable zero offset			m	8
G55	2nd settable zero offset			m	8
G56	3rd settable zero offset			m	8
G57	4th settable zero offset			m	8
G58	Axial programmable zero offset, absolute			s	3
G59	Axial programmable zero offset, additive			s	3
G60[1]	Exact stop - deceleration			m	10
G62	Corner deceleration at inside corners when tool radius offset is active (G41, G42)	Together with continuous-path mode only	G62 Z... G1	m	57
G63	Tapping with compensating chuck		G63 Z... G1	s	2
G64	Exact stop - continuous-path mode			m	10
G70	Dimension in inches (lengths)			m	13
G71[1]	Metric dimension (lengths)			m	13
G74	Reference point approach		G74 X... Z... ;separate block	s	2
G75	Fixed point approach	Machine axes	G75 FP=.. X1=... Z1=... ;separate block	s	2
G90[1]	Absolute dimensions		G90 X... Y... Z...(...) Y=AC(...) or X=AC Z=AC(...)	m n	14
G91	Incremental dimension input		G91 X... Y... Z... or X=IC(...) Y=IC(...) Z=IC(...)	m n	14
G93	Inverse-time feedrate 1/rpm	Execution of a block: Time	G93 G01 X... F...	m	15
G94[1]	Linear feedrate F in mm/min or inch/min and °/min			m	15
G95	Revolutional feedrate F in mm/rev or inches/rev			m	15
G96	Constant cutting speed (as for G95) ON		G96 S... LIMS=... F...	m	15
G97	Constant cutting speed (as for G95) OFF			m	15
G110	Pole programming relative to the last programmed setpoint position		G110 X.. Y.. Z..	s	3
G111	Polar programming relative to origin of current workpiece coordinate system		G110 X.. Y.. Z..	s	3
G112	Pole programming relative to the last valid pole		G110 X.. Y.. Z..	s	3
G140[1]	SAR approach direction defined by G41/G42			m	43
G141	SAR approach direction to left of contour			m	43
G142	SAR approach direction to right of contour			m	43
G143	SAR approach direction tangent-dependent			m	43
G147	Soft approach with straight line			s	2

G148	Soft retraction with straight line				s	2
G153	Suppress current frames including base frame		Including system frame		s	9
G247	Soft approach with quadrant				s	2
G248	Soft retraction with quadrant				s	2
G290	Switch to SINUMERIK mode ON				m	47
G291	Switch to ISO2/3 mode ON				m	47
G331	Tapping	±0.001, ..., 2000.00 mm/rev	Motion commands		m	1
G332	Retraction (tapping)				m	1
G340 [1]	Spatial approach block (depth and in plane (helix))		Effective during soft approach/ retraction		m	44
G341	Initial infeed on perpendicular axis (z), then approach in plane		Effective during soft approach/ retraction		m	44
G347	Soft approach with semicircle				s	2
G348	Soft retraction with semicircle				s	2
G450 [1]	Transition circle		Corner behavior with tool radius compensation		m	18
G451	Intersection of equidistances				m	18
G460 [1]	Collision monitoring for approach and retraction block ON				m	48
G461	Extend border block with arc if		no intersection in TRC block		m	48
G462	Extend border block with line if				m	48
G500 [1]	Deactivation of all settable frames if G500 does not contain a value				m	8
G505 .... G599	5 ... 99. Settable zero offset				m	8
G601 [1]	Block change at exact stop fine		only effective if G60 active or G9 with programmable transition rounding		m	12
G602	Block change at exact stop coarse				m	12
G603	Block change at IPO - end of block				m	12
G641	Exact stop - continuous-path mode			G641 AIDS=...	m	10
G642	Corner rounding with axial precision				m	10
G643	Block-internal corner rounding				m	10
G644	Corner rounding with specified axis dynamics				m	10
G621	Corner deceleration at all corners		Together with continuous-path mode only	G621 AIDS=...	m	57
G700	Dimension in inches and inch/min (lengths + velocities + system variable)				m	13
G710 [1]	Metric dimension in mm and mm/min (lengths + velocities + system variable)				m	13
G810[1], ..., G819	G group reserved for the OEM					31
G820[1], ..., G829	G group reserved for the OEM					32

G931	Feedrate specified by travel time	Travel time		m	15
G942	Freeze linear feedrate and constant cutting rate or spindle speed			m	15
G952	Freeze revolutional feedrate and constant cutting rate or spindle speed			m	15
G961	Constant cutting speed (as for G94) ON	Feedrate type	G961 S... LIMS=... F...	m	15
G962	Linear or revolutional feedrate and constant cutting rate			m	15
G971	Constant cutting speed (as for G94) OFF	Feedrate type		m	15
G972	Freeze linear or revolutional feedrate and constant spindle speed			m	15
GOTOF	Jump forwards (toward the end of the program)				
GOTOB	Jump backwards (toward the start of the program)				
GWPSOF	Deselect constant grinding wheel peripheral speed (GWPS)		GWPSOF(T No.)	s	
GWPSON	Select constant grinding wheel peripheral speed (GWPS)		GWPSON(T No.)	s	

H...	Auxiliary function output to the PLC	Real/INT progr.: REAL : ±3,4028-ex38 INT: -2147483648 +2147483648 Display: ±999 999 999,9999	settable via machine data (machine manufacturer)	H100 or H2=100		
I [4]	Interpolation parameters	Real			s	
I1	Intermediate point coordinate	Real			s	
IC	Incremental dimensioning	0, ..., ±99999.999°		X=IC(10)	s	

INCW	Travel on a circle involute in CW direction with interpolation of involute by G17/G18/G19	Real	End point: Center point: Radius with CR > 0:	INCW/INCCW X... Y... Z... INCW/INCCW I... J... K... INCW/INCCW CR=... AR...	m	1
INCCW	Travel on a circle involute in CCW direction with interpolation of involute by G17/G18/G19	Real			m	1
ISD	Insertion depth	Real			m	
J[4]	Interpolation parameters	Real			s	
J1	Intermediate point coordinate	Real			s	
JERKLIMA[5]	Reduction or overshoot of maximum jerk (jerk axial)	1, ..., 200	Valid range is 1 to 200%	JERKLIMA[X]= ...[%]	m	
K[4]	Interpolation parameters	Real			s	
K1	Intermediate point coordinate	Real			s	
KONT	Travel round contour on tool offset				m	17
KONTC	Approach/traverse with continuous-curvature polynomial				m	17
KONTT	Approach/traverse with continuous-tangent polynomial				m	17
L	Subprogram number	Integer, up to 7 places		L10	s	
LEAD[5]	Lead angle	Real			m	
LFOF[1]	Interrupt thread cutting OFF				m	41
LFON	Interrupt thread cutting ON				m	41
LFPOS	Axial retraction to a position				m	46
LFTXT[1]	Tangential tool direction on retraction				m	46
LFWP	Non-tangential tool direction on retraction				m	46
LIMS	Spindle speed limitation with G96/G961 and G97 (limit spindle speed)	0.001, ... 99 999. 999			m	
M...	Switching operations	INT Display: 0, ..., 999 999 999 Program: 0;...; 2147483647	Up to 5 unassigned M functions can be assigned by the machine manufacturer			
M0[10]	Programmed stop					
M1[10]	Optional stop					

M2 [10]	End of main program with return to beginning of program					
M3	Direction of spindle rotation clockwise for master spindle					
M4	Direction of spindle rotation counterclockwise for master spindle					
M5	Spindle stop for master spindle					
M6	Tool change					
M17 [9]	End of subprogram					
M19	For SSL accumulated spindle programming					
M30 [9]	End of program, same effect as M2					
M40	Automatic gear change					
M41... M45	Gear stage 1, ..., 5					
M70	Transition to axis mode					
MEAC	Continuous measurement without deleting distance-to-go	Integer, w/o signs			s	
MEAS	Measure with touch-trigger probe	Integer, w/o signs			s	
MEASA	Measurement with deletion of distance-to-go				s	
MEAW	Measure with touch-trigger probe without deleting distance-to-go	Integer, w/o signs			s	
MEAWA	Measurement without deletion of distance-to-go				s	
MIRROR	Mirroring, programmable			MIRROR X0 Y0 Z0 ;separate block	s	3
MOV						
MSG	Programmable messages			MSG("message")	m	
N	Block number - subblock	0, ..., 9999 9999 integers only, without leading signs	Can be used for assigning a number to a block; located at beginning of block	e.g., N20		
NORM [1]	Standard setting in starting point and end point with tool offset				m	17
OEMIPO1[6,8]	OEM interpolation 1				m	1
OEMIPO2[6,8]	OEM interpolation 2				m	1
OFFN	Allowance on the programmed contour			OFFN=5		
OMA1 [6]	OEM address 1	Real			m	
OMA2 [6]	OEM address 2	Real			m	
OMA3 [6]	OEM address 3	Real			m	
OMA4 [6]	OEM address 4	Real			m	
OMA5 [6]	OEM address 5	Real			m	
OFFN	Offset - normal	Real			m	

ORIC[1,6]	Orientation changes at outside corners are superimposed on the circle block to be inserted (orientation change continuously)				m	27
ORID[6]	Orientation changes are performed before the circle block (orientation change discontinuously)				m	27
ORIAXPOS	Orientation angle via virtual orientation axes with rotary axis positions				m	50
ORIEULER	Orientation angle via Euler angle				m	50
ORIAXES	Linear interpolation of machine axes or orientation axes		Final orientation: Vector specification A3, B3, C3, or Euler/RPY angle A2, B2, C2 Additional inputs: Rotational vectors A6, B6, C6 Opening angle of cone in degrees 0 < SLOT <180 degrees Intermediate vectors: A7, B7, C7 contact point of tool: XH, YH, ZH,	Parameter settings as follows: Direction vectors normalized A6=0 B6=0 C6=1 Opening angle implemented as travel angle with SLOT=... SLOT=+... at ≤ 180 degrees SLOT= -... at ≥ 180 degrees Intermediate orientation normalized A7=0 B7=0 C7=1	m	51
ORICONCW	Interpolation on a circular peripheral surface in CW direction				m	51
ORICONCCW	Interpolation on a circular peripheral surface in CCW direction				m	51
ORICONIO	Interpolation on a circular peripheral surface with intermediate orientation setting				m	51
ORICONTO	Interpolation on a circular peripheral surface in tangential transition				m	51
ORICURVE	Interpolation of orientation with specification of motion of two contact points of tool				m	51
ORIPLANE	Interpolation in a plane (corresponds to ORIVECT) large-radius circular interpolation				m	51
ORIPATH	Tool orientation in relation to path		Transformation package handling, see /FB/, TE4		m	51
ORIPATHS	Tool orientation in relation to path, blips in the orientation characteristic are smoothed		In relation to the path as a whole		m	51
ORIROTA	Angle of rotation to an absolute direction of rotation.				m	54
ORIROTC	Tangential rotational vector in relation to path tangent		In relation to path tangent		m	54
ORIROTR	Angle of rotation relative to the plane between the start and end orientation.				m	54
ORIROTT	Angle of rotation relative to the change in the orientation vector.				m	54
ORIRPY	Orientation angle via RPY angle				m	50
ORIS[5]	Orientation modification (orientation smoothing factor)	Real	Referring to the path		m	
ORIVECT	Large-radius circular interpolation (identical to ORIPLANE)				m	51
ORIVIRT1	Orientation angle via virtual orientation axes (definition 1)				m	50
ORIVIRT2	Orientation angle via virtual orientation axes (definition 1)				m	50

ORIMKS [6]	Tool orientation in the machine coordinate system				m	25
ORIWKS [1,6]	Tool orientation in the workpiece coordinate system				m	25
OS	Oscillation on/off	Integer, w/o signs				
OSC [6]	Continuous tool orientation smoothing				m	34
OSCILL	Axis assignment for oscillation-activate oscillation		Axis: 1 - 3 infeed axes		m	
OSCTRL	Oscillation control options	Integer, w/o signs			m	
OSB	Oscillating: Start point				mF	
OSE	Oscillating: End point				m	
OSNSC	Oscillating: Number of spark-out cycles (oscillating: number spark-out cycles)				m	
OSOF [1,6]	Tool orientation smoothing OFF				m	34
OSP1	Oscillating: Left reversal point (oscillating: position 1)	Real			m	
OSP2	Oscillating: Right reversal point (oscillating: position 2)	Real			m	
OSS [6]	Tool orientation smoothing at end of block				m	34
OSSE [6]	Tool orientation smoothing at start and end of block				m	34
OST1	Oscillating: Stopping point in left reversal point	Real			m	
OST2	Oscillating: Stopping point in right reversal point	Real			m	
OVR	Speed override	1, ..., 200%			m	
OVRA	Axial speed override	1, ..., 200%			m	
P	Number of subprogram passes	1, ..., 9999, integers w/o signs		e.g., L781 P... ;separate block		
PAROTOF	Deactivate workpiece-related frame rotation				m	52
PAROT	Align workpiece coordinate system on workpiece				m	52
PDELAYOF [6]	Punch with delay OFF				m	36
PDELAYON [1.6]	Punch with delay ON				m	36

PL	Parameter interval length	Real, w/o signs			s	
POLY [5]	Polynomial interpolation				m	1
PON [6]	Punch ON				m	35
PONS [6]	Punch ON in IPO cycle (punch ON slow)				m	35
POS	Axis positioning			POS[X]=20		
POSA	Position axis across block boundary			POSA[Y]=20		
POLF	LIFTFAST position				m	
PRESETON	Sets the actual value for programmed axes		One axis identifier is programmed at a time, with its respective value in the next parameter. Up to 8 axes possible	PRESETON(X,10,Y,4.5)		
PTP	**p**oint **t**o **p**oint; point to point motion		synchronous axis		m	49
PTPG0	Point to point motion only with G0, otherwise CP		synchronous axis		m	49
PUTFTOC	PutFineToolCorrection: Fine tool offset for parallel dressing					
PUTFTOCF	PutFineToolCorrectionFunctionDependent: Fine tool offset dependent on a function for parallel dressing defined with FCtDEF					
PW	Point weight	Real, w/o signs			s	
R...	Arithmetic parameter also as settable address identifier and with numerical extension	±0.0000001, ..., 9999 9999	Number of R parameters can be set by MD	R10=3 ;R parameter assignment X=R10 ;axis valueR[R10]=6 ;indirect prog.		
REPOSA	Repositioning linear all axes: Linear repositioning with all axes				s	2
REPOSH	Repositioning semicircle: Repositioning in semicircle				s	2
REPOSHA	Repositioning semicircle all axes: Repositioning with all axes; geometry axes in semicircle				s	2
REPOSL	Repositioning linear: Linear repositioning				s	2
REPOSQ	Repositioning quarter-circle: Repositioning in a quadrant				s	2

REPOSQA	Repositioning quarter-circle all axes: Return to contour linear all axes; geometry axes in quarter-circle				s	2
RET	End of subprogram		Use in place of M17 – without function output to PLC	RET		
RMB	Repositioning at beginning of block (Repos mode begin of block)				m	26
RME	Repositioning at end of block (Repos mode end of block)				m	26
RMI [1]	Repositioning at interruption point (Repos mode interrupt)				m	26
RMN	Reapproach to nearest path point (Repos mode of nearest orbital block)				m	26
RND	Round the contour corner	Real, w/o signs		RND=...	s	
RNDM	Modal rounding	Real, w/o signs		RNDM=... RNDM=0: disable modal rounding	m	
ROT	Programmable rotation	Rotation about 1st geom. axis: -180 180° 2nd geom. axis: -89.999°, ..., 90° 3rd geom. axis: -180° .. 180°		ROT X... Y... Z... ROT RPL= ;separate block	s	3
ROTS	Programmable frame rotations with solid angles (rotation)			ROTS X... Y... ROTS Z... X... ROTS Y... Z... ROTS RPL= ;separate block	s	3
RP	Polar radius	Real			m,n [3]	
RPL	Rotation in the plane	Real, w/o signs			s	
RTLION	G0 with linear interpolation				m	55
RTLIOF	G0 without linear interpolation (single-axis interpolation)				m	55

S	Spindle speed or (with G4, G96/G961) other meaning	REAL Display: ±999 999 999.9999 Program: ±3,4028 ex38	Spindle speed in rpm G4: Dwell time in spindle revolutions G96/G961: Cutting rate in m/min.	S...: Speed for master spindle S1...: Speed for spindle 1	m, n	
SCALE	Programmable scaling (scale)			SCALE X... Y... Z... ;separate block	s	3
SD	Spline degree	Integer, w/o signs			s	
SETMS	Reset to the master spindle defined in machine data					
SETMS(n)	Set spindle n as master spindle					
SF	Starting point offset for thread cutting (spline offset)	0.0000,..., 359.999°			m	
SOFT	Soft smoothed path acceleration				m	21
SON [6]	Nibbling ON (stroke ON)				m	35
SONS [6]	Nibbling ON in IPO cycle (stroke ON slow)				m	35
SPATH [1]	Path reference for FGROUP axes is arc length				m	45
SPCOF	Switch master spindle or spindle(s) from speed control to position control			SPCON SPCON (n)		
SPCON	Switch master spindle or spindle(s) from position control to speed control			SPCON SPCON (n)		
SPIF1 [1,6]	Fast NCK inputs/outputs for punching/nibbling byte 1 (stroke/punch interface 1)				m	38
SPIF2 [6]	Fast NCK inputs/outputs for punching/nibbling byte 2 (stroke/punch interface 2)				m	38
SPLINE-PATH	Define spline grouping		Max. of 8 axes			
SPOF [1,6]	Stroke OFF, punching, nibbling OFF				m	35
SPN [6]	Number of path sections per block (stroke/punch number)	Integer			s	
SPP [6]	Length of path section (stroke/punch path)	Integer			m	
SPOS	Spindle position			SPOS=10 or SPOS[n]=10	m	
SPOSA	Spindle position across block boundaries			SPOSA=5 or SPOSA[n]=5	m	
SR	Retraction path (sparking-out retract path)	Real, w/o signs			s	

SRA	Retraction path axial with external input (sparking-out retract)			SRA[Y]=0.2	m	
ST	Sparking-out time	Real, w/o signs			s	
STA	Sparking-out time axial				m	
STAT	Position of joints	Integer			s	
STARTFIFO[1]	Execute; simultaneously fill preprocessing memory				m	4
STOPFIFO	Stop machining; fill preprocessing memory until STARTFIFO is detected, FIFO full or end of program				m	4
SUPA	Suppression of current zero offset, including programmed offsets, system frames, handwheel offsets (DRF), external zero offset and overlaid motion				s	9
T	Call tool (only change if specified in machine data; otherwise M6 command necessary)	1, ..., 32 000	Call via T No.: or via tool identifier:	e.g., T3 or T=3 e.g., T="DRILL"		
TCARR	Request toolholder (number "m")	Integer	m=0: deselect active toolholder	TCARR=1		
TCOABS [1]	Determine tool length components from the current tool orientation		Necessary after reset, e.g., through manual setting		m	42
TCOFR	Determine tool length components from the orientation of the active frame				m	42
TCOFRX	Determine tool orientation of an active frame on selection of tool, tool points in X direction		Tool perpendicular to inclined surface		m	42
TCOFRY	Determine tool orientation of an active frame on selection of tool, tool points in Y direction		Tool perpendicular to inclined surface		m	42
TCOFRZ	Determine tool orientation of an active frame on selection of tool, tool points in Z direction		Tool perpendicular to inclined surface		m	42
TILT [5]	Tilt angle	Real			m	
TMOF	Deselect tool monitoring		T no. is only necessary if the tool with this number is not active.	TMOF (T no.)		
TMON	Activate tool monitoring		T No. = 0: Deactivate monitoring for all tools	TMON (T no.)		

TOFRAME	Set current programmable frame to tool coordinate system		Frame rotation in tool direction		m	53
TOFRAMEX	X axis parallel to tool direction, secondary axis Y, Z				m	53
TOFRAMEY	Y axis parallel to tool direction, secondary axis Z, X				m	53
TOFRAMEZ	Z axis parallel to tool direction, secondary axis X, Y				m	53
TOROTOF	Frame rotations in tool direction OFF				m	53
TOROT	Z axis parallel to tool orientation		Frame rotations ON Rotation component of programmable frame		m	53
TOROTX	X axis parallel to tool orientation				m	53
TOROTY	Y axis parallel to tool orientation				m	53
TOROTZ	Z axis parallel to tool orientation				m	53
TOWSTD	Initial setting value for offsets in tool length		Inclusion of tool wear		m	56
TOWBCS	Wear values in basic coordinate system (BCS)				m	56
TOWKCS	Wear values in the coordinate system of the tool head for kinetic transformation (differs from MCS by tool rotation)				m	56
TOWMCS	Wear data in the machine coordinate system (MCS)				m	56
TOWTCS	Wear values in the tool coordinate system (tool carrier ref. point T at the tool holder)				m	56
TOWWCS	Wear values in workpiece coordinate system (WCS)				m	56
TRAFOOF	Deactivate transformation			TRAFOOF( )		
TRANS	Programmable translation			TRANS X... Y... Z... ;separate block	s	3
TU	Axis angle	Integer		TU=2	s	
TURN	Number of turns for helix	0, ..., 999			s	
UPATH	Path reference for FGROUP axes is curve parameter				m	45
VELOLIMA[5]	Reduction or overshoot of maximum axial velocity	1, ..., 200	Valid range is 1 to 200%	VELOLIMA[X]= ...[%]	m	
WAITM	Wait for marker in specified channel; end previous block with exact stop			WAITM(1,1,2)		
WAITMC	Wait for marker in specified channel; exact stop only if the other channels have not yet reached the marker			WAITMC(1,1,2)		
WAITP	Wait for end of traversing			WAITP(X) ; separate block		
WAITS	Waiting to reach spindle position			WAITS (main spindle) WAITS (n,n,n)		

WALIMOF	Working area limitation OFF			;separate block	m	28
WALIMON[1]	Working area limitation ON			;separate block	m	28
X	Axis	Real			m,n [3]	
Y	Axis	Real			m,n [3]	
Z	Axis	Real			m,n [3]	

Legend:

[1] Default setting at beginning of program (factory settings of the control, if nothing else programmed).

[2] The groups are numbered according to the table in section "List of G functions/preparatory functions"

[3] Absolute end points: modal; incremental end points: non-modal; otherwise modal/non-modal (m, n) depending on syntax of G function.

[4] As arc centers, IPO parameters act incrementally. They can be programmed in absolute mode with AC. The address modification is ignored when the parameters have other meanings (e.g., thread pitch).

[5] The vocabulary word is not valid for SINUMERIK 810D

[6] The vocabulary word is not valid for SINUMERIK 810D/NCU571

[7] The vocabulary word is only valid for SINUMERIK FM-NC

[8] The OEM can add two extra interpolation types. The names can be changed by the OEM.

[9] Extended address notation cannot be used for these functions.

12.2 List of addresses

List of addresses

The list of addresses consists of

- Address letters
- Fixed addresses
- Fixed addresses with axis expansion
- Settable addresses

Address letters

Available address letters

Letter	Meaning	Numeric extension
A	Variable address identifier	x
B	Variable address identifier	x
C	Variable address identifier	x
D	Selection/deselection of tool length compensation, tool cutting edge	
E	Variable address identifier	
F	Feedrate dwell time in seconds	x
G	G function	
H	H function	x
I	Variable address identifier	x
J	Variable address identifier	x
K	Variable address identifier	x
L	Subprograms, subprogram call	
M	M function	x
N	Subblock number	
O	Unassigned	
P	Number of program runs	
Q	Variable address identifier	x
R	Variable identifier (arithmetic parameter)/variable address identifier without numerical Expansion	x
S	Spindle value dwell time in spindle revolutions	x x
T	Tool number	x
V	Variable address identifier	x

W	Variable address identifier	x
X	Variable address identifier	x
Y	Variable address identifier	x
Z	Variable address identifier	x
%	Start character and separator for file transfer	
:	Main block number	
/	Skip identifier	

Available fixed addresses

Address identifier	Address type	Modal/ non-modal	G70/ G71	G700/ G710	G90/ G91	IC	AC	DC, ACN, ACP	CIC, CAC, CDC, CACN, CACP	Qu	Data type
L	Subprogram no.	n									Integer without sign
P	Number of subprogram passes	n									Integer without sign
N	Block number	n									Integer without sign
G	G function	See list of G functions									Integer without sign
F	Feed, dwell time	m, n	x							x	Real without sign
OVR	Override	m									Real without sign
S	Spindle, dwell time	m, n								x	Real without sign
SPOS	Spindle position	m	x	x	x						Real
SPOSA	Spindle position beyond block limits	m	x	x	x						Real
T	Tool number	m								x	Integer without sign
D	Offset number	m								x	Integer without sign
M, H,	Auxiliary functions	n								x	M: Integer without sign H: Real

Fixed addresses with axis expansion

Address identifier	Address type	Modal/ non-modal	G70/ G71	G700/ G710	G90/ G91	IC	AC	DC, ACN, ACP	CIC, CAC, CDC, CACN, CACP	Qu	Data type
AX: Axis	Variable axis identifier	*)	x	x	x	x	x	x			Real
IP: Interpolation parameter	Variable interpolation parameter	n	x	x	x	x	x				Real
POS: Positioning axis	Positioning axis	m	x	x	x	x	x	x	x		Real
POSA: Positioning axis above end of block	Positioning axis across block boundaries	m	x	x	x	x	x	x	x		Real
POSP: Positioning axis in parts	Positioning axis in parts (oscillation)	m	x	x	x	x	x	x			Real: End position/real: partial length integer: Option
PO: Polynomial [1])	Polynomial coefficient	n	x	x							Real without sign 1 - 8 times
FA: Feed axial	Axial feedrate	m	x							x	Real without sign
FL: Feed limit	Axial feed limit	m	x								Real without sign
OVRA: Override	Axial override	m	x								Real without sign
ACC [2]): Axial acceleration	Axial acceleration	m									Real without sign
FMA: Feedrate multiple axial	Synchronous feedrate axial	m	x								Real without sign
STW: Sparking-out time axial	Sparking out time axial	m									Real without sign
SRA: Sparking-out retract	Retraction path on external input axial	m	x	x							Real without sign
OS: Oscillating ON/OFF	Oscillation ON/OFF	m									Integer without sign
OST1: Oscillating	Stopping time at left reversal	m									Real

time 1	point (oscillation)										
OST2: Oscillating time 2	Stopping time at right reversal point (oscillation)	m									Real
OSP1: Oscillating position 1	Left reversal point (oscillation)	m	x	x	x	x	x	x			Real
OSP2: Oscillating position 2	Right reversal point (oscillation)	m	x	x	x	x	x	x			Real
OSB: Oscillating start	position	m	x	x	x	x	x	x			Real
OSE: Oscillating end position	Oscillation end position	m	x	x	x	x	x	x			Real
OSNSC: Oscillating: number spark-out cycles	Number of spark-out cycles (oscillation)	m									Integer without sign
OSCTRL: Oscillating control	Oscillation control options	m									Integer without sign: set options, integer without sign: reset options
OSCILL: Oscillating	Axis assignment for oscillation activate oscillation	m									Axis: 1 - 3 infeed axes
FDA: Feedrate DRF axial	Axis feedrate for handwheel override	n	x								Real without sign
FGREF	Reference radius	m	x	x							Real without sign
POLF	LIFTFAST position	m	x	x							Real without sign
FXS: Fixed stop	Travel to fixed stop ON	m									Integer without sign
FXST: Fixed stop torque	Torque limit for travel to fixed stop	m									Real

FXSW: Fixed stop window	Monitoring window for travel to fixed stop	m									Real

In these addresses, an axis or an expression of axis type is specified in square brackets. The data type in the above column shows the type of value assigned.
*) Absolute end points: modal, incremental end points: non-modal, otherwise modal/non-modal depending on syntax of G function.

1) The vocabulary word is **not** valid for SINUMERIK FM-NC.

2) The vocabulary word is not valid for SINUMERIK FM-NC/810D

Settable addresses

Address identifier	Address type	Modal/ non-modal	G70/ G71	G700/ G710	G90/ G91	IC	AC	DC, ACN, ACP	CIC, CAC, CDC, CACN, CACP	Qu	Max. number	Data type
Axis values and end points												
X, Y, Z, A, B, C	Axis	*)	x	x	x	x	x	x		8		Real
AP: Angle polar	Polar angle	m/n*	x	x	x					1		Real
RP: Polar radius	Polar radius	m/n*	x	x	x	x	x			1		Real without sign
Tool orientation												
A2, B2, C2 1)	Euler angle or RPY angle	n								3		Real
A3, B3, C3 1)	Direction vector component	n								3		Real
A4, B4, C4 for block beginning 1)	Normal vector component	s								3		Real
A5, B5, C5 for end of block 1)	Normal vector component	n								3		Real
A6, B6, C6 standardized vector 1)	Direction vector component	s								3		Real
A7, B7, C7 standardized vector 1)	Intermediat e orientation component	s								3		Real
LEAD: Lead angle 1)	Lead angle	m								1		Real
THETA: third degree of	Angle of rotation,	s			x	x	x			1		Real

freedom Tool orientation¹)	rotation about the tool direction											
TILT: Tilt angle ¹)	Tilt angle	m								1		Real
ORIS:¹) Orientation Smoothing factor	Orientation change (referring to the path)	m								1		Real
Interpolation parameters												
I, J, K** I1, J1, K1	Interpolation parameter Intermediate point coordinate	n n	x x	x x	 x	x** x	x** x			3		Real Real
RPL: Rotation plane	Rotation in the plane	n								1		Real
CR: Circle radius	Circle radius	n	x	x						1		Real without sign
AR: Angle circular	Opening angle									1		Real without sign
TURN	Number of turns for helix	n								1		Integer without sign
PL: Parameter interval length	Parameter interval length	n								1		Real without sign
PW: Point	weight	n								1		Real without sign
SD: Spline	degree	n								1		Integer without sign
TU: Turn	Turn	m										Integer Int
STAT: State	State	m										Integer without sign
SF: Spindle offset	Starting point offset for thread cutting	m								1		Real
DISR: Distance for repositioning	Distance for repositioning	n	x	x						1		Real without sign
DISPR: Distance path for repositioning	Repos path difference	n	x	x						1		Real without sign

ALF: Angle lift fast	Fast retraction angle	m								1		Integer without sign
DILF: Distance lift fast	Fast retraction length	m	x	x						1		Real
FP	Fixed point: Number of fixed point to approach	n								1		Integer without sign
RNDM: Round modal	Modal rounding	m	x	x						1		Real without sign
RND: Round	Non-modal rounding	n	x	x						1		Real without sign
CHF: Chamfer	Chamfer non-modal	n	x	x						1		Real without sign
CHR: Chamfer	Chamfer in initial direction of motion	n	x	x						1		Real without sign
ANG: Angle	Contour angle	n								1		Real
ISD: Insertion depth	Insertion depth	m	x	x						1		Real
DISC: Distance	Transition circle overshoot tool offset	m	x	x						1		Real without sign
OFFN	Offset contour - normal	m	x	x						1		Real
DITS	Thread run-in path	m	x	x						1		Real
DITE	Thread run-out path	m	x	x						1		Real
Nibbling/punching												
SPN: Stroke/Punch Number [2])	Number of path sections per block	n								1		INT
SPP: Stroke/Punch Path [2])	Length of a path section	m								1		Real

Grinding												
ST: Sparking-out time	Sparking-out time	n								1		Real without sign
SR: Sparking-out retract path	Return path	n	x	x						1		Real without sign
Approximate positioning criteria												
ADIS	Rounding clearance	m	x	x						1		Real without sign
ADISPOS	Rounding clearance for rapid traverse	m	x	x						1		Real without sign
Measurement												
MEAS: Measure	Measure with touch-trigger probe	n								1		Integer without sign
MEAW: Measure without deleting distance-to-go	Measure without deleting distance-to-go	n								1		Integer without sign
Axis, spindle behavior												
LIMS: Limit spindle speed	Spindle speed limitation	m								1		Real without sign
Feedrates												
FAD	Speed of the slow infeed motion	n		x						1		Real without sign
FD: Feed DRF	Path feed for handwheel override	n		x						1		Real without sign
FRC	Feed for radius and chamfer	n		x								Real without sign
FRCM	Feed for radius and chamfer, modal	m		x								Real without sign

OEM addresses												
OMA1: OEM address 1 2)	OEM address 1	m				x	x	x		1		Real
OMA2: OEM address 2 2)	OEM address 2	m				x	x	x		1		Real
OMA3: OEM address 3 2)	OEM address 3	m				x	x	x		1		Real
OMA4: OEM address 4 2)	OEM address 4	m				x	x	x		1		Real
OMA5: OEM address 5 2)	OEM address 5	m				x	x	x		1		Real

*) Absolute end points: modal, incremental end points: non-modal, otherwise modal/non-modal depending on syntax of G function.

**)As circle center points, IPO parameters act incrementally. They can be programmed in absolute mode with AC. The address modification is ignored when the parameters have other meanings (e.g., thread pitch).

1) The vocabulary word is not valid for SINUMERIK FM-NC/810D.

2) The vocabulary word is not valid for SINUMERIK FM-NC/810D/NCU571.

12.3 List of G functions/preparatory functions

List of G functions/preparatory functions

In the list of G functions/motion commands you will find all available G codes according to the appropriate functional groups.

Legend for describing the G groups

No.: internal number for, e.g., PLC interface
X: No. for GCODE_RESET_VALUES not permitted
m: modal **or** n: non-modal
Def.: Siemens AG (SAG) default setting, M: Milling: T: Turning or other conventions
MM.: Default setting, please see machine manufacturer's instructions

Group 1: Modally valid motion commands						
Name	**No.**	**Meaning**	**X**	**m/n**	**SAG**	**MM**
G0	1.	Rapid traverse		m		
G1	2.	Linear interpolation (linear interpolation)		m	Def.	
G2	3.	Circular interpolation clockwise		m		
G3	4.	Circular interpolation counter-clockwise		m		
CIP	5.	Circle through points: Circular interpolation through intermediate point		m		
ASPLINE	6.	Akima spline		m		
BSPLINE	7.	B-spline		m		
CSPLINE	8.	Cubic spline		m		
POLY	9.	Polynomial: Polynomial interpolation		m		
G33	10.	Thread cutting with constant lead		m		
G331	11.	Tapping		m		
G332	12.	Retraction (tapping)		m		
OEMIPO1 ##	13.	Reserved		m		
OEMIPO2 ##	14.	Reserved		m		
CT	15.	Circle with tangential transition		m		
G34	16.	Increase in thread pitch (progressive change)		m		
G35	17.	Decrease in thread pitch (degressive change)		m		
INVCW	18.	Involute interpolation in CW direction		m		
INVCCW	19.	Involute interpolation in CCW direction		m		

If no function from the group is programmed with modal G functions, the default setting (which can be changed in the machine data) applies: $MC_GCODE_RESET_VALUES

## The vocabulary word is **not** valid for SINUMERIK 810D/NCU571.

Group 2: Non-modally valid motions, dwell time						
Name	No.	Meaning	X	m/n	SAG	MM
G4	1.	Dwell time preset	X	n		
G63	2.	Tapping without synchronization	X	n		
G74	3.	Reference point approach with synchronization	X	n		
G75	4.	Fixed point approach	X	n		
REPOSL	5.	Repositioning linear: Linear repositioning	X	n		
REPOSQ	6.	Repositioning quadrant: Repositioning in a quadrant	X	n		
REPOSH	7.	Repositioning semicircle: Repositioning in semicircle	X	n		
REPOSA	8.	Repositioning linear all axis: Linear repositioning with all axes	X	n		
REPOSQA	9.	Repositioning quadrant all axes: Linear repositioning with all axes, geometry axes in quadrant	X	n		
REPOSHA	10.	Repositioning semicircle all axes: Repositioning with all axes; geometry axes in semicircle	X	n		
G147	11.	Soft approach with straight line	X	n		
G247	12.	Soft approach with quadrant	X	n		
G347	13.	Soft approach with semicircle	X	n		
G148	14.	Soft retraction with straight line	X	n		
G248	15.	Soft retraction with quadrant	X	n		
G348	16.	Soft retraction with semicircle	X	n		
G05	17.	Oblique plunge-cut grinding	X	n		
G07	18.	Compensatory motion during oblique plunge-cut grinding	X	n		

Group 3: Programmable frame, working area limitation and pole programming						
Name	No.	Meaning	X	m/n	SAG	MM
TRANS	1.	TRANSLATION: translation, programmable	X	n		
ROT	2.	ROTATION: rotation, programmable	X	n		
SCALE	3.	SCALE: scaling, programming	X	n		
MIRROR	4.	MIRROR: Mirroring, programmable	X	n		
ATRANS	5.	Additive TRANSLATION: additive translation, programming	X	n		
AROT	6.	Additive ROTATION: rotation, programmable	X	n		
ASCALE	7.	Additive SCALE: scaling, programming	X	n		
AMIRROR	8.	Additive MIRROR: Mirroring, programmable	X	n		
	9.	Unassigned				
G25	10.	Minimum working area limitation/spindle speed limitation	X	n		
G26	11.	Maximum working area limitation/spindle speed limitation	X	n		
G110	12.	Pole programming relative to the last programmed setpoint position	X	n		
G111	13.	Polar programming relative to origin of current workpiece coordinate system	X	n		
G112	14.	Pole programming relative to the last valid pole	X	n		
G58	15.	Programmable offset, absolute axial substitution	X	n		
G59	16.	Programmable offset, additive axial substitution	X	n		

ROTS	17.	Rotation with solid angles	X	n		
AROTS	18.	Additive rotation with solid angles	X	n		

Group 4: FIFO						
Name	No.	Meaning	X	m/n	SAG	MM
STARTFIFO	1.	Start FIFO Execute and simultaneously fill preprocessing memory		m	Def.	
STOPFIFO	2.	STOP FIFO, Stop machining; fill preprocessing memory until STARTFIFO is detected, FIFO full or end of program		m		
FIFOCTRL	3.	FIFO CTRL, Preprocessing memory control		m		

Group 6: Plane selection						
Name	No.	Meaning	X	m/n	SAG	MM
G17	1.	Plane selection 1st - 2nd geometry axis		m	Def.	
G18	2.	Plane selection 3rd - 1st geometry axis		m		
G19	3.	Plane selection 2nd - 3rd geometry axis		m		

Group 7: Tool radius compensation						
Name	No.	Meaning	X	m/n	SAG	MM
G40	1.	No tool radius compensation		m	Def.	
G41	2.	Tool radius compensation left of contour	X	m		
G42	3.	Tool radius compensation right of contour	X	m		

Group 8: Settable zero offset						
Name	No.	Meaning	X	m/n	SAG	MM
G500	1.	Deactivate all settable G54-G57 frames if G500 does not contain a value		m	Def.	
G54	2.	Settable zero offset		m		
G55	3.	Settable zero offset		m		
G56	4.	Settable zero offset		m		
G57	5.	Settable zero offset		m		
G505	6.	Settable zero offset		m		
G5xx	n+1	nth settable zero offset		m		
G599	100.	Settable zero offset		m		

The G functions of this group activate a settable user frame $P_UIFR[ ].
G54 corresponds to frame $P_UIFR[1], G505 corresponds to frame $P_UIFR[5].
The number of settable user frames and, therefore, the number of G functions in this group can be configured in the machine data $MC_MM_NUM_USER_FRAMES.

Group 9: Frame suppression						
Name	No.	Meaning	X	m/n	SAG	MM
G53	1.	Suppression of current frames: Programmable frame including system frame for TOROT and TOFRAME and active settable frame G54 ... G599.	X	n		
SUPA	2.	Suppression as for G153 and including system frames for actual-value setting, scratching, zero offset external, PAROT including handwheel offsets (DRF), [zero offset external], overlaid motion	X	n		
G153	3.	Suppression as for G53 and including all channel-specific and/or NCU-global basic frame	X	n		

Group 10: Exact stop - continuous-path mode						
Name	No.	Meaning	X	m/n	SAG	MM
G60	1.	Velocity reduction, exact positioning		m	Def.	
G64	2.	Continuous-path mode		m		
G641	3.	Continuous-path mode (G64) with programmable approximate distance		m		
G642	4.	Corner rounding with axial precision		m		
G643	5.	Block-internal axial corner rounding		m		
G644	6.	Corner rounding with specified axis dynamics		m		

Group 11: Exact stop, non-modal						
Name	No.	Meaning	X	m/n	SAG	MM
G9	1.	Velocity reduction, exact positioning	X	n		

Group 12: Block change criteria at exact stop (G60/G09)						
Name	No.	Meaning	X	m/n	SAG	MM
G601	1.	Block change at exact stop fine		m	Def.	
G602	2.	Block change at exact stop coarse		m		
G603	3.	Block change at IPO - end of block		m		

Group 13: Workpiece measuring inch/metric						
Name	No.	Meaning	X	m/n	SAG	MM
G70	1.	Input system inches (lengths)		m		
G71	2.	Input system metric (lengths)		m	Def.	
G700	3.	Input system in inches; inch/min (lengths + velocity + system variable)		m		
G710	4.	Input system, metric; mm; mm/min (lengths + velocity + system variable)		m		

Group 14: Workpiece measuring absolute/incremental						
Name	No.	Meaning	X	m/n	SAG	MM
G90	1.	Absolute dimensions		m	Def.	
G91	2.	Incremental dimension input		m		

Group 15: Feed type						
Name	No.	Meaning	X	m/n	SAG	MM
G93	1.	Inverse-time feedrate 1/rpm		m		
G94	2.	Linear feed mm/min, inch/min		m	Def.	
G95	3.	Rotational feed in mm/rev, inch/rev		m		
G96	4.	Constant cutting speed (type of feed as for G95) ON		m		
G97	5.	Constant cutting speed (type of feed as for G95) OFF		m		
G931	6.	Feedrate specified by travel time, deactivate constant path velocity		m		
G961	7.	Constant cutting speed (type of feed as for G94) ON		m		
G971	8.	Constant cutting speed (type of feed as for G94) OFF		m		
G942	9.	Freeze linear feedrate and constant cutting rate or spindle speed		m		
G952	10.	Freeze revolutional feedrate and const. cutting rate or spindle speed		m		
G962	11.	Linear or revolutional feedrate and constant cutting rate		m		
G972	12.	Freeze linear or revolutional feedrate and constant spindle speed		m		

Group 16: Feedrate override on inside and outside curvature						
Name	No.	Meaning	X	m/n	SAG	MM
CFC	1.	Constant feed at contour		m	Def.	
CFTCP	2.	Constant feed in tool center point (center-point path)		m		
CFIN	3.	Constant feed at internal radius, acceleration at external radius		m		

Group 17: Approach and retraction response, tool offset						
Name	No.	Meaning	X	m/n	SAG	MM
NORM	1.	Normal position at start and end points		m	Def.	
KONT	2.	Travel around contour at start and end points		m		
KONTT	3.	Insert polynominal with constant tangent (approach/retract)		m		
KONTC	4.	Insert polynominal with constant curvature (approach/retract)		m		

Group 18: Corner behavior, tool offset						
Name	No.	Meaning	X	m/n	SAG	MM
G450	1.	Transition circle (tool travels round workpiece corners on a circular path)		m	Def.	
G451	2.	Intersection of equidistant paths (tool backs off from the workpiece corner)		m		

Group 19: Curve transition at beginning of spline						
Name	No.	Meaning	X	m/n	SAG	MM
BNAT	1.	Begin natural: natural transition to first spline block		m	Def.	
BTAN	2.	Begin tangential: tangential transition to first spline block		m		
BAUTO	3.	Begin not a node: (no node) Start is determined by the position of the 1st point		m		

Group 20: Curve transition at end of spline						
Name	No.	Meaning	X	m/n	SAG	MM
ENAT	1.	End natural: natural transition to next traversing block		m	Def.	
ETAN	2.	End tangential: tangential transition to next traversing block at spline begin		m		
EAUTO	3.	Begin not a node: (no node) End is determined by the position of the last point		m		

Group 21: Acceleration profile						
Name	No.	Meaning	X	m/n	SAG	MM
BRISK	1.	Fast non-smoothed path acceleration		m	Def.	
SOFT	2.	Soft smoothed path acceleration		m		
DRIVE	3.	Velocity-dependent path acceleration		m		

Group 22: Tool offset types						
Name	No.	Meaning	X	m/n	SAG	MM
CUT2D	1.	Cutter compensation type 2-dimensional 2 1/2D tool offset determined by G17 – G19		m	Def.	
CUT2DF	2.	Cutter compensation type 2-dimensional frame – relative: 2 1/2D tool offset determined by frame The tool offset is effective in relation to the current frame (inclined plane)		m		
CUT3DC #	3.	Cutter compensation type 3-dimensional circumference: 3D tool compensation circumference milling		m		
CUT3DF #	4.	Cutter compensation type 3-dimensional face: 3D tool offset with inconstant tool orientation		m		
CUT3DFS #	5.	Cutter compensation type 3-dimensional face: 3D tool offset face milling with constant tool orientation independent of active frame		m		
CUT3DFF #	6.	Cutter compensation type 3-dimensional face frame: 3D tool offset face milling with constant tool orientation dependent on active frame		m		
CUT3DCC #	7.	Cutter compensation type 3-dimensional circumference: 3D tool offset circumferential milling with limitation surfaces		m		
CUT3DCCD #	8.	Cutter compensation type 3-dimensional circumference: 3D tool offset circumferential milling with limitation surfaces with differential tool		m		

# The vocabulary word is **not** valid for SINUMERIK 810D/NCU571.

Group 23: Collision monitoring at inside contours						
Name	No.	Meaning	X	m/n	SAG	MM
CDOF	1.	Collision detection OFF: Collision monitoring OFF		m	Def.	
CDON	2.	Collision detection ON: Collision monitoring ON		m		
CDOF2	3.	Collision detection OFF: Collision monitoring OFF (currently only for CUT3DC)		m		

Group 24: Feedforward control						
Name	No.	Meaning	X	m/n	SAG	MM
FFWOF	1.	Feed forward OFF: Feedforward control OFF		m	Def.	
FFWON	2.	Feed forward OFF: Feed forward control ON		m		

Group 25: Tool orientation reference						
Name	No.	Meaning	X	m/n	SAG	MM
ORIWKS #	1.	Tool orientation in workpiece coordinate system: Tool orientation in workpiece coordinate system (WCS)		m	Def.	
ORIMKS #	2.	Tool orientation in machine coordinate system: Tool orientation in machine coordinate system (MCS)		m		

# The vocabulary word is **not** valid for SINUMERIK 810D/NCU571.

Group 26: Repositioning point for REPOS						
Name	No.	Meaning	X	m/n	SAG	MM
RMB	1.	REPOS mode beginning of block: Reapproach to start of block position		m		
RMI	2.	REPOS – Mode interrupt: Reapproach to interruption point		m	Def.	
RME	3.	REPOS mode end of block: Repositioning to end-of-block position		m		
RMN	4.	REPOS mode end of nearest orbital block: Reapproach to nearest path point		m		

Group 27: Tool offset for change in orientation at outside corners						
Name	No.	Meaning	X	m/n	SAG	MM
ORIC #	1.	Orientation change continuously: Orientation changes at outside corners are superimposed on the circle block to be inserted		m	Def.	
ORID #	2.	Orientation change discontinuously: Orientation changes are performed before the circle block		m		

# The vocabulary word is **not** valid for SINUMERIK 810D/NCU571.

Group 28: Working area limitation ON/OFF						
Name	No.	Meaning	X	m/n	SAG	MM
WALIMON	1.	Working area limitation ON: Working area limitation ON		m	Def.	
WALIMOF	2.	Working area limitation OFF: Working area limitation OFF		m		

Group 29: Radius - diameter						
Name	No.	Meaning	X	m/n	SAG	MM
DIAMOF	1.	Diametral programming OFF: Diameter programming OFF; radius programming for G90/G91		m	Def.	
DIAMON	2.	Diametral programming ON: Diameter programming ON for G90/G91		m		
DIAM90	3.	Diametral programming G90: Diameter programming for G90; radius programming for G91		m		
DIAMCYCOF	4.	Diametral programming OFF: Radius programming for G90/G91: ON. The G-code of this group that was last active remains active for display		m		

Group 30: Compressor ON/OFF						
Name	No.	Meaning	X	m/n	SAG	MM
COMPOF #	1.	Compressor OFF		m	Def.	
COMPON #	2.	Compressor ON		m		
COMPCURV #	3.	Compressor ON: Polynomials with constant curvature		m		
COMPCAD #	4.	Compressor ON: Optimized surface quality CAD program		m		

Group 31: OEM - G group						
Name	No.	Meaning	X	m/n	SAG	MM
G810 #	1.	OEM - G function			Def.	
G811 #	2.	OEM - G function				
G812 #	3.	OEM - G function				
G813 #	4.	OEM - G function				
G814 #	5.	OEM - G function				
G815 #	6.	OEM - G function				
G816 #	7.	OEM - G function				
G817 #	8.	OEM - G function				
G818 #	9.	OEM - G function				
G819 #	10.	OEM - G function				

Two G groups are reserved for the OEM. This enables the OEM to program functions that can be customized.
# The vocabulary word is **not** valid for SINUMERIK 810D/NCU571.

Group 32: OEM - G group						
Name	No.	Meaning	X	m/n	SAG	MM
G820 #	1.	OEM - G function			Def.	
G821 #	2.	OEM - G function				
G822 #	3.	OEM - G function				
G823 #	4.	OEM - G function				
G824 #	5.	OEM - G function				
G825 #	6.	OEM - G function				
G826 #	7.	OEM - G function				
G827 #	8.	OEM - G function				
G828 #	9.	OEM - G function				
G829 #	10.	OEM - G function				

Two G groups are reserved for the OEM. This enables the OEM to program functions that can be customized.

Group 33: Settable fine tool offset						
Name	No.	Meaning	X	m/n	SAG	MM
FTOCOF #	1.	Fine tool offset compensation OFF: Online fine tool offset OFF		m	Def.	
FTOCON #	2.	Fine tool offset compensation ON: Online fine tool offset ON	X	m		

Group 34: Tool orientation smoothing						
Name	No.	Meaning	X	m/n	SAG	MM
OSOF #	1.	Tool orientation smoothing OFF		m	Def.	
OSC #	2.	Continuous tool orientation smoothing		m		
OSS #	3.	Tool orientation smoothing at end of block		m		
OSSE #	4.	Tool orientation smoothing at start and end of block		m		

Group 35: Punching and nibbling						
Name	No.	Meaning	X	m/n	SAG	MM
SPOF#	1.	Stroke/punch OFF: Stroke OFF, nibbling, punching OFF		m	Def.	
SON #	2.	Stroke ON: Nibbling ON		m		
PON #	3.	Punch ON: Punching ON		m		
SONS #	4.	Stroke ON slow: Nibbling ON in IPO cycle	X	m		
PONS #	5.	Punch ON slow: Punching ON in IPO cycle	X	m		

Group 36: Punching with delay						
Name	No.	Meaning	X	m/n	SAG	MM
PDELAYON #	1.	Punch with delay ON: Punching with delay ON		m	Def.	
PDELAYOF #	2.	Punch with delay OFF: Punching with delay OFF		m		

# The vocabulary word is **not** valid for SINUMERIK 810D/NCU571.

Group 37: Feed profile						
Name	No.	Meaning	X	m/n	SAG	MM
FNORM #	1.	Feed normal: Feed normal (as per DIN 66025)		m	Def.	
FLIN #	2.	Feed linear: Feed linear variable		m		
FCUB #	3.	Feed cubic: Feedrate variable according to cubic spline		m		

Group 38: Assignment of high-speed inputs/outputs for punching/nibbling						
Name	No.	Meaning	X	m/n	SAG	MM
SPIF1 #	1.	Stroke/punch interface 1: fast NCK inputs/outputs for punching/nibbling byte 1		m	Def.	
SPIF2 #	2.	Stroke/punch interface 2: fast NCK inputs/outputs for punching/nibbling byte 2		m		

Group 39: Programmable contour accuracy						
Name	No.	Meaning	X	m/n	SAG	MM
CPRECOF	1.	Contour precision OFF: Programmable contour precision OFF		m	Def.	
CPRECON	2.	Contour precision ON: Programmable contour precision ON		m		

#The vocabulary word is **not** valid for SINUMERIK NCU571.

Group 40: Tool radius compensation constant						
Name	No.	Meaning	X	m/n	SAG	MM
CUTCONOF	1.	Constant radius compensation OFF		m	Def.	
CUTCONON	2.	Constant radius compensation ON		m		

Group 41: Interrupt thread cutting						
Name	No.	Meaning	X	m/n	SAG	MM
LFOF	1.	Interrupt thread cutting OFF		m	Def.	
LFON	2.	Interrupt thread cutting ON		m		

Group 42: Toolholder						
Name	No.	Meaning	X	m/n	SAG	MM
TCOABS	1.	Tool Carrier Orientation Absolute: Toolholder orientation absolute		m	Def.	
TCOFR	2.	Toolholder orientation frame alignment of tool on Z axis		m		
TCOFRZ	3.	Orientable toolholder frame-related (tool on Z axis)		m		
TCOFRY	4.	Orientable toolholder frame-related (tool on Y axis)		m		
TCOFRX	5.	Orientable toolholder frame-related (tool on X axis)		m		

Group 43: SAR approach direction						
Name	No.	Meaning	X	m/n	SAG	MM
G140	1.	SAR approach direction defined by G41/G42		m	Def.	
G141	2.	SAR approach direction to left of contour		m		
G142	3.	SAR approach direction to right of contour		m		
G143	4.	SAR approach direction tangent-dependent		m		

Group 44: SAR path segmentation						
Name	No.	Meaning	X	m/n	SAG	MM
G340	1.	Spatial approach block (depth and in plane (helix))		m	Def.	
G341	2.	Initial infeed on perpendicular axis (z), then approach in plane		m		

Group 45: Path reference for FGROUP axes:						
Name	No.	Meaning	X	m/n	SAG	MM
SPATH	1.	Path reference for FGROUP axes is arc length		m	Def.	
UPATH	2.	Path reference for FGROUP axes is curve parameter		m		

Group 46: Plane definition for rapid lift:						
Name	No.	Meaning	X	m/n	SAG	MM
LFTXT	1.	Tangential tool direction on retraction		m	Def.	
LFWP	2.	Non-tangential tool direction on retraction		m		
LFPOS	3.	Axial retraction to a position		m		

Group 47: Mode switchover for external NC code						
Name	No.	Meaning	X	m/n	SAG	MM
G290	1.	Switchover to SINUMERIK mode (activate SINUMERIK language mode)		m	Def.	
G291	2.	Switchover to ISO 2/3 mode (activate ISO language mode)		m		

Group 48: Approach and retraction response, TRC						
Name	No.	Meaning	X	m/n	SAG	MM
G460	1.	Collision monitoring for approach and retraction block ON		m	Def.	
G461	2.	If no intersection in TRC block, extend border block with arc		m		
G462	3.	If no intersection in TRC block, extend border block with straight line		m		

Group 49: Point to point motion						
Name	No.	Meaning	X	m/n	SAG	MM
CP	1.	continuous path; path motion		m	Def.	
PTP	2.	point to point; point to point motion (synchronized axis motion)		m		
PTPG0	3.	point to point; point to point motion only with G0, otherwise path motion CP		m		

Group 50: Orientation programming						
Name	No.	Meaning	X	m/n	SAG	MM
ORIEULER	1.	Orientation angle via Euler angle		m	Def.	
ORIRPY	2.	Orientation angle via RPY angle (rotation sequence XYZ)		m		
ORIVIRT1	3.	Orientation angle via virtual orientation axes (definition 1)		m		
ORIVIRT2	4.	Orientation angle via virtual orientation axes (definition 2)		m		
ORIAXPOS	5.	Orientation angle via virtual orientation axes with rotary axis positions		m		
ORIRPY2	6.	Orientation angle via RPY angle (rotation sequence ZYX)		m		

Group 51: Orientation interpolation						
Name	No.	Meaning	X	m/n	SAG	MM
ORIVECT	1.	Large-radius circular interpolation (identical to ORIPLANE)		m	Def.	
ORIAXES	2.	Linear interpolation of machine axes or orientation axes		m		
ORIPATH	3.	Tool orientation trajectory referred to path		m		
ORIPLANE	4.	Interpolation in plane (identical to ORIVECT)		m		
ORICONCW	5.	Interpolation on a peripheral surface of the cone in clockwise direction		m		
ORICONCCW	6.	Interpolation on a conical peripheral surface in CCW direction		m		
ORICONIO	7.	Interpolation on a conical peripheral surface with intermediate orientation setting		m		
ORICONTO	8.	Interpolation on a peripheral surface of the cone with tangential transition		m		
ORICURVE	9.	Interpolation with additional space curve for orientation		m		
ORIPATHS	10.	Tool orientation in relation to path, blips in the orientation characteristic are smoothed		m		

Group 52: Workpiece-oriented WCS						
Name	No.	Meaning	X	m/n	SAG	MM
PAROTOF	1.	Deactivate workpiece-related frame rotation		m	Def.	
PAROT	2.	Align workpiece coordinate system (WCS) on workpiece		m		

Group 53: Frame rotations in tool direction						
Name	No.	Meaning	X	m/n	SAG	MM
TOROTOF	1.	Frame rotation in tool direction OFF		m	Def.	
TOROT	2.	Frame rotation ON Z axis parallel to tool orientation		m		
TOROTZ	3.	Frame rotation ON Z axis parallel to tool orientation		m		
TOROTY	4.	Frame rotation ON Y axis parallel to tool orientation		m		
TOROTX	5.	Frame rotation ON X axis parallel to tool orientation		m		
TOFRAME	6.	Frame rotation in tool direction Z axis parallel to tool orientation		m		
TOFRAMEZ	7.	Frame rotation in tool direction Z axis parallel to tool orientation		m		
TOFRAMEY	8.	Frame rotation in tool direction Y axis parallel to tool orientation		m		
TOFRAMEX	9.	Frame rotation in tool direction X axis parallel to tool orientation		m		

Group 54: Rotation of the rotational vector						
Name	No.	Meaning	X	m/n	SAG	MM
ORIROTA	1.	Orientation Rotation Absolute Rotation **absolute**		m	Def.	
ORIROTR	2.	Orientation Rotation Relative **relative** rotational vector		m		
ORIROTT	3.	Orientation Rotation Tangential **tangential** rotational vector in relation to change in orientation		m		
ORIROTC	4.	Orientation Rotation Tangential **tangential** rotational vector in relation to path tangent		m		

Group 55: Rapid traverse with/without linear interpolation						
Name	No.	Meaning	X	m/n	SAG	MM
RTLION	1.	Rapid traverse (G0) with linear interpolation ON: G0 with linear interpolation		m	Def.	
RTLIOF	2.	Rapid traverse (G0) with linear interpolation OFF: G0 without linear interpolation (single-axis interpolation)		m		

Group 56: Inclusion of tool wear						
Name	No.	Meaning	X	m/n	SAG	MM
TOWSTD	1.	Tool wear default initial setting value for offsets in tool length		m	Def.	
TOWMCS	2.	Tool WearCoard MCS: Wear values in machine coordinate system (MCS)		m		
TOWWCS	3.	Tool WearCoard WCS: Wear values in workpiece coordinate system (WCS)		m		
TOWBCS	4.	Tool WearCoard BCS: Wear values in basic coordinate system (MCS)		m		
TOWTCS	5.	Tool WearCoard TCS: Wear values in the tool coordinate system (tool carrier ref. point T at the tool holder)		m		
TOWKCS	6.	Wear values in the coordinate system of the tool head for kinetic transformation (differs from MCS by tool rotation)		m		

Group 57: Automatic corner override						
Name	No.	Meaning	X	m/n	SAG	MM
FENDNORM	1.	Corner deceleration deactivated		m	Def.	
G62	2.	Corner deceleration at inside corners when tool radius offset is active		m		
G621	3.	Corner deceleration at all corners		m		

Group 58: Reserved for retracting from software end position						
Name	No.	Meaning	X	m/n	SAG	MM
RELIEVEON	1.	Retracting from software limit switch ON		m		
RELIEVEOF	2.	Retracting from software limit switch OFF		m	Def.	

Group 59: Technology G groups						
Name	No.	Meaning	X	m/n	SAG	MM
DYNNORM	1.	Standard dynamic, as previously		m	Def.	
DYNPOS	2.	Positioning mode, tapping		m		
DYNROUGH	3.	Roughing		m		
DYNSEMIFIN	4.	Finishing		m		
DYNFINISH	5.	Smooth-finishing		m		

12.4 List of predefined subprograms

12.4.1 Predefined subprogram calls

List of predefined subprograms

The list of predefined subprograms contains all available subprograms grouped according to function.

Some control functions are activated with subprogram call syntax.

1. Coordinate system					
Vocabulary word/ function identifier	**1st parameter**	**2nd parameter**	**3rd-15th parameter**	**4th-16th parameter**	**Explanation**
PRESETON	AXIS*: Axis identifier Machine axis	REAL: Preset offset G700/G7100 context	3.-15. Parameter as 1 ...	4.-16. parameter as 2 ...	Sets the actual value for programmed axes. One axis identifier is programmed at a time, with its respective value in the next parameter. Preset offsets can be programmed for up to 8 axes with PRESETON.
DRFOF					Deletes the DRF offset for all axes assigned to the channel.

*) As a general rule, geometry or special axis identifiers can also be used instead of the machine axis identifier, as long as the reference is unambiguous.

Predefined subprogram calls

2. Axis groupings			
	1st-8th parameter	**Explanation**	
FGROUP	Channel axis identifiers	Variable F value reference: defines the axes to which the path feed refers. Maximum axis number: 8 The default setting for the F value reference is activated with FGROUP ( ) without parameters.	
	1st-8th parameter	**2nd-9th parameter**	**Explanation**
SPLINEPATH	INT: Spline group (must be 1)	AXIS: Geometry or special axis identifier	Definition of the spline group Maximum number of axes: 8
BRISKA	AXIS		Switch on brisk axis acceleration for the programmed axes
SOFTA	AXIS		Switch on jerk limited axis acceleration for programmed axes

DRIVEA ###	AXIS		Switch on knee-shaped acceleration characteristic for programmed axes
JERKA	AXIS		The acceleration behavior set in machine data $MA_AX_JERK_ENABLE is active for the programmed axes.

# The vocabulary word is **not** valid for SINUMERIK 810D/NCU571.
#The vocabulary word is **not** valid for SINUMERIK 810D.
### The vocabulary word is only valid for SINUMERIK FM-NC.

3. Coupled motion							
Vocabulary word/ subprogram identifier	**1st parameter**	**2nd param.**	**3rd param.**	**4th param.**	**5th param.**	**6th param.**	**Explanation**
TANG	AXIS: Axis name Following axis	AXIS: Leading axis 1	AXIS: Leading axis 2	REAL: Coupling factor	CHAR: Option: "B": follow-up in basic coordinate system "W": follow-up in work-piece coord. system	CHAR Optimiz-ation: "S" default"P" " autom. with rounding travel, angle tolerance	Preparatory instruction for the definition of a tangential follow-up: The tangent for the follow-up is determined by the two master axes specified. The coupling factor specifies the relationship between a change in the angle of tangent and the following axis. It is usually 1. Optimization: See PGA
TANGON	AXIS: Axis name Following axis	REAL: Offset Angle	REAL: Rounding travel	REAL: Angle tolerance			Tangential follow-up mode ON par. 3, 4 with TANG Par. 6 = "P"
TANGOF	AXIS: Axis name Following axis						Tangential follow-up mode OFF
TLIFT	AXIS: Following axis	REAL: Lift-off path	REAL: Factor				Tangential lift: tangential follow-up mode, stop at contour end rotary axis lift-off possible
TRAILON	AXIS: Following axis	AXIS: Master axis	REAL: Coupling factor				Trailing ON: Asynchronous coupled motion ON
TRAILOF	AXIS: Following axis	AXIS: Master axis					Trailing OFF: Asynchronous coupled motion OFF

6. Revolutional feedrate			
Vocabulary word/ function identifier	1st parameter	2nd parameter	Explanation
FPRAON	AXIS: Axis, for which revolutional feedrate is deactivated	AXIS: Axis/spindle from which the revolutional feedrate is derived. If no axis has been programmed, the revolutional feedrate is derived from the master spindle.	Feedrate per revolution axial ON: Axial revolutional feedrate ON
FPRAOF	AXIS: Axis for which revolutional feedrate is deactivated		Feedrate per revolution axial OFF: Axial revolutional feedrate OFF The revolutional feedrate can be deactivated for several axes at once. You can program as many axes as are permitted in a block.
FPR	AXIS: Axis/spindle from which the revolutional feedrate is derived. If no axis has been programmed, the revolutional feedrate is derived from the master spindle.		Feedrate per revolution: selection of a rotary axis or spindle from which the revolutional feedrate of the path is derived if G95 is programmed. If no axis/spindle has been programmed, the revolutional feedrate is derived from the master spindle. The setting made via FPR is modal.

It is also possible to program a spindle instead of an axis: FPR(S1) or FPR(SPI(1))

7. Transformations			
Vocabulary word/ function identifier	1st parameter	2nd parameter	Explanation
TRACYL	REAL: Working diameter	INT: Number of transformation	Cylinder: Peripheral surface transformation Several transformations can be set per channel. The transformation number specifies which transformation is to be activated. If the second parameter is omitted, the transformation group defined in the MD is activated.
TRANSMIT	INT: Number of transformation		Transmit: Polar transformation Several transformations can be set per channel. The transformation number specifies which transformation is to be activated. If the parameter is omitted, the transformation group defined in the MD is activated.
TRAANG	REAL: Angle	INT: Number of transformation	Inclined axis transformation Several transformations can be set per channel. The transformation number specifies which transformation is to be activated. If the 2nd parameter is omitted, the transformation group defined in the MD is activated. If the angle is not programmed: TRAANG (,2) or TRAANG, and the last angle is modal.
TRAORI	INT: Number of transformation		Transformation oriented: 4, 5-axis transformation Several transformations can be set per channel. The transformation number specifies which transformation is to be activated.
TRACON	INT: Number of transformation	REAL: Further parameters, MD-dependent	Transformation concentrated: Cascaded transformation; the meaning of the parameters depends on the type of cascading.
TRAFOOF			Deactivate transformation

For each transformation type, there is one command for one transformation per channel. If there are several transformations of the same transformation type per channel, the transformation can be selected with the corresponding command and parameters. It is possible to deselect the transformation by a transformation change or an explicit deselection.

8. Spindles			
Vocabulary word/ subprogram identifier	1st parameter	2nd parameter and others	Explanation
SPCON	INT: Spindle number	INT: Spindle number	Spindle position control ON: Switch to position-controlled spindle operation
SPCOF	INT: Spindle number	INT: Spindle number	Spindle position control OFF: Switch to speed-controlled spindle operation
SETMS	INT: Spindle number		Set master spindle: declare the spindle master spindle for the current channel. SETMS( ) without parameters activates the default setting in the machine data.

9. Grinding		
Vocabulary word/ subprogram identifier	1st parameter	Explanation
GWPSON	INT: Spindle number	Grinding wheel peripheral speed ON: Constant grinding wheel peripheral speed ON If the spindle number is not programmed, then grinding wheel peripheral speed is selected for the spindle of the active tool.
GWPSOF	INT: Spindle number	Grinding wheel peripheral speed OFF. Constant grinding wheel peripheral speed OFF. If the spindle number is not programmed, grinding wheel peripheral speed is deselected for the spindle of the active tool.
TMON	INT: Spindle number	Tool monitoring ON: If no T number is programmed, monitoring is activated for the active tool.
TMOF	INT: T number	Tool monitoring OFF: If no T number is programmed, monitoring is deactivated for the active tool.

10. Stock removal					
Vocabulary word/ subprogram identifier	1st parameter	2nd parameter	3rd parameter	4th parameter	Explanation
CONTPRON	REAL [, 11]: Contour table	CHAR: Stock removal method "L": Longitudinal turning: External machining "P": Face turning: External machining "N": Face turning: Internal machining: "G": Longitudinal turning: Internal machining	INT: Number of relief cuts	INT: Status of calculation: 0: unchanged 1: Calculation forwards and backwards	Contour preparation on: Activate reference point editing The contour programs and NC blocks which are called in the following steps are divided into individual movements and stored in the contour table. The number of relief cuts is returned.
CONTDCON	REAL [, 6]: Contour table	INT: 0: in programmed direction			Contour decoding The blocks for a contour are stored in a named table with one table line per block and coded to save memory.
EXECUTE	INT: Error status				EXECUTE: Activate program execution. This switches back to normal program execution from reference point editing mode or after setting up a protection zone.

11. Execute table		
Vocabulary word/ subprogram identifier	1st parameter	Explanation
EXECTAB	REAL [11]: Element from motion table	Execute table: Execute an element from a motion table.

12. Protection zones						
Vocabulary word/ function identifier	1st parameter	2nd parameter	3rd parameter	4th parameter	5th parameter	Explanation
CPROTDEF	INT: Number of protection zone	BOOL: TRUE: tool-oriented protection zone	INT: 0: 4. U.G. Parameters not evaluated 1: 4. Parameter evaluated 2: 5. Parameter evaluated 3: 4. U. 5. Parameter evaluated	REAL: Limit in plus direction	REAL: Limit in minus direction	Channel-specific protection area definition: Definition of a channel-specific protection zone
NPROTDEF	INT: Number of protection zone	BOOL: TRUE: tool-oriented protection zone	INT: 0: 4. U. 5. parameters not evaluated 1: 4. Parameter evaluated 2: 5. Parameter evaluated 3: 4. U. 5. parameter evaluated	REAL: Limit in plus direction	REAL: Limit in minus direction	NCK-specific protection area definition: Definition of a machine-specific protection zone
CPROT	INT: Number of protection zone	INT: Option 0: Protection zone OFF 1: Preactivate protection zone 2: Protection zone ON 3: Preactivate protection zone with conditional stop	REAL: Offset of protection zone in 1st geometry axis	REAL: Offset of protection zone in 2nd geometry axis	REAL: Offset of protection zone in 3rd geometry axis	Channel-specific protection zone ON/OFF

NPROT	INT: Number of protection zone	INT: Option	REAL: Offset of protection zone in 1st geometry axis	REAL: Offset of protection zone in 2nd geometry axis	REAL: Offset of protection zone in 3rd geometry axis	Machine-specific protection zone ON/OFF
EXECUTE	VAR INT: Error status	EXECUTE: Activate program execution. This switches back to normal program execution from reference point editing mode or after setting up a protection zone.				

13. Preprocessing/single block		
STOPRE		Stop processing: Stop preprocessing until all prepared blocks are executed in main run.

14. Interrupts		
Vocabulary word/ function identifier	1st parameter	Explanation
ENABLE #	INT: Number of interrupt input	Activate interrupt: Activates the interrupt routine assigned to the hardware input with the specified number. An interrupt is enabled after the SETINT instruction.
DISABLE #	INT: Number of interrupt input	Deactivate interrupt: Deactivates the interrupt routine assigned to the hardware input with the specified number. Fast retraction is not executed. The assignment between the hardware input and the interrupt routine made with SETINT remains valid and can be reactivated with ENABLE.
CLRINT #	INT: Number of interrupt input	Select interrupt: Cancel the assignment of interrupt routines and attributes to an interrupt input. The interrupt routine is deactivated and no reaction occurs when the interrupt is generated.

#The vocabulary word is **not** valid for SINUMERIK 810D.

15. Motion synchronization		
CANCEL	INT: Number of synchronized action	Aborts the modal motion-synchronous action with the specified ID

16. Function definition					
	1st parameter	2nd parameter	3rd parameter	4th-7th parameter	Explanation
FCTDEF	INT: Function number	REAL: Lower limit value	REAL: Upper limit value	REAL: Coefficients a0 – a3	Define polynomial. This is evaluated in SYFCT or PUTFTOCF.

17. Communication			
Vocabulary word/subprogram identifier	1st parameter	2nd parameter	Explanation
MMC #	STRING: Command	CHAR: Acknowledgment mode** "N": Without acknowledgment "S": Synchronous acknowledgment "A": Asynchronous acknowledgment	MMC command: Command ON MMC command interpreter for the configuration of windows via NC program see /AM/IM1 Start-Up Functions for the MMC

#)The vocabulary word is **not** valid for SINUMERIK 810D.
****)Acknowledgement mode:**
Commands are acknowledged at the request of the executing component (channel, NC …).
Without acknowledgement: Program execution is continued when the command has been transmitted. The sender is not informed if the command cannot be executed successfully.

18. Program coordination							
Vocabulary word/subprogram identifier	1st parameter	2nd parameter	3rd parameter	4th parameter	5th parameter	6th-8th parameter	Explanation
INIT #	INT: Channel number	STRING: path	CHAR: acknowledgement mode**				Selection of a module for execution in a channel.
START #	INT: Channel number	INT: Channel number					Starts selected programs simultaneously on multiple channels from running program. The command has no effect on the existing channel. 1 : 1st channel; 2 : 2. channel.
WAITE #	INT: Channel number	INT: Channel number					Wait for end of program: Wait for end of program on another channel

WAITM #	INT: Marker number 0-9	INT: Channel number	INT: Channel number	INT: Channel number			Wait: Wait for a marker to be reached in other channels. The program waits until the WAITM with the relevant marker has been reached in the other channel. The number of the own channel can also be specified.
WAITP	AXIS: Axis identifier	AXIS: Axis identifier	AXIS: Axis identifier	AXIS: Axis identifier	AXIS: Axis identifier	AXIS: Axis identifier	Wait for positioning axis: Wait for positioning axes to reach their programmed endpoint.
WAITS	INT: Spindle number	INT: Spindle number	INT: Spindle number	INT: Spindle number	INT: Spindlenumber		Wait for positioning spindle: Wait until programmed spindles previously programmed with SPOSA reach their programmed endpoint.
RET							End of subprogram with no function output to the PLC.
GET #	AXIS	AXIS	AXIS	AXIS	AXIS	AXIS	Assign machine axis
GETD#	AXIS	AXIS	AXIS	AXIS	AXIS	AXIS	Assign machine axis directly
RELEASE #	AXIS	AXIS	AXIS	AXIS	AXIS	AXIS	Release machine axis
PUTFTOC #	REAL: Offset value	INT: Parameter number	INT: Channel number	INT: Spindle number			Put fine tool correction: Fine tool compensation
PUTFTOCF #	INT: No. of function. The number used here must be specified in FCTDEF.	VAR REAL: Reference value *)	INT: Parameter number	INT: Channel number	INT: Spindlenumber		Put fine tool correction function dependent: Change online tool compensation according to a function defined with FCTDEF (max. 3rd degree polynomial).

The SPI function can also be used to program a spindle instead of an axis: GET(SPI(1))

#)The vocabulary word is not valid for SINUMERIK FM-NC/NCU571.

****) Acknowledgment mode:**

Commands are acknowledged on request from the executing component (channel, NC, etc.).

Without acknowledgement: Program execution is continued when the command has been transmitted. The executing component is not informed if the command cannot be executed successfully. Acknowledgment mode "N" or "n".
Synchronous acknowledgement: The program execution is paused until the receiving component acknowledges the command. If the acknowledgement is positive, the next command is executed.
If the **acknowledgement is negative** an error is output.
Acknowledgement mode "S", "s" or omit.

The acknowledgment behavior is defined for some commands and programmable for others. The acknowledgment behavior is always synchronous for program coordination commands. If the acknowledgment mode is omitted, synchronous acknowledgment is taken as the default.

19. Data access		
	1st parameter	Explanation
CHANDATA	INT: Channel number	Set channel number for channel data access (only permitted in initialization block); the subsequent accesses refer to the channel set with CHANDATA.

20. Messages			
	1st parameter	2nd parameter	Explanation
MSG	STRING: signal		Message modal: The message is active until the next message is queued

22. Alarms			
	1st parameter	2nd parameter	Explanation
SETAL	INT: Alarm number (cycle alarms)		Set alarm: Sets alarm

23. Compensation			
Vocabulary word/subprogram identifier	1st parameter-4th parameter		Explanation
QECLRNON	AXIS: Axis number		Quadrant error compensation learning ON Quadrant error compensation learning ON
QECLRNOF			Quadrant error compensation learning OFF: Quadrant error compensation learning OFF

24. Tool management					
	1st parameter	2nd parameter	3rd parameter		Explanation
DELT	STRING[32]: Tool designation	INT: Duplo number			Delete tool. Duplo number can be omitted.
GETSELT	VAR INT: T number (return value)	INT: Spindle number			Get selected T number. If no spindle number is specified, the command for the master spindle applies.
SETPIECE	INT: Count	INT: Spindle number			Set number of workpieces for all tools assigned to the spindle. If the spindle number is omitted, the command applies to the master spindle.
SETDNO	INT: Tool number T	INT: Tool edge no.	INT: D No.		Set D no. of tool (T) and its tool edge to new
DZERO					Set D numbers of all tools of the TO unit assigned to the channel to invalid
DELDL	INT: Tool number T	INT: D No.			Delete all additive offsets of the tool edge (or of a tool if D is not specified)
SETMTH	INT: Tool-holder no.				Set toolholder no.
POSM	INT: Location no. for positioning	INT: No. of the magazine to be moved	INT: Location number of the internal magazine	INT: Magazine number of the internal magazine	Position magazine
SETTIA	VAR INT: Status=result of operation (return value)	INT: Magazine number	INT: Wear grouping no.		Deactivate tool from wear group
SETTA	VAR INT: Status=result of operation (return value)	INT: Magazine number	INT: Wear grouping no.		Activate tool from wear group
RESETMON	VAR INT: Status=result of operation (return value)	INT: Internal T no.	INT: D no. of tool		Set actual value of tool to setpoint

#) The vocabulary word is not valid for SINUMERIK FM-NC.

25. Synchronous spindle							
	1st parameter	2nd parameter	3rd parameter	4th parameter	5th parameter	6th parameter	Explanation
COUPDEF #	AXIS: Following axis	AXIS: Leading axis	REAL: Counter transformation ratio (FA)	REAL: Denominator transformation ratio (LA)	STRING[8]: Block change behavior: "NOC": no block change control, block change is enabled immediately, "FINE": block change on "synchronism fine", "COARSE": block change on synchronism coarse and "IPOSTOP": block change in setpoint-dependent termination of overlaid movement. If the block change behavior is not specified, the set behavior is applicable and there is no change.	STRING[2]: "DV": Setpoint coupling "AV": Actual-value coupling	Couple definition: definition of synchronized spindle grouping
COUPDEL #	AXIS: Following axis	AXIS: Leading axis					Couple delete: Delete synchronous spindle group
COUPRES #	AXIS: Following axis	AXIS: Leading axis					Couple reset: Reset synchronous spindle group. The programmed values become invalid. The machine data values are valid.

For synchronous spindles, the axis parameters are programmed with SPI(1) or S1.

26. Structure instructions in the STEP editor (editor-based program support)					
	1st parameter	2nd parameter	3rd parameter		Explanation
SEFORM	STRING[128]: section name	INT: level	STRING[128]: icon		Current section name for STEP editor

#) The vocabulary word is not valid for SINUMERIK 810 D.

Vocabulary word/subprogram identifier	1st parameter	2nd parameter	3rd parameter	4th parameter	Explanation
COUPON #	AXIS: Following axis	AXIS: Master axis	REAL: Activation position of following axis		Couple on: Activate ELG group/synchronous spindle pair. If no activation positions are specified, the couple is activated as quickly as possible (ramp). If an activation position is specified for the following axis and spindle, this refers absolutely or incrementally to the master axis or spindle. Parameters 4 and 5 only have to be programmed if the 3rd parameter is specified.
COUPOF #	AXIS: Following axis	AXIS: Master axis	REAL: Deactivation position of following axis (absolute)	REAL: Deactivation position of master axis (absolute)	Couple OFF: Deactivate ELG group/synchronous spindle pair. The couple parameters are retained. If positions are specified, the couple is only canceled when all the specified positions have been overtraveled. The following spindle continues to revolve at the last speed programmed before deactivation of the couple.
WAITC #	AXIS: Axis/ spindle	STRING[8]: Block change criterion	AXIS: Axis/ spindle	STRING[8]: Block change criterion	Wait for couple condition: Wait until couple block change criterion for the axes/spindles is fulfilled. Up to 2 axes/spindles can be programmed. Block change criterion: "NOC": no block change control, block change is enabled immediately, "FINE": block change on "synchronism fine", "COARSE": block change on synchronism coarse and "IPOSTOP": Block change in setpoint-dependent termination of overlaid movement. If the block change behavior is not specified, the set behavior is applicable and there is no change.
AXCTSWE	AXIS: Axis/spindle				Advance container axis

#)The vocabulary word is **not** valid for SINUMERIK 810D.

12.4.2 Predefined subprogram calls in motion-synchronous actions

Predefined subprogram calls in motion-synchronous actions

27. Synchronous procedures				
Vocabulary word/ function identifier	1st parameter	2nd parameter	3rd parameter to 5th parameter	Explanation
STOPREOF				Stop preparation OFF: A synchronized action with a STOPREOF command causes a preprocessing stop after the next output block (= block for the main run). The preprocessing stop is canceled with the end of the output block or when the STOPREOF condition is fulfilled. All synchronized action instructions with the STOPREOF command are therefore interpreted as having been executed.
RDISABLE				Read-in disable Read-in disable
DELDTG	AXIS: Axis for axial delete distance-to-go (optional). If the axis is omitted, delete distance-to-go is triggered for the path distance			Delete distance-to-go: A synchronized action with the DELDTG command causes a preprocessing stop after the next output block (= block for the main run). The preprocessing stop is canceled with the end of the output block or when the first DELDTG condition is fulfilled. The axial distance to the destination point on an axial delete distance-to-go is stored in $AA_DELT[axis]; the distance-to-go is stored in $AC_DELT.
SYNFCT	INT: Number of polynomial function defined with FCTDEF.	VAR REAL: Reference variable*)	VAR REAL: input variable**)	If the condition in the motion synchronous action is fulfilled, the polynomial determined by the first expression is evaluated at the input variable. The upper and lower range of the value is limited and the input variable is assigned.
FTOC	INT: Number of polynomial function defined with FCTDEF	VAR REAL: input variable**)	INT: Length 1, 2, 3 INT: Channel number INT: Spindle number	Modify tool fine compensation according to a function defined with FCTDEF (polynomial no higher than 3rd degree). The number used here must be specified in FCTDEF.

*) Only special system variables are permissible as result variables. These are described in the Programming Guide Advanced in the section on "Write main run variable".

**) Only special system variables are permissible as input variables. These variables are described in the Programming Guide Advanced in the list of system variables.

12.4.3 Predefined functions

Predefined functions

Predefined functions are invoked by means of a function call. Function calls return a value. They can be included as an operand in an expression.

1. Coordinate system						
Vocabulary word/ function identifier	Result	1st parameter	2nd parameter			Explanation
CTRANS	FRAME	AXIS	REAL: Offset	3. - 15. Parameter as 1 ...	4. - 16. Parameter as 2 ...	Translation: Zero offset for multiple axes. One axis identifier is programmed at a time, with its respective value in the next parameter. CTRANS can be used to program offset for up to 8 axes.
CROT	FRAME	AXIS	REAL: Rotation	3./5. Parameter as 1 ...	4./6. Parameter as 2 ...	Rotation: Rotation of the current coordinate system. Maximum number of parameters: 6 (one axis identifier and one value per geometry axis)
CSCALE	FRAME	AXIS	REAL: Scale factor	3. - 15. Parameter as 1 ...	4. - 16. Parameter as 2 ...	Scale: Scale factor for multiple axes. Maximum number of parameters is 2* maximum number of axes (axis identifier and value per axis). One axis identifier is programmed at a time, with its respective value in the next parameter. CSCALE can be used to program scale factors for up to 8 axes.
CMIRROR	FRAME	AXIS	2. - 8. Parameter as 1 ...			Mirror: Mirror on a coordinate axis
MEAFRAME	FRAME	2-dim. REAL array	2-dim. REAL array	3. Parameter: REAL variables		Frame calculation from 3 measuring points in space

Frame functions CTRANS, CSCALE, CROT and CMIRROR are used to generate frame expressions.

2. Geometry functions					
Vocabulary word/ function identifier	Result	1st parameter	2nd parameter	3rd parameter	Explanation
CALCDAT	BOOL: Error status	VAR REAL [,2]: Table with input points (abscissa and ordinate for points 1, 2, 3, etc.)	INT: Number of input points for calculation (3 or 4)	VAR REAL [3]: Result: Abscissa, ordinate and radius of calculated circle center point	CALCDAT: Calculate circle data Calculates radius and center point of a circle from 3 or 4 points (according to parameter 1), which must lie on a circle. The points must be different.

Names	Result	1st parameter	2nd parameter	3rd parameter	4th parameter	5th parameter	6th parameter
CALCPOSI	INT: Status 0 OK -1 DLIMIT neg. -2 Trans. n.def. 1 SW limit 2 Working area 3 Prot. zone See PGA for more	REAL: Starting position in WCS [0] Abscissa [1] Ordinate [2] Applicate	REAL: Increment: Path definition [0] Abscissa [1] Ordinate [2] Applicate referred to starting position	REAL: Minimum clearances of limits to be observed [0] Abscissa [1] Ordinate [2] Applicate [3] Lin. machine Axis [4] Rot. Axis	REAL: **Return value** possible incr. path if path from parameter 3 cannot be fully traversed without violating limit	BOOL: 0: Evaluation G code group 13 (inch/metr.) 1: Reference to basic control system, independent of active G codes group 13	bin encoded to be monitored 1 SW limits 2 working area 4 active protection zone 8 preactive protection zone
	Explanation: **CALCPOSI**	CALCPOSI is for checking whether, starting from a defined starting point, the geometry axes can traverse a defined path without violating the axis limits (software limits), working area limitations, or protection zones. If the defined path cannot be traversed without violating limits, the maximum permissible value is returned.					

INTERSEC	BOOL: Error status	VAR REAL [11]: First contour element	VAR REAL [11]: Second contour element	VAR REAL [2]: Result vector: Intersection coordinate, abscissa and ordinate	Intersection: Calculation of intersection The intersection between two contour elements is calculated. The intersection coordinates are return values. The error status indicates whether an intersection was found.

3. Axis functions				
	Result	**1st parameter**	**2nd parameter**	**Explanation**
AXNAME	AXIS: Axis identifier	STRING [ ]: Input string		AXNAME: Get axis identifier Converts the input string to an axis identifier. An alarm is generated if the input string does not contain a valid axis identifier.
AXTOSPI	INT: Spindle number	AXIS: Axis identifier		AXTOSPI: Convert axis to spindle Converts an axis identifier into a spindle number. An alarm is set if the transfer parameter does not contain a valid axis identifier.
SPI	AXIS: Axis identifier	INT: Spindle number		SPI: Convert spindle to axis Converts a spindle number to an axis identifier. An alarm is generated if the passed parameter does not contain a valid spindle number.
ISAXIS	BOOL TRUE: Axis exists: Otherwise: FALSE	INT: Number of the geometry axis (1 to 3)		Check whether the geometry axis 1 to 3 specified as parameter exists in accordance with $MC_AXCONF_GEOAX_ASSIGN_TAB.
AXSTRING	STRING	AXIS		Convert axis identifier into string.

4. Tool management				
	Result	**1st parameter**	**2nd parameter**	**Explanation**
NEWT #	INT: T number	STRING [32]: Tool name	INT: Duplo number	Create new tool (prepare tool data). The duplo number can be omitted.
GETT #	INT: T number	STRING [32]: Tool name	INT: Duplo number	Get T number for tool identifier
GETACTT #	INT: Status	INT: T number	STRING[32]: Tool name	Get active tool from a group of tools with the same name
TOOLENV	INT: Status	STRING: Name		Save a tool environment in SRAM with the specified name
DELTOOLENV	INT: Status	STRING: Name		Delete a tool environment in SRAM with the specified name. All tool environments if no name specified.

GETTENV	INT:
	Status
		STRING:
		Name
			INT:
			Number = [0]
			Number = [1]
			Number = [2]
				Reading:
				T number,
				D number,
				DL number
				from a tool environment with the specified name

#) The vocabulary word is not valid for SINUMERIK FM-NC.

	Result	1st par.	2nd par.	3rd par.	4th par.	5th par.	6th par.	Explanation
GETTCOR	INT: Status	REAL: Length [11]	STRING: Compon ents: Coordi-nate system	STRING: Tool environm ent/ " "	INT: Int. T number	INT: D number	INT: DL number	Read tool lengths and tool length components from tool environment or current environment Details: See Description of Functions W1

	Result	1st par.	2nd par.	3rd par.	4th par.	5th par.	6th par.	7th par.	8th par.	9th par.
SETTCOR	INT: Status	REAL: Offset vector [0-3]	STRING: Compo-nent(s)	INT: Compon ent(s) to be offset	INT: Type of write operation	INT: Index of geo. axis	STRING: Name of tool environm ent	INT: Int. T number	INT: D number	INT: DL number
Explanation	Changing tool components whilst observing all marginal conditions that are included in the evaluation of the individual components. Details: See Description of Functions W1									

	Result	1st parameter	2nd parameter	3rd parameter	Explanation
LENTOAX	INT: Status	INT: Axis index [0-2]	REAL: L1, L2, L3 for abscissa, ordinate, applicate [3], [3] Matrix	STRING: Coordinate system for the assignment	The function provides information about the assignment of the tool lengths L1, L2, L3 of the active tools to abscissa, ordinate, applicate. The assignment to the geometry axes is affected by frames and the active plane (G17 - 19). Details: See Description of Functions W1

5. Arithmetic				
	Result	1st parameter	2nd parameter	Explanation
SIN	REAL	REAL		Sine
ASIN	REAL	REAL		Arcsine
COS	REAL	REAL		Cosine
ACOS	REAL	REAL		Arccosine
TAN	REAL	REAL		Tangent
ATAN2	REAL	REAL	REAL	Arctangent 2
SQRT	REAL	REAL		Square root
POT	REAL	REAL		Square
TRUNC	REAL	REAL		Truncate decimal places
ROUND	REAL	REAL		Round decimal places
ABS	REAL	REAL		Generate absolute value
LN	REAL	REAL		Natural logarithm
EXP	REAL	REAL		Exponential function ex

6. String functions				
	Result	1st parameter	2nd parameter to 3rd parameter	Explanation
ISNUMBER	BOOL	STRING		Check whether the input string can be converted to a number. Result is TRUE if conversion is possible.
ISVAR	BOOL	STRING		Check whether the transfer parameter contains a variable known in the NC. (Machine data, setting data, system variable, general variables such as GUDs Result is TRUE if all the following checks produce positive results according to the (STRING) transfer parameter: – The identifier exists – It is a 1- or 2-dimensional array – An array index is allowed. For axial variables, the axis names are accepted as an index but not checked.
NUMBER	REAL	STRING		Convert the input string into a number.
TOUPPER	STRING	STRING		Convert all alphabetic characters in the input string to upper case.
TOLOWER	STRING	STRING		Convert all alphabetic characters in the input string to lower case.
STRLEN	INT	STRING		The result is the length of the input string up to the end of the string (0).
INDEX	INT	STRING	CHAR	Find the character (2nd parameter) in the input string (1st parameter). The reply gives the place, at which the character was first found. The search is from left to right. The 1st character in the string has the index 0.
RINDEX	INT	STRING	CHAR	Find the character (2nd parameter) in the input string (1st parameter). The reply gives the place, at which the character was first found. The search is from right to left. The 1st character in the string has the index 0.
MINDEX	INT	STRING	STRING	Find one of the characters specified in the 2nd parameter in the input string (1st parameter). The place where one of the characters was first found is output. The search is from left to right. The first character in the string has the index 0.
SUBSTR	STRING	STRING	INT	Returns the substring of the input string (1st parameter), defined by the start character (2nd parameter) and number of characters (3rd parameter). Example: SUBSTR("Hello world",1.5) returns "ello"

12.4.4 Data types

Data types

Data types		
Type	**Note**	**Value range**
INT	Integers with sign	± (2^{31} -1)
REAL	Real numbers (fractions with decimal point, LONG REAL to IEEE)	± (10^{-300} ... 10^{+300})
BOOL	Boolean value TRUE, FALSE or 1, 0	1, 0
CHAR	1 character in ASCII code	0 ... 255
STRING	Character string, number of characters in [...] (max. of 200 characters)	Sequence of values with 0 ... 255
AXIS	Axis names (axis addresses) only	Any axis identifiers in the channel
FRAME	Geometrical parameters for translation, rotation, scaling, mirroring	—

List of abbreviations

A

A	Output
AS	Automation system
ASCII	American Standard Code for Information Interchange: American coding standard for the exchange of information
ASIC	Application Specific Integrated Circuit: User switching circuit
ASUB	Asynchronous subroutine
AV	Job planning
STL	Statement list
BA	Operating mode
Mode group	Mode group
BB	Ready to run
HMI	Human Machine Interface
BCD	Binary Coded Decimals: Decimal numbers encoded In binary code
HHU	Handheld unit
BIN	Binary files (Binary Files)
BIOS	Basic Input Output System
BCS	Basic Coordinate System
UI	User interface
BOT	Boot files: Boot files for SIMODRIVE 611 digital
OP	Operator Panel
OPI	Operator Panel Interface
CAD	Computer-Aided Design
CAM	Computer-Aided Manufacturing
CNC	Computerized Numerical Control: Computerized numerical control
COM	Communication
CP	Communications Processor
CPU	Central Processing Unit: Central processing unit
CR	Carriage Return
CRT	Cathode Ray Tube picture tube
CSB	Central Service Board: PLC module
CTS	Clear To Send: Signal from serial data interfaces
CUTOM	Cutter radius compensation: Tool radius compensation
DAC	Digital-to-Analog Converter
DB	Data block in the PLC
DBB	Data block byte in the PLC
DBW	Data block word in the PLC
DBX	Data block bit in the PLC

DC	Direct Control: Movement of the rotary axis via the shortest path to the absolute position within one revolution
DCD	Data Carrier Detect
DDE	Dynamic Data Exchange
DTE	Data Terminal Equipment
DIN	Deutsche Industrie Norm (German Industry Standard)
DIO	Data Input/Output: Data transfer display
DIR	Directory: Directory
DLL	Dynamic Link Library
DOE	Data transmission equipment
DOS	Disk Operating System
DPM	Dual-Port Memory
DPR	Dual-Port RAM
DRAM	Dynamic Random Access Memory
DRF	Differential Resolver Function: Differential resolver function (DRF)
DRY	Dry Run: Dry run feedrate
DSB	Decoding Single Block: Decoding single block
DW	Data word
E	Input
I/O	Input/Output
I/R	Infeed/regenerative-feedback unit (power supply) of the SIMODRIVE 611digital
EIA code	Special punched tape code, number of holes per character always odd
ENC	Encoder: Actual value encoder
EPROM	Erasable Programmable Read Only Memory
Error	Error from printer
FB	Function block
FBS	Slimline screen
FC	Function Call: Function block in the PLC
FDB	Product database
FDD	Floppy Disk Drive
FEPROM	Flash-EPROM: Read and write memory
FIFO	First In First Out: Memory that works without address specification and whose data are read in the same order in which they were stored.
FIPO	Fine InterPOlator
FM	Function Module
FM-NC	Function module – numerical control
FPU	Floating Point Unit Floating Point Unit
FRA	Frame block
FRAME	Data record (frame)
CRC	Cutter radius compensation
FST	Feed Stop: Feed stop
CSF	Function plan (PLC programming method)

BP	Basic program
GUD	Global User Data: Global user data
HD	Hard Disk Hard disk
HEX	Abbreviation for hexadecimal number
AuxF	Auxiliary function
HMI	Human Machine Interface: Operator functionality of SINUMERIK for operation, programming and simulation.
HMS	High-resolution Measuring System
MSD	Main Spindle Drive
HW	Hardware
IBN	Startup
IF	Drive module pulse enable
IK (GD)	Implicit communication (global data)
IKA	Interpolative Compensation: Interpolatory compensation
IM	Interface Module Interconnection module
IMR	Interface Module Receive: Interconnection module for receiving data
IMS	Interface Module Send: Interconnection module for sending data
INC	Increment: Increment
INI	Initializing Data: Initializing data
IPO	Interpolator
ISA	Industry Standard Architecture
ISO	International Standardization Organization
ISO code	Special punched tape code, number of holes per character always even
JOG	Jogging: Setup mode
K1 .. K4	Channel 1 to channel 4
C Bus	Communication bus
COR	Coordinate rotation
LAD	Ladder diagram (PLC programming method)
K_v	Servo gain factor
K_{UE}	Speed ratio
LCD	Liquid Crystal Display: Liquid crystal display
LED	Light-Emitting Diode: Light emitting diode
LF	Line Feed
PMS	Position measuring system
LR	Position controller
LUD	Local User Data
MB	Megabyte
MD	Machine data
MDI	Manual Data Automatic: Manual input
MC	Measuring circuit
MCS	Machine coordinate system
MLFB	Machine-readable product designation
MPF	Main Program File: NC parts program (main program)

MPI	Multiport Interface Multiport Interface
MS	Microsoft (software manufacturer)
MCP	Machine control panel
NC	Numerical Control: Numerical Control
NCK	Numerical Control Kernel: NC kernel with block preparation, traversing range, etc.
NCU	Numerical Control Unit: Hardware unit of the NCK
NRK	Name for the operating system of the NCK
IS	Interface signal
NURBS	Non-Uniform Rational B-Spline
ZO	Work offset
OB	Organization block in the PLC
OEM	Original Equipment Manufacturer
OP	Operator Panel: Operating setup
OPI	Operator Panel Interface: Interface for connection to the operator panel
OPT	Options: Options
OSI	Open Systems Interconnection: Standard for computer communications
P bus	Peripheral Bus
PC	Personal Computer
PCIN	Name of the SW for data exchange with the control
PCMCIA	Personal Computer Memory Card International Association: Standard for plug-in memory cards
PCU	PC Unit: PC box (computer unit)
PG	Programming device
PLC	Programmable Logic Control: Interface control
POS	Positioning
RAM	Random Access Memory: Program memory that can be read and written to
REF	Reference point approach function
REPOS	Reposition function
RISC	Reduced Instruction Set Computer: Type of processor with small instruction set and ability to process instructions at high speed
ROV	Rapid override: Input correction
RPA	R-Parameter Active: Memory area on the NCK for R parameter numbers
RPY	Roll Pitch Yaw: Rotation type of a coordinate system
RTS	Request To Send: RTS, control signal of serial data interfaces
SBL	Single Block: Single block
SD	Setting Data
SDB	System Data Block
SEA	Setting Data Active: Identifier (file type) for setting data
SFB	System Function Block
SFC	System Function Call
SK	Softkey

SKP	SKiP: Skip block
SM	Stepper Motor
SPF	Sub Routine File: Subroutine
PLC	Programmable Logic Controller
SRAM	Static RAM (non-volatile)
TNRC	Tool Nose Radius Compensation
LEC	Leadscrew error compensation
SSI	Serial Synchronous Interface: Synchronous serial interface
SW	Software
SYF	System Files System files
TEA	Testing Data Active: Identifier for machine data
TO	Tool Offset: Tool offset
TOA	Tool Offset Active: Identifier (file type) for tool offsets
TRANSMIT	TRANSform Milling Into Turning: Coordinate conversion on turning machine for milling operations
UFR	User Frame: Work offset
SR	Subroutine
FDD	Feed Drive
RS-232-C	Serial interface (definition of the exchange lines between DTE and DCE)
WCS	Workpiece coordinate system
T	Tool
TLC	Tool length compensation
WOP	Workshop-oriented Programming
WPD	Workpiece Directory: Workpiece directory
TRC	Tool Radius Compensation
TO	Tool offset
TC	Tool change
ZOA	Zero Offset Active: Identifier (file type) for zero offset data
µC	Micro Controller

Glossary

Absolute dimensions

A destination for an axis movement is defined by a dimension that refers to the origin of the currently active coordinate system. See -> incremental dimension.

Acceleration with jerk limitation

In order to optimize the acceleration response of the machine whilst simultaneously protecting the mechanical components, it is possible to switch over in the machining program between abrupt acceleration and continuous (jerk-free) acceleration.

Access authorization

The CNC program blocks and data are protected via a 7-stage access authorization procedure.

- three password levels for system manufacturers, machine manufacturers and users, plus
- four keyswitch positions which can be evaluated via the PLC.

Address

An address is the identifier for a certain operand or operand range, e.g. input, output etc.

Analog input/output module

Analog input/output modules are signal formers for analog process signals.

Analog input modules convert analog measured values into digital values which can be processed in the CPU.

Analog output modules convert digital values into analog output signals.

Approach machine fixed-point

Approach motion towards one of the predefined -> fixed machine points.

Archiving

Reading out data and/or directories to an external memory device.

A-Spline

The Akima-Spline runs under a continuous tangent through the programmed interpolation points (3rd order polynomial).

Asynchronous subroutine

A parts program which can be started asynchronously to (independently of) the current program status by an interrupt signal (e.g. "rapid NC input" signal).

Automatic

Operating mode of the control (block sequence operation according to DIN): Operating Mode in NC systems in which a -> parts program is selected and continuously executed.

Auxiliary functions

Auxiliary functions can be used to transfer -> parameters to the -> PLC in -> parts programs, where they trigger reactions which are defined by the machine manufacturer.

Axes

In accordance with their functional scope, the CNC axes are subdivided into:

- Axes: interpolating path axes
- Auxiliary axes: non-interpolating feed and positioning axes with an axis-specific feed rate. Auxiliary axes are not involved in the actual machining, and include for example tool feeders and tool magazines.

Axis address

See -> axis identifier

Axis identifier

Axes are labeled in accordance with DIN 66217 (for a clockwise orthogonal -> coordinate system) with the letters X,Y, Z.

-> Rotary axes which rotate around are labeled with the letters A, B, C. Additional axes parallel to the above can be identified with further address letters.

Axis name

See -> axis identifier

B spline

With the B-Spline, the programmed positions are not interpolation points, as they are just "control points" instead. The generated curve only runs near to the control points, not directly through them (optional 1st, 2nd or 3rd order polynomials).

Backlash compensation

Compensation for mechanical machine backlash, e.g. backlash on reversal for feed screws. Backlash compensation can be entered separately for each axis.

Backup

Saving the memory contents to an external memory device.

Backup battery

The backup battery ensures that the -> user program is reliably backed up in the -> CPU against mains failure and that fixed data areas and markers, times and counters are kept in non-volatile memory.

Back-up memory

The backup memory enables buffering of memory areas of the -> CPU without a buffer battery. Buffering can be performed for a configurable number of times, counters, markers and data bytes.

Basic axis

Axis whose setpoint or actual value position forms the basis of the calculation of a compensation value.

Basic Coordinate System

Cartesian coordinate system which is mapped by transformation onto the machine coordinate system.

In the -> parts program, the programmer uses the axis names of the basic coordinate system. The basic coordinate system exists in parallel to the -> machine coordinate system when no -> transformation is active. The difference between the systems relates to the axis identifiers.

Baud rate

Rate of data transfer (Bit/s).

Blank

Workpiece as it is before a part is machined.

Block

"Block" is the term given to any files required for creating and processing programs.

A section of a -> parts program terminated with a line feed. A distinction is made between -> main blocks and -> subblocks.

Block search

For testing part programs or following interruption of machining, it is possible to select any point in the part program using the block search function in order to start or resume at this point.

Booting

Loading the system program after power on.

Bus connector

A bus connector is an S7-300 accessory part which is supplied together with the -> I/O modules. The bus connector expands the -> S7-300 bus from the -> CPU or an I/O module to the neighboring I/O module.

C axis

Axis around which the tool spindle describes a controlled rotational and positioning movement.

C spline

The C-spline is the most well-known and widely used spline. The transitions at the interpolation points are continuous, both tangentially and in terms of curvature. 3rd order polynomials are used.

Channel

A channel is characterized by its ability to execute a -> parts program independently of other channels. A channel exclusively controls the axes and spindles assigned to it. Parts programs run on various channels can be coordinated by -> synchronization.

Channel structure

The channel structure enables the -> programs of the individual channels to be executed simultaneously and asynchronously.

Circular interpolation

The -> tool is required to travel in a circle between defined points on the contour at a specified feedrate while machining the workpiece.

CNC

See -> NC

CNC programming language

The CNC programming language is based on DIN 66025 with high-level expansions. The -> high-level CNC language and programming allow, among other things, macros to be defined (groupings of individual instructions).

COM

Component of the NC control for the implementation and coordination of communication.

Compensation axis

Axis with a setpoint or actual value modified by the compensation value

Compensation table

Table containing interpolation points. It provides the compensation values of the compensation axis for selected positions on the basic axis.

Compensation value

Difference between the axis position measured by the position sensor and the desired, programmed axis position.

Continuous-path mode

The purpose of continuous-path mode is to prevent excessive deceleration of the -> path axes at the part program block boundaries (in terms of the control, machine and other properties of the operation and the user) and to effect the transition to the next block at as uniform a path speed as possible.

Contour

Outline of the -> workpiece

Contour monitoring

The following error is monitored within a defined tolerance band to ensure contour precision. An impermissibly high following error might be caused by a drive overload, for example. In this case an alarm is triggered and the axes are stopped.

coordinate system

See -> Machine Coordinate System, -> Workpiece Coordinate System

CPU

Central Processor Unit, see -> Programmable Logic Controller

Cycle

Protected subroutine for executing a repetitive machining process on the -> workpiece.

Cycle support

In the "Program" operating area, the available cycles are listed under the menu "Cycle Support". After selecting the desired machining cycle the required parameters for the value assignment are displayed in clear text.

Data Block

1. Data unit of the -> PLC, which the -> HIGHSTEP programs can access.
2. Data unit of the -> NC: Data blocks contain data definitions for global user data. These data can be initialized directly when they are defined.

Data transmission program PCIN

PCIN is an auxiliary program which is used to send and receive CNC user data via the serial interface, such as e.g. parts programs, tool offsets etc. The PCIN program can be executed under MS-DOS on standard industrial PCs.

Data word

A data unit, two bytes in size, within a -> data block.

Diagnosis

1. Control operating area
2. The control has both a self-diagnostics program and testing aids for service. Status, alarm and service indicators.

Digital input/output module

Digital modules are signal formers for binary process signals.

Dimensions in metric units and inches

Position and gradient values can be entered in the machining program in inches. The control can be set to a basic system regardless of the programmed measuring system (G70/G71).

DRF

Differential Resolver Function: An NC function which generates an incremental zero offset in automatic mode in conjunction with an electronic handwheel.

Drive

The SINUMERIK 840D control system is connected to the SIMODRIVE 611 digital converter system by means of a high-speed digital parallel bus.

Dynamic feedforward control

Inaccuracies in the contour caused by following errors can be almost entirely eliminated with the aid of dynamic, acceleration-dependent feedforward control. The result is extraordinary machining precision even at high tool path feed-rates. The feedforward control can be individually selected and deselected for each axis in the parts program.

Editor

The editor is used to create, modify, add to, compress, and insert programs/texts/program blocks.

Electronic handwheel

The electronic handwheels can be used to simultaneously traverse selected axes manually. The meaning of the lines on the handwheels is defined by the external zero offset increment weighting.

Exact stop

With a programmed exact stop instruction, the position stated in a block is approached precisely and very slowly, if necessary. In order to reduce the approach time, -> exact stop limits are defined for rapid traverse and feed.

Exact stop limit

When all path axes reach their exact stop limits, the control responds as if it had reached its destination point precisely. The -> part program continues execution at the next block.

External zero offset

Zero offset specified by the -> PLC.

Fast retraction from contour

When an interrupt arrives a motion can be initiated via the CNC machining program which allows the tool to be quickly retracted from the workpiece contour currently being machined. The retraction angle and the distance retracted can also be parameterized. An interrupt routine can also be executed following the fast retraction of the tool. (SINUMERIK 840D).

Feed override

The programmed velocity is overridden by the current velocity setting via the machine control panel or by the PLC (0-200 %). The feedrate can also be corrected by a programmable percentage factor (1 - 200%) in the machining program.

Finished-part contour

Contour of the finished workpiece. See -> blank.

Fixed machine point

A point defined uniquely by the machine tool, e.g. the reference point.

Fixed-point approach

Machine tools can approach fixed points such as a tool change point, loading point, pallet change point, etc. in a defined way. The coordinates of these points are stored in the control. Where possible, the control moves these axes in -> rapid traverse.

Frame

A frame is an arithmetic rule that transforms one Cartesian coordinate system into another Cartesian coordinate system. A frame contains the components -> zero offset, -> rotation, -> scaling, -> mirroring.

Geometry

Description of a -> workpiece in the -> workpiece coordinate system.

geometry axis

Geometry axes are used to describe a 2- or 3-dimensional range in the workpiece coordinate system.

Global main program/subroutine

Every global main program/subroutine can only appear once under its own name in the directory, and it is not possible to have the same program name in different directories with different contents as a global program.

Ground

Ground is taken as the total of all linked inactive parts of a device which will not become live with a dangerous contact voltage even in the event of a malfunction.

Helical interpolation

Helical interpolation is especially suitable for easy machining inside or outside threads with form cutters and for milling lubrication grooves. The helix consists of two motions:

1. A circular movement in one plane
2. A linear movement perpendicular to this plane

High-level CNC language

The high-level language offers: -> User-defined variable, -> System variable, -> Macro technique.

High-speed digital inputs/outputs

The digital inputs can be used for example to start fast CNC program routines (interrupt routines). The digital CNC outputs can be used to trigger fast, program-controlled switching functions (SINUMERIK 840D).

HIGHSTEP

Summary of the programming options for the -> PLC in the AS300/AS400 system.

I/O module

I/O modules represent the link between the CPU and the process. I/O modules are:

- ->Digital input/output modules
- ->Analog input/output modules
- ->Simulator modules

Inch system

Dimension system which defines distances in inches and fractions of inches.

Inclined surface machining

Drilling and milling operations on workpiece surfaces that do not lie in the coordinate planes of the machine can be performed easily using the function "inclined-surface machining".

Increment

Traversed distance information via the number of increments. The number of increments can be stored as -> setting data or selected using keys labeled with 10, 100, 1000, 10 000.

Incremental dimension

Also incremental dimension: A destination for axis traversal is defined by a distance to be covered and a direction referenced to a point already reached. See -> Absolute dimension.

Initialization block

Initialization blocks are special -> program blocks. They contain value assignments that are performed before program execution. The primary purpose of initialization blocks is to initialize predefined data or global user data.

Initialization files

It is possible to create an initialization file for each -> workpiece. Various variable assignments which are intended to apply specifically to one workpiece can be stored in this file.

Interconnecting cables

Connecting cables are pre-assembled or user-assembled 2-wire cables with a connector at each end. These interconnecting cables connect the -> CPU via the -> multipoint interface (MPI) with a -> programming device or other CPUs.

Intermediate blocks

Traversing movements with tool offset selected (G41/G42) can be interrupted by a limited number of intermediate blocks (block without axis movements in the compensation plane) whereby the tool offset can still be correctly calculated. The permissible number of intermediate blocks which the control reads ahead can be set in system parameters.

Interpolator

Logical unit of the -> NCK which determines intermediate values for the movements to be traversed on the individual axes on the basis of destination positions specified in the parts program.

Interpolatory compensation

The interpolatory compensation allows manufacturing related **L**eadscrew **E**rror **C**ompensation and **M**easuring **S**ystem **E**rror **C**ompensation (LEC, MSEC).

interrupt routine

Interrupt routines are special -> subroutines which can be started on the basis of events (external signals) in the machining process. A parts program block which is currently being worked through is interrupted and the position of the axes at the point of interruption is automatically saved.

Interrupts

All alarms and -> messages are output on the operator panel in plain text with the date and time and a symbol indicating the cancel criterion. The display is divided into alarms and messages.

1. Alarms and messages in the part program:

   Alarms and messages can be displayed in plain text directly from the part program.

2. Alarms and messages from PLC

   Alarms and messages for the machine can be displayed in plain text from the PLC program. No additional function block packages are required to do this.

Inverse time feedrate

With the SINUMERIK 840D, it is possible to program the time required to traverse the path of a block instead of programming the feedrate for the axis movement (G93).

Jog

Control operating mode (setup mode): In JOG mode, it is possible to set up the machine. Individual axes and spindles can be moved in this mode using the direction keys. Other functions available in JOG mode are -> reference point approach, -> repositioning and -> preset (setting an actual value).

Key switch

1. S7-300: The keyswitch is the operating mode switch of the -> CPU. A removable key is used to operate the keyswitch.
2. 840D: The keyswitch on the -> machine control panel has 4 settings, to which functions are assigned by the operating system of the control. Three differently colored keys also belong to the keyswitch. These keys can be pulled out of the keyswitch in the indicated positions.

Keywords

Words with a specific notation, which have a defined meaning in the programming language for -> parts programs.

Kv

Servo gain factor, a control variable in a control loop.

Leadscrew error compensation

Compensation for the mechanical inaccuracies of a leadscrew participating in the feed. The control uses stored deviation values for the compensation.

Limit speed

Maximum/minimum (spindle) speed: The maximum speed of a spindle may be limited by values defined in the machine data, the -> PLC or -> setting data.

Linear axis

The linear axis is an axis which, in contrast to a rotary axis, describes a straight line.

Linear interpolation

The tool travels along a straight line to the destination point while machining the workpiece.

Load memory

For the CPU 314 of the -> PLC, the load memory is equal to the -> Work memory.

Look ahead

With the look ahead function, a configurable number of traversing blocks is read in advance in order to calculate the optimum machining velocity.

Look-ahead detection of contour violations

The control can recognize and signal the following types of collision:

1. The path distance is shorter than the tool radius.
2. The width of the inner corner is smaller than the tool diameter.

Machine

Control operating area

Machine axes

Axes which exist physically on the machine tool.

Machine control panel

An operator panel on a machine tool with operating elements such as keys, rotary switches etc. and simple indicators such as LEDs. It is used to control the machine tool directly via the PLC.

Machine coordinate system

System of coordinates based on the axes of the machine tool.

Machine zero

A fixed point on the machine tool, which can be referenced by all (derived) measuring systems.

Machining channel

Via a channel structure, parallel sequences of movements, such as positioning a loading gantry during machining, can shorten unproductive times. Here, a CNC channel must be regarded as a separate CNC control system with decoding, block preparation and interpolation.

Macro techniques

Grouping of a set of instructions under a single identifier. The identifier in the program refers to the grouped set of instructions.

Main block

A block prefixed by ":" containing all the parameters required to start execution of a -> parts program.

Main program

Parts program identified by a number or identifier in which further main programs, subroutines or -> cycles may be called.

Mains

The term "network" describes the connection of several S7-300 and other terminal devices, e.g. a programming device, via -> interconnecting cables. A data exchange takes place over the network between the connected devices.

MDI

Control operating mode: Manual Data Automatic. In MDA mode, it is possible to enter individual program blocks or sequences of blocks without reference to a main program or subroutine and to then execute them immediately via the NC start key.

Messages

All messages programmed in the parts program and -> alarms recognized by the system are output on the operator panel in plain text with the date and time and a symbol indicating the cancel criterion. The display is divided into alarms and messages.

Metric system

Standardized measuring system: for lengths in millimeters (mm), meters (m), etc.

Mirroring

Mirroring inverts the signs of the coordinate values of a contour with respect to an axis. It is possible to mirror in relation to more than one axis at a time.

Mode group

At any one time, all axes/spindles are assigned to just one channel. Each channel is assigned to a mode group. The same -> mode is always assigned to the channels in a mode group.

Mounting rail

A mounting rail is used to attach the modules of an S7-300.

Multipoint interface

The multipoint interface (MPI) is a 9-pole Sub-D interface. A configurable number of devices can be connected to a multipoint interface and then communicate with each other.

- Programming devices
- Operator control and monitoring equipment
- Further automation systems

The parameter block "Multipoint Interface MPI" of the CPU contains the -> parameters which define the properties of the multipoint interface.

Name of identifier

The words according to DIN 66025 are supplemented by the identifiers (names) for variables (computer variable, system variable, user variable), for subroutines, for keywords and words with several address letters. In terms of the block format, these supplements have the same significance as the words. Identifiers must be unique. The same identifier must not be used for different objects.

NC

Numerical Control: NC control incorporates all the components of the of the machine tool control system: -> NCK, -> PLC, HMI, -> COM.

Note

CNC (Computerized Numerical Control) is a more accurate term for the SINUMERIK 840D controls. MARS and Merkur controls.

NCK

Numerical Control Kernel: Component of the NC control which executes -> parts programs and essentially coordinates the movements on the machine tool.

Node number

The node number represents the "contact address" of a -> CPU or the -> programming device or any other intelligent periphery module if these are communicating via a -> network with each other. The node number is assigned to the CPU or the programming device with the S7 tool -> "S7 configuration".

NRK

Numeric Robotic Kernel (operating system of the -> NCK)

NURBS

Internal motion control and path interpolation are performed using NURBS (non-uniform rational B-splines). This provides a uniform internal method for all interpolations in the control (SINUMERIK 840D).

OEM

For machine manufacturers who manufacture their own user interface or wish to integrate their own technology-specific functions in the control, free space has been left for individual solutions (OEM applications) for SINUMERIK 840D.

Offset memory

Data range in the control in which the tool offset data are stored.

Operating mode

An operating concept on a SINUMERIK control. The operating modes -> Jog, -> MDA and -> Automatic are defined.

Oriented spindle stop

Stops the workpiece spindle with a specified orientation angle, e.g. to perform an additional machining operation at a specific position.

Oriented tool retraction

RETTOOL: If machining is interrupted (because of tool breakage, for example), a program command can be used retract the tool with a defined orientation by a defined path.

Overall reset

During a general reset the following memories of the -> CPU are deleted:

- the -> working memory
- the read/write area of the -> load memory
- the -> system memory
- the -> backup memory

Override

Manual or programmable control feature which enables the user to override programmed feedrates or speeds in order to adapt them to a specific workpiece or material.

Parameter

- S7-300: a distinction is made between 2 types of parameters:
  - Parameters of a STEP 7 instruction

    A parameter of a STEP 7 instruction is the address of the operand to be processed or a constant.

    Parameters of a -> parameter block

    A parameter of a parameter block determines the behavior of a module.
- 840D:
  - Control operating area
  - Arithmetic parameter for which the programmer of the parts program can assign or request values as required in the program.

Part program

A sequence of instructions to the NC control which combine to produce a specific -> workpiece. Likewise, performing a certain machining operation on a specific -> blank.

Part program management

Part programs can be organized according to -> workpieces. The size of the user memory determines the number of programs and the amount of data that can be managed. Each file (programs and data) can be assigned a name comprising up to 24 alphanumeric characters.

Path axis

Path axes are all the machining axes in the -> channel which are controlled by the -> interpolator so that they start, accelerate, stop and reach their end positions simultaneously.

Path feed

Path feed acts on -> path axes. It represents the geometrical sum of the feeds on the participating -> geometry axes.

Path velocity

The maximum programmable path velocity depends on the input resolution. For example, with a resolution of 0.1 mm the maximum programmable path velocity is 1000 m/min.

PG

Programming device

PLC

Programmable Logic Control: -> Programmable Logic Controller. Component of the -> NC: Programmable controller for processing the control logic of the machine tool.

See -> Programmable Logic Controller

PLC program memory

SINUMERIK 840D: PLC user memory is used to store the PLC user program and the user data together with the PLC basic program. The PLC user memory can be upgraded to 96 kByte with memory expansions.

PLC Programming

The PLC is programmed using the STEP 7 software. The STEP 7 programming software is based on the WINDOWS operating system, and combines the STEP 5 programming functions with additional innovative functional developments.

Polar coordinates

A coordinate system, which defines the position of a point on a plane in terms of its distance from the zero point and the angle formed by the radius vector with a defined axis.

Polynominal interpolation

With polynomial interpolation, it is possible to generate many different curve characteristics, such as **straight line-, parabolic-, and exponential functions** (SINUMERIK 840D).

Position axis

Axis which performs an auxiliary movement on a machine tool (e.g. tool magazine, pallet transport). Positioning axes are axes that do not interpolate with -> path axes.

Power on

Switching the control off and back on again.

Preset

The preset function can be used to redefine the control zero in the machine coordinate system. Preset does not move the axes, but a new position value is entered for the current axis positions.

Program

1. Control operating area
2. Sequence of instructions to the control.

Program block

Program blocks contain the main and subroutines of the -> part programs.

Programmable frames

Programmable -> frames can be used to define new coordinate system starting points dynamically while the parts program is running. A distinction is made between absolute definition using a new frame and additive definition with reference to an existing starting point.

Programmable Logic Controller

Programmable logic controllers (PLC) are electronic controls, the function of which is stored as a program in the control unit. This means that the layout and wiring of the device do not depend on the function of the control. The programmable logic controller has the same structure as a computer; it consists of a CPU (central module) with memory, input/output modules and an internal bus system. The peripherals and the programming language are matched to the requirements of the control technology.

Programmable working area limitation

Limitation of the motion space of the tool to a space defined by programmed limitations.

Programming key

Characters and character sequences, which have a defined meaning in the programming language for -> parts programs (see programming guide).

Protection zone

Three-dimensional space within the -> working area which the tool tip is not permitted to enter.

Quadrant error compensation

Contour errors at quadrant transitions, which arise as a result of changing friction conditions on the guideways, can be virtually entirely eliminated with the quadrant error compensation. Parameterization of the quadrant error compensation is performed by means of a circuit test.

R parameters

Arithmetic parameter for which the programmer of the -> parts program can freely assign or request values in the program.

Rapid traverse

The highest speed of an axis. It is used for example to move the tool from rest position to the -> workpiece contour or retract the tool from the contour.

Reference point

Point on the machine tool used to reference the measuring system of the -> machine axes.

Reference point approach

If the utilized distance measuring system is not an absolute value encoder then it is necessary to perform a reference point approach to ensure that the actual values returned by the measuring system match the machine coordinate values.

REPOS

1. Repositioning on the contour using operator input

   The REPOS function can use the direction keys to reposition at the point of interruption.

2. Repositioning on the contour by program

   The program commands provide various approach strategies: approaching the interruption point, approaching the block starting point, approaching the block end point, approaching a point on the path between the start of the block and the interruption point.

Retentive memory

Data areas in data blocks and times, counters and markers are retentive (non-volatile) if their contents are not lost when the system is restarted or the mains supply is disconnected.

Rotary axis

Rotary axes rotate a workpiece or tool to a defined angular position.

Rotary axis, continuously turning

Depending on the application, the traversing range of a rotary axis can be selected to be limited to less than 360 degrees or to be endlessly turning in both directions. Endlessly-turning rotary axes are used for non-circular turning, grinding, and winding.

Rotation

Component of a -> frame which defines a rotation of the coordinate system through a specific angle.

Rounding axis

Rounding axes rotate a workpiece or tool to an angular position corresponding to an indexing grid. When a grid index is reached, the rounding axis is "in position".

S7-300 bus

The S7-300 bus is a serial data bus by means of which the modules communicate with each other and are supplied with the required voltage. The connections between the modules are made with the -> bus connectors.

S7-Configuration

S7 configuration is a tool with the aid of which modules can be parameterized. With S7 configuration, various -> parameter blocks of the -> CPU and the I/O modules are set on -> PG. These parameters are transmitted to the CPU.

Safety functions

The controls contain watchdog monitors which are always active. These monitors detect problems in the CNC, PLC or machine in time to prevent damage to workpiece, tool or machine as far as possible. In the event of a malfunction the machining sequence is interrupted and the drives are stopped, the cause of the malfunction is saved and displayed as an alarm. At the same time, the PLC is informed that a CNC alarm is pending.

Scaling

Component of a -> frame, which causes axis-specific scale modifications.

Serial RS-232-C interface

For data input/output, there is

- a serial RS-232-C interface on the PCU 20, on the
- PCU 50/70, two RS-232-C interfaces

are available. Machining programs and manufacturer and user data can be loaded and saved via these interfaces.

Services

Control operating area

Setting data

Data which communicate properties of the machine tool to the NC control in a way defined by the system software.

Simulator module

A simulator module is a module

- on which digital input variables can be simulated via control elements and
- digital output variables can be displayed.

Softkey

A key whose name appears on an area of the screen. The selection of keys displayed is adapted dynamically to the operating situation. The freely assignable function keys (softkeys) are assigned defined functions in the software.

Software limit switch

Software limit switches limit the traversing range of an axis and prevent the slide from striking the hardware limit switch. 2 pairs of values can be assigned on each axis, which can then be activated separately via the -> PLC.

Spindle

The spindle functionality is divided into two power ranges.

1. Spindles: speed or position-controlled spindle drives, digital (SINUMERIK 840D)
2. Auxiliary spindles: speed-controlled spindle drives, "auxiliary spindle" function package e.g. for driven tools.

Spline interpolation

Using spline interpolation, the control can obtain a very smooth curve from just a few defined interpolation points along a set contour.

SRT

Speed ratio

Standard cycles

Standard cycles are available for frequently recurring machining tasks:

- for drilling/milling technology
- for turning technology

In the "Program" operating area, the available cycles are listed under the menu "Cycle Support". After selecting the desired machining cycle the required parameters for the value assignment are displayed in clear text.

Subblock

Block prefixed by "N" containing information for a machining step such as position data.

Subroutine

A sequence of instructions in a -> parts program, which can be called repeatedly with various defining parameters. The subroutine is called from a main program. It is not possible to block every subroutine against unauthorized reading and displaying. -> Cycles are a form of subroutine.

Synchronization

Instructions in -> parts programs for coordination of sequences in different -> channels at specific machining points.

Synchronized actions

1. Auxiliary function output

   During the workpiece machining, technology functions (-> auxiliary functions) can be issued from the CNC program to the PLC. These auxiliary functions are used for example to control additional equipment for the machine tool, such as quills, grabbers, clamping chucks etc.

2. Fast auxiliary function output

   For switching functions which are time-critical, the confirmation times for the -> auxiliary functions are minimized, and unnecessary stopping points in the machining process can be avoided.

Synchronized axes

Synchronized axes take the same time to traverse as the geometry axes take for their path.

System memory

The system memory is a memory in the CPU in which the following data are stored:

- Data required by the operating system
- The operands times, counters, markers

system variables

A variable, which exists although it has not been programmed by the -> parts program programmer. It is defined by a data type and the variable name preceded by the character **$**. See -> user-defined variable.

TappingRigid

This function allows threads to be tapped without a compensating chuck. By using the method whereby the spindle, as a rotary axis, and the drilling axis interpolate, threads can be cut to a precise final drilling depth (e.g. for blind hole threads) (requirement: spindles in axis operation).

Teach In

Teach In can be used to create or correct parts programs. The individual program blocks can be input via the keyboard and traversed immediately. Positions approached via the direction keys or the handwheel can also be saved. Additional details such as G functions, feeds or M functions can be input in the same block.

Text editor

See -> Editor

Tool

A tool is a software tool for inputting and changing the -> parameters of a parameter block. Tools include:

- -> S7 configuration
- S7-TOP
- S7-Info

Tool Nose Radius Compensation

Contour programming assumes that the tool is pointed. Because this is not actually the case in practice, the curvature radius of the tool used must be communicated to the control which then takes it into account. The curvature center is maintained equidistantly around the contour offset by the radius of curvature.

Tool offset

By programming a **T function** (5 integer decades) in the block, you can select the tool. Up to 9 cutting edges (D addresses) can be assigned to every T number. The number of tools to be managed in the controller is specified in the configuring data.

Tool radius compensation

In order to be able to program a desired -> workpiece contour directly, the control must traverse a path equidistant to the programmed contour, taking into account the radius of the tool used (G41/G42).

Tools

A part used on the machine tool for machining. Examples of tools include lathe tools, milling cutters, drills, laser beams, etc.

Transformation

Programming in a Cartesian coordinate system, execution in a non-Cartesian coordinate system (e.g. with machine axes as rotary axes).

Traversing range

The maximum permissible traversing range on linear axes is ± 9 decades. The absolute value depends on the selected input sensitivity and positioning resolution and the measuring system used (inches or metric).

User interface

The user interface (UI) is the display medium for a CNC control in the form of a screen. It is laid out with horizontal and vertical softkeys.

User memory

All program and data, such as part programs, subroutines, comments, tool compensations, and work offsets/frames, as well as channel- and program user data can be stored in the shared CNC user memory.

User program

User programs for the S7-300 automation systems are created using the programming language STEP 7. The user program has a modular layout and consists of individual blocks.

The basic block types are:
code modules: these blocks contain the STEP 7 commands.
Data blocks: these blocks contain the constants and variables for the STEP 7 program.

User-defined variable

The user can declare user-defined variables for any use in the -> parts program or data block (global user data). A definition contains a data type specification and the variable name. See -> system variable.

Variable definition

A variable definition includes the specification of a data type and a variable name. The variable names can be used to access the value of the variables.

Velocity control

In order to be able to achieve an acceptable traversing velocity on very short traverse movements within a single block, predictive velocity control can be set over several blocks (-> look ahead).

Work offset

Specification of a new reference point for a coordinate system through reference to an existing zero point and a -> frame.

1. Adjustable

   SINUMERIK 840D: A configurable number of adjustable zero offsets is available for each CNC axis. The offsets which can be selected via G functions are effective on an alternating basis.

2. External

   In addition to all the offsets which define the position of the workpiece zero point, an external zero offset can be overlaid by means of the handwheel (DRF offset) or from the PLC.

3. Programmable

   Zero offsets are programmable for all path and positioning axes with the TRANS command.

Working area

Three-dimensional zone into which the tool tip can be moved on account of the physical design of the machine tool. See -> protection zone.

Working area limitation

With the aid of the working area limitation, the traversing range of the axes can be further restricted in addition to the limit switches. One value pair per axis may be used to describe the protected working area.

Working memory

The working area is a RAM area in the -> CPU which is accessed by the processor to access the user program during program execution.

Workpiece

Part to be created/machined by the machine tool.

Workpiece contour

Setpoint contour of the -> workpiece to be created/machined.

Workpiece coordinate system

The starting position of the workpiece coordinate system is the -> workpiece zero. In machining operations programmed in the workpiece coordinate system, the dimensions and directions refer to this system.

Workpiece zero

The workpiece zero is the starting point for the -> workpiece coordinate system. It is defined in terms of the distance from the machine zero.

Index

To
Siemens AG

A&D MC BMS
Postfach 3180

91050 ERLANGEN, GERMANY

Phone: +49 (0) 180 / 5050-222 [Hotline]

Fax: +49 (0) 9131 / 98-2176 [Documentation]

E-mail: motioncontrol.docu@siemens.com

Suggestions

Corrections

For Publication/Manual:

SINUMERIK 840 sl
SINUMERIK 840D/840Di/810D
User Documentation

From

Name:

Company/Dept.

Street: ____

Zip code: ___________ City: _____________________

Phone: _________ / ____________________________

Fax: _______ / _________________________________

Programming Guide

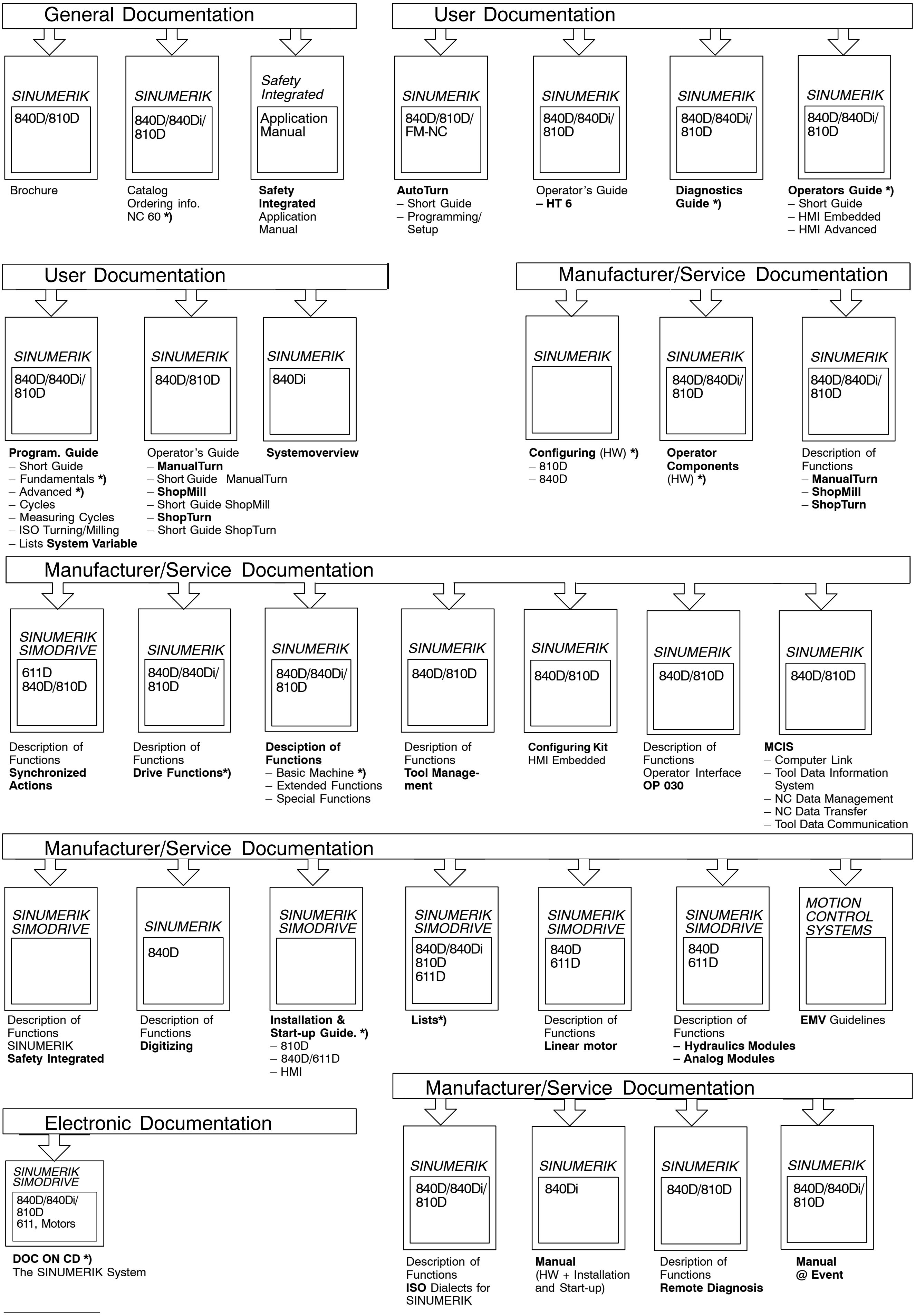
Order No.: 6FC5398-1BP10-0BA0

Edition: 08/05

Should you come across any printing errors when reading this publication, please notify us on this sheet. Suggestions for improvement are also welcome.

Suggestions and/or corrections

Documentation overview SINUMERIK 840D/840Di/810D (10.2004)



General Documentation

SINUMERIK 840D/810D
Brochure

SINUMERIK 840D/840Di/ 810D
Catalog
Ordering info.
NC 60 ***)**

Safety Integrated Application Manual
Safety Integrated
Application Manual

User Documentation

SINUMERIK 840D/810D/ FM-NC
AutoTurn
– Short Guide
– Programming/ Setup

SINUMERIK 840D/840Di/ 810D
Operator’s Guide
– HT 6

SINUMERIK 840D/840Di/ 810D
Diagnostics Guide *)

SINUMERIK 840D/840Di/ 810D
Operators Guide *)
– Short Guide
– HMI Embedded
– HMI Advanced

User Documentation

SINUMERIK 840D/840Di/ 810D
Program. Guide
– Short Guide
– Fundamentals ***)**
– Advanced ***)**
– Cycles
– Measuring Cycles
– ISO Turning/Milling
– Lists **System Variable**

SINUMERIK 840D/810D
Operator’s Guide
– **ManualTurn**
– Short Guide ManualTurn
– **ShopMill**
– Short Guide ShopMill
– **ShopTurn**
– Short Guide ShopTurn

SINUMERIK 840Di
Systemoverview

Manufacturer/Service Documentation

SINUMERIK
Configuring (HW) ***)**
– 810D
– 840D

SINUMERIK 840D/840Di/ 810D
Operator Components
(HW) ***)**

SINUMERIK 840D/840Di/ 810D
Description of Functions
– **ManualTurn**
– **ShopMill**
– **ShopTurn**

Manufacturer/Service Documentation

SINUMERIK SIMODRIVE 611D 840D/810D
Description of Functions
Synchronized Actions

SINUMERIK 840D/840Di/ 810D
Desription of Functions
Drive Functions*)

SINUMERIK 840D/840Di/ 810D
Desciption of Functions
– Basic Machine ***)**
– Extended Functions
– Special Functions

SINUMERIK 840D/810D
Desription of Functions
Tool Management

SINUMERIK 840D/810D
Configuring Kit
HMI Embedded

SINUMERIK 840D/810D
Description of Functions
Operator Interface
OP 030

SINUMERIK 840D/810D
MCIS
– Computer Link
– Tool Data Information System
– NC Data Management
– NC Data Transfer
– Tool Data Communication

Manufacturer/Service Documentation

SINUMERIK SIMODRIVE
Description of Functions
SINUMERIK
Safety Integrated

SINUMERIK 840D
Description of Functions
Digitizing

SINUMERIK SIMODRIVE
Installation & Start-up Guide. *)
– 810D
– 840D/611D
– HMI

SINUMERIK SIMODRIVE 840D/840Di 810D 611D
Lists*)

SINUMERIK SIMODRIVE 840D 611D
Description of Functions
Linear motor

SINUMERIK SIMODRIVE 840D 611D
Description of Functions
– Hydraulics Modules
– Analog Modules

MOTION CONTROL SYSTEMS
EMV Guidelines

Electronic Documentation

SINUMERIK SIMODRIVE 840D/840Di/ 810D 611, Motors
DOC ON CD *)
The SINUMERIK System

Manufacturer/Service Documentation

SINUMERIK 840D/840Di/ 810D
Description of Functions
ISO Dialects for SINUMERIK

SINUMERIK 840Di
Manual
(HW + Installation and Start-up)

SINUMERIK 840D/810D
Desription of Functions
Remote Diagnosis

SINUMERIK 840D/840Di/ 810D
Manual
@ Event

***)** These documents are a minimum requirement