SIEMENS

840D sl SINUMERIK Operate
5-Axis-Workshop
Technology Milling

840D sl SINUMERIK
Operate V4.7

Edition 2017.07
Training documentation

SIEMENS

SINUMERIK 840D sl

5-Axis-Workshop
Technology Milling

Valid for:

SINUMERIK 840D sl v4.7

Contents

A061

Sinumerik Operate

Menu Structure

Module Description:

The 840D sl SINUMERIK Operate system is split into 6 menu areas. Each menu area is sub-divided to allow the user easy access to the necessary areas of the system.

Menu Areas:

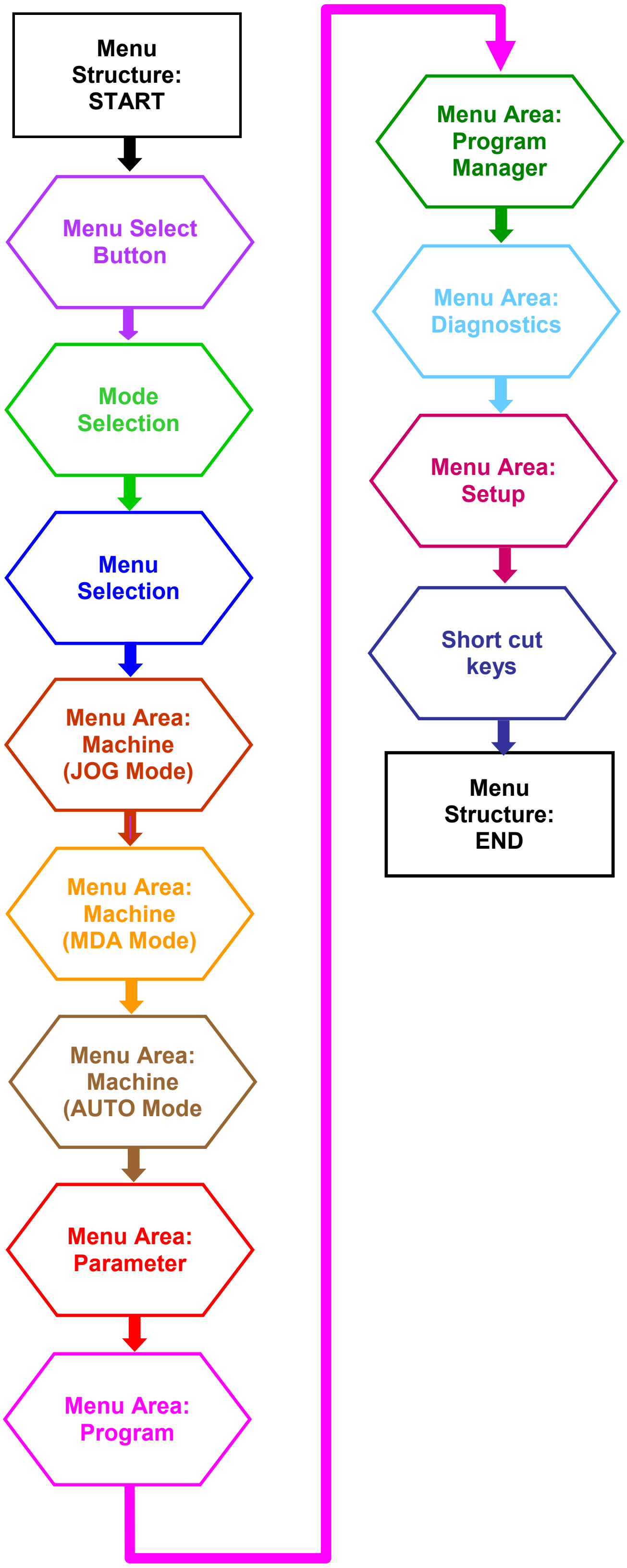
- Machine
- Parameter
- Program
- Program Manager
- Diagnostics
- Setup

Module Objective:

Upon completion of this module you will know the menu areas of the 840D sl SINUMERIK Operate system.

Content:

Sinumerik


Menu Structure: START
Menu Select Button
Mode Selection
Menu Selection
Menu Area: Machine (JOG Mode)
Menu Area: Machine (MDA Mode)
Menu Area: Machine (AUTO Mode
Menu Area: Parameter
Menu Area: Program
Menu Area: Program Manager
Menu Area: Diagnostics
Menu Area: Setup
Short cut keys
Menu Structure: END

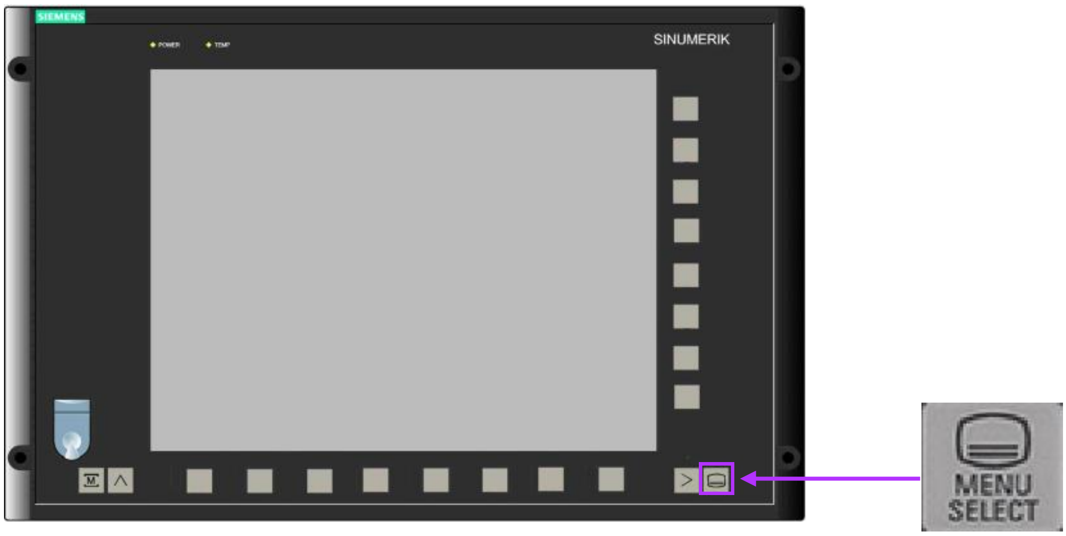
Notes

Menu Select Button

The menu system of the 840D sl SINUMERIK Operate is accessed via the “Menu Select” button.

The location of the “Menu Select” button is always to the lower right of the screen of the operator panel.

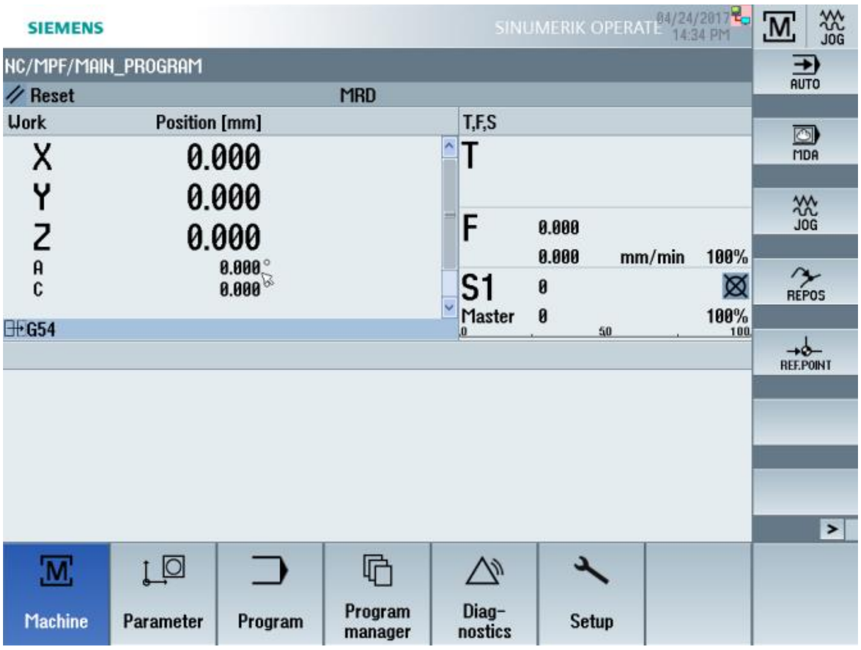
Examples:



Notes

Mode Selection

Pressing the Menu Select button reveals 6 horizontal menu soft-keys and 5 vertical soft-keys.



The 5 vertical soft-keys can be used to change between the machine modes: This can also be achieved via the Machine Control Panel.

Vertical Soft-key Functionality:

Automatic mode (AUTO) is selected.

Manual Data Automatic mode (MDA) is selected.

Jog mode (JOG) is selected.

Reposition mode (REPOS) is selected.

Reference Point (REF POINT) mode is selected.

Notes

Menu Selection

The 6 horizontal soft-keys are used to select the different menus.
It is also possible to select the menus via the integrated NC keyboard of certain types of OP or the remote NC keyboard versions.

Menu Soft-keys: Brief Description

The **Machine** menu displays information such as the axis positions, distances to go, feedrates, spindle speeds, active tool etc
Functionality such as manual tool selection, tool/work-piece measurement, hand-wheel selection can be performed by selecting the relevant soft-key.
The functionality depends upon the selected operating mode.

The **Parameter** menu displays information such tool/magazine data, work offsets, user variables etc.
It is possible to carry out functions such as creating tools, editing existing tool/ magazine data, editing work offsets/variables/setting data.

The **Program** menu is used to display a part program, only if a program is already open will its contents be displayed. Editing of the part program is possible here. If a part program is not already open, the Program Manager will be open.

The **Program Manager** menu displays the “Part program/subprogram/workpiece” directories. Programs/subprograms/workpieces can be created or opened for editing. It is also possible to save to/from controller using memory devices (CF/USB) or RS232 interface.

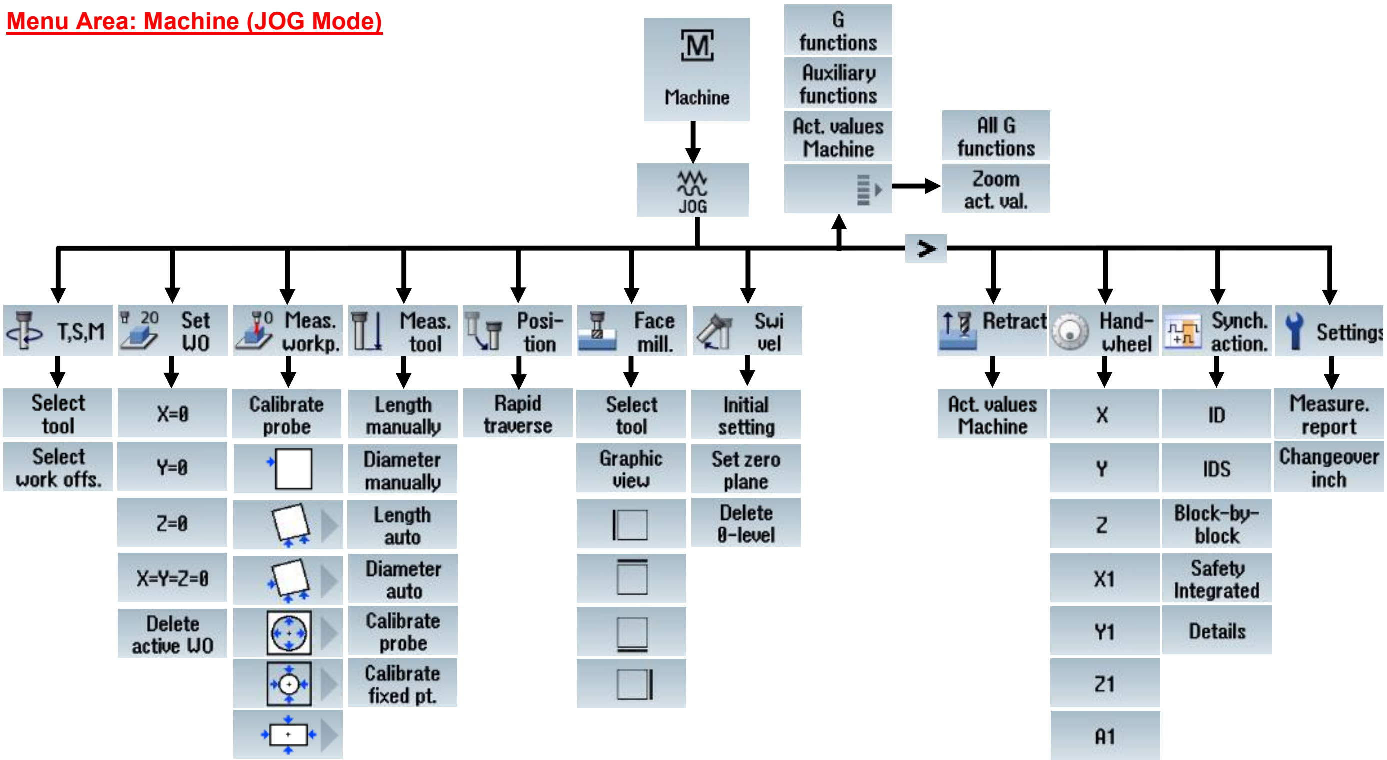
The **Diagnostics** menu is used for diagnosing machine problems, information about current/past alarms can be viewed. The status of drives, axes, networks, NC/PLC variables, utilization etc can be viewed. It is also possible to check the software/hardware versions of the system.
Machine information and service reports are also entered via the diagnostics menu.

The **Setup** menu is used for a wide variety of functions, these Include password entry, machine data modifications, creating archives, creating alarm files, time/date settings, network set-up, licensing/option selection, system data management, servo optimisation etc.

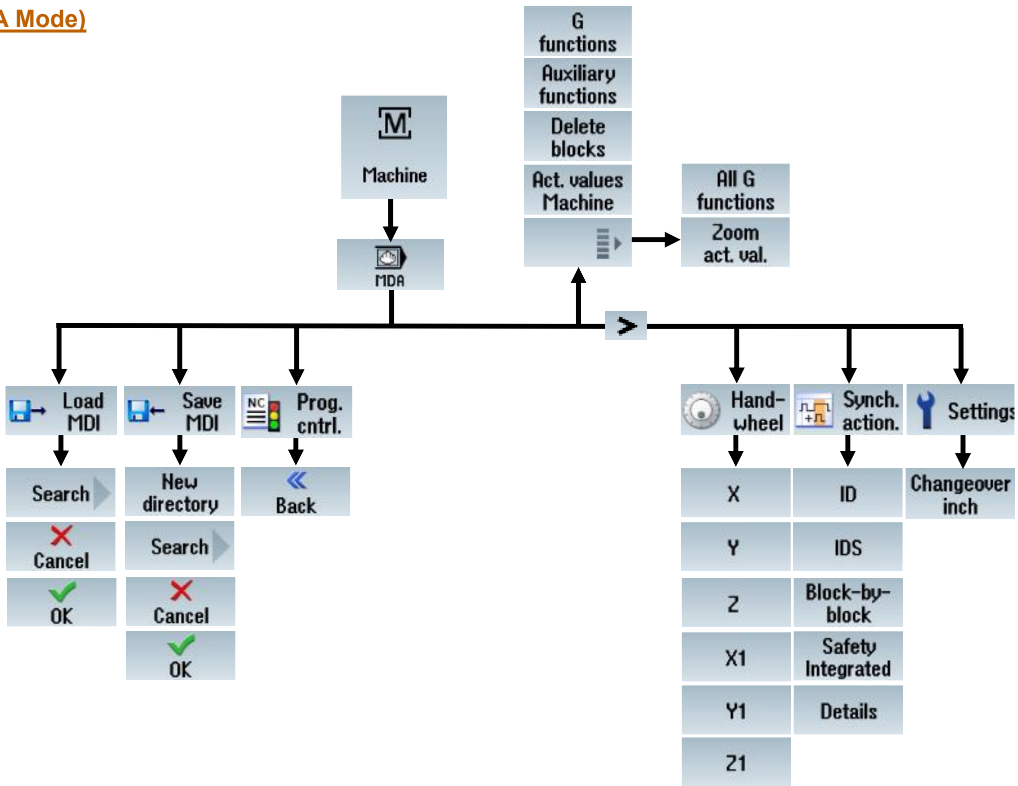
Reboot HMI
A further soft-key is available which is used for rebooting the HMI only. This is found via the Menu extension key.

Notes

Menu Area: Machine (JOG Mode)

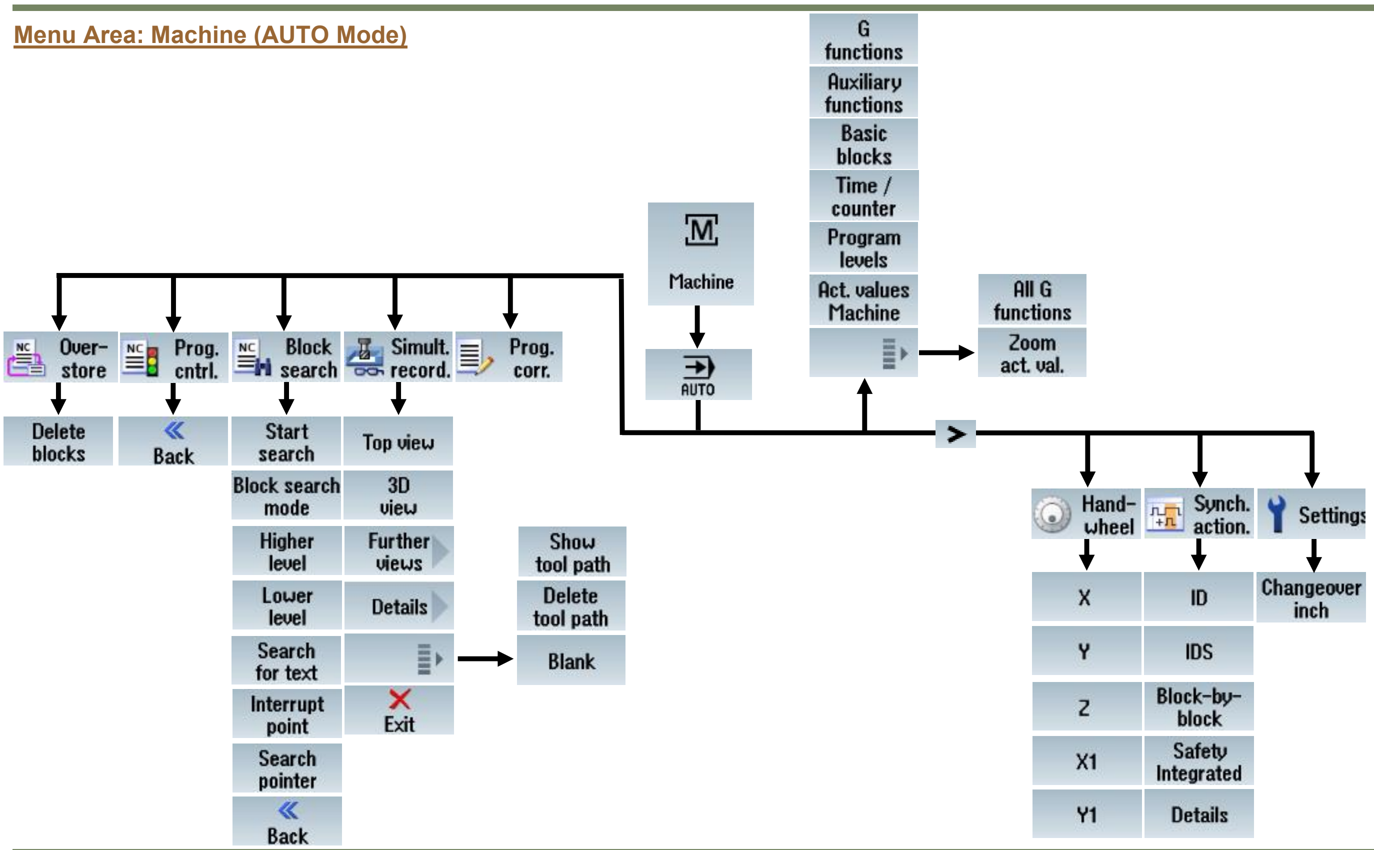


Menu Area: Machine (MDA Mode)

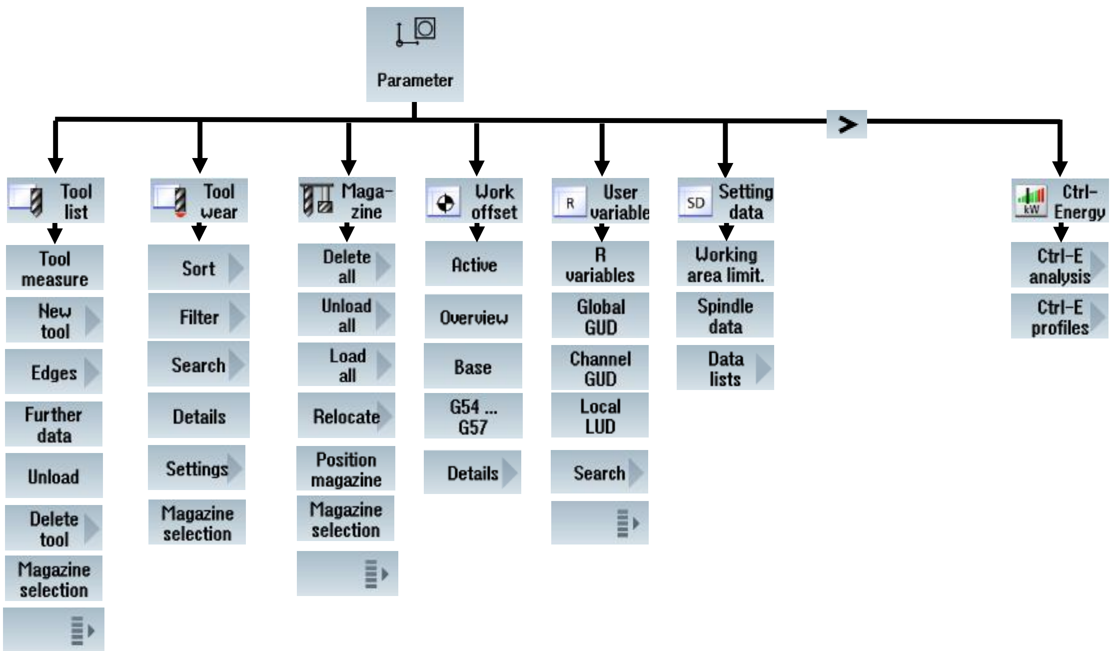


Notes

Menu Area: Machine (AUTO Mode)

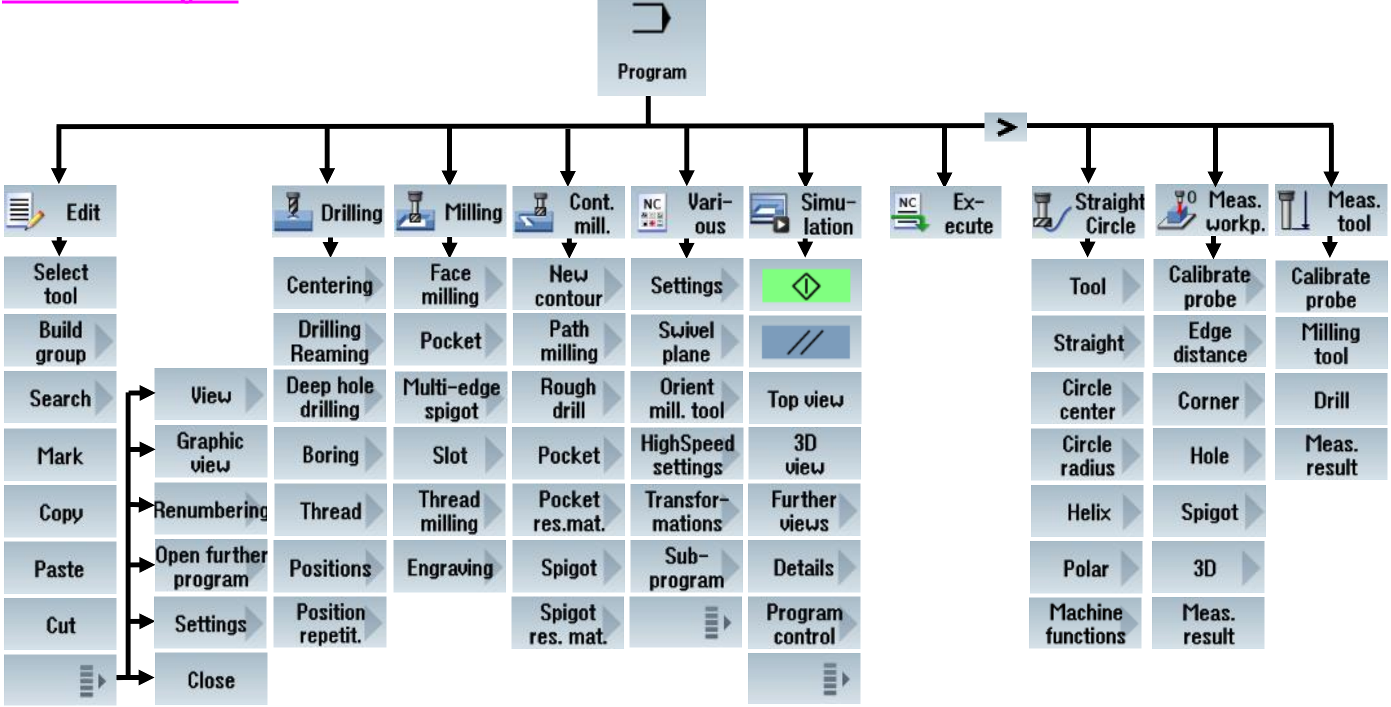


Menu Area: Parameter

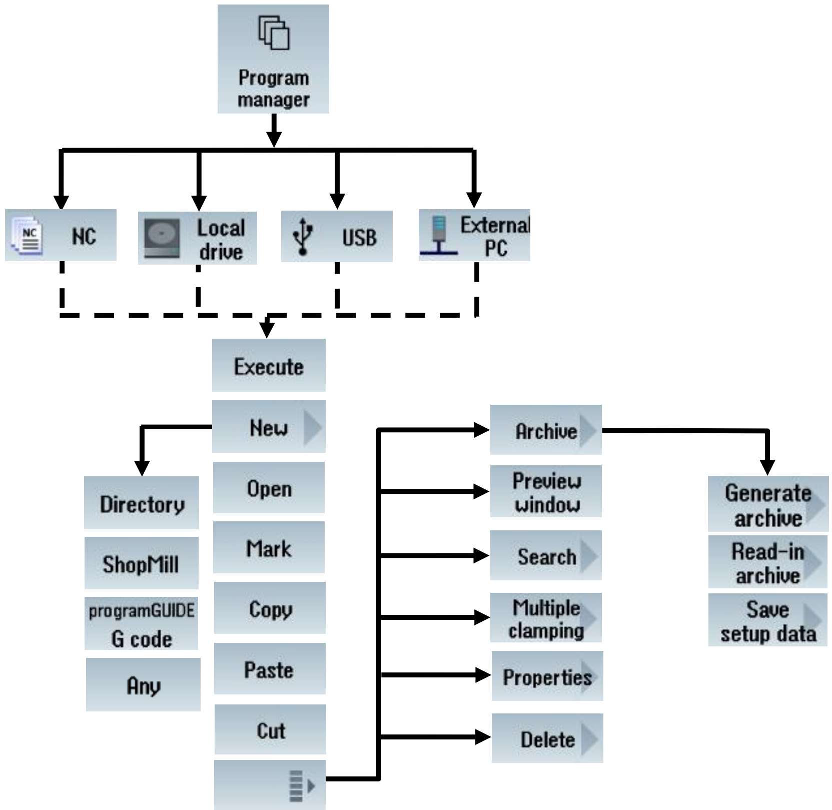


Notes

Menu Area: Program

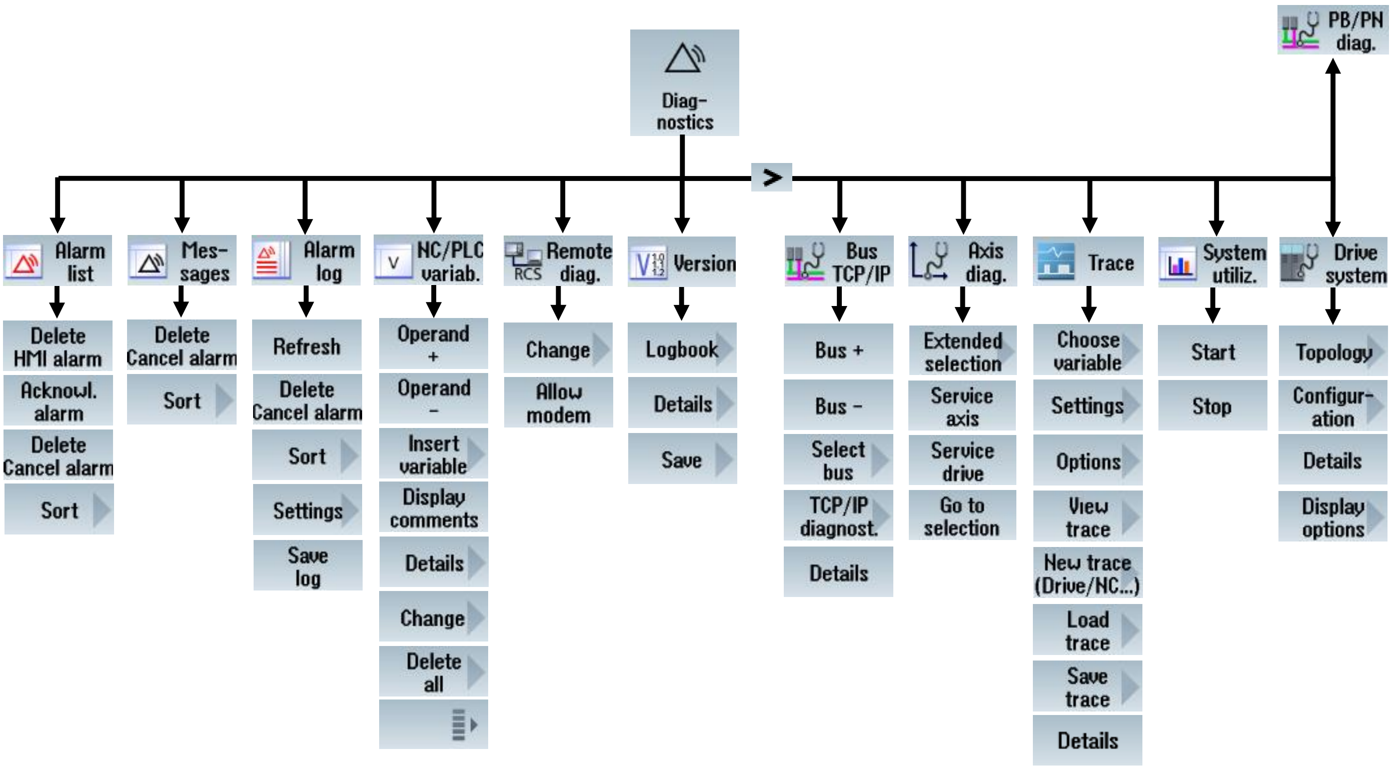


Menu Area: Program Manager

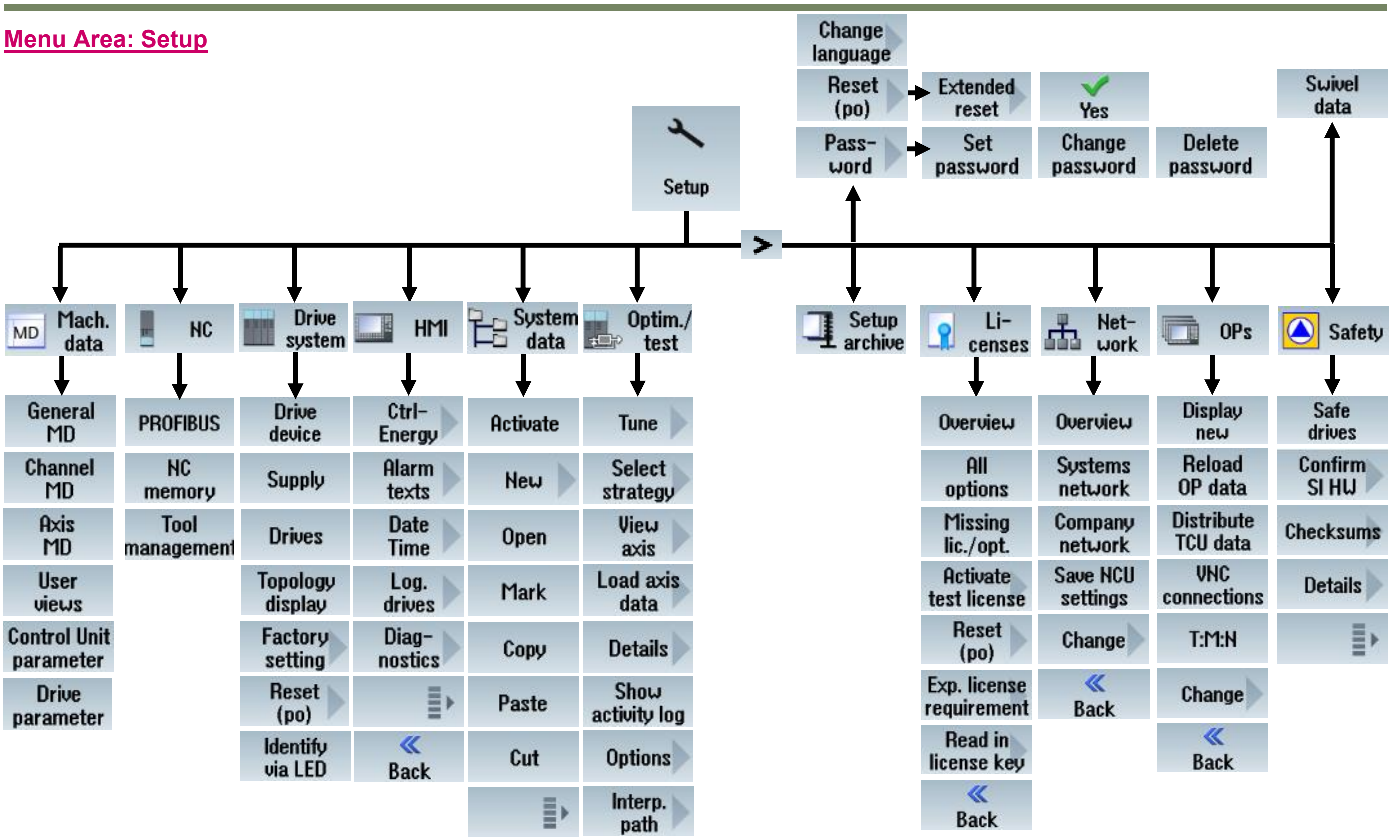


Notes

Menu Area: Diagnostics



Menu Area: Setup



Notes

Short cut Key sequences

Keys	Function
PAGE UP + SHIFT	In the program manager and in the program editor from the cursor position, selects directories or program blocks up to the beginning of the window.
PAGE UP + CTRL	Positions the cursor to the topmost line of a window.
PAGE DOWN + SHIFT	In the program manager and in the program editor from the cursor position, selects directories or program blocks up to the end of the window.
PAGE DOWN + CTRL	Positions the cursor to the lowest line of a window.
▶ + CTRL	Editing box - Moves the cursor further to the right by one word. Navigation - Moves the cursor in a table to the next cell to the right.

Keys	Function
◀ + CTRL	Editing box - Moves the cursor further to the left by one word. Navigation - Moves the cursor in a table to the next cell to the left.
▲ + CTRL	Editing box - Moves the cursor in the table to the beginning of the table. Navigation - Moves the cursor to the beginning of a window.
▲ + SHIFT	In the program manager and in the program editor, selects a contiguous selection of directories and program blocks.
▼ + CTRL	Editing box - Moves the cursor in the table to the end of the table. Navigation - Moves the cursor to the end of a window.
▼ + SHIFT	In the program manager and in the program editor, selects a contiguous, selection of directories and program blocks.

Notes

Keys	Function
END + SHIFT	Moves the cursor to the last entry.
END + CTRL	Moves the cursor to the last entry in the last line.
PAGE DOWN + SHIFT	In the program manager and in the program editor from the cursor position, selects directories or program blocks up to the end of the window.
BACKSPACE + CTRL	Deletes a word selected to the left of the cursor.
CTRL + A	In the actual window, selects all entries (only in the program editor and program manager).

Keys	Function
CTRL + C	Copies the selected content. The text is located in the clipboard.
CTRL + L	Scrolls the actual user interface through all installed languages one after the other.
CTRL + SHIFT + L	Scrolls the actual operator interface through all installed languages in the inverse sequence.
CTRL + P	Generates a screenshot from the actual operator interface and saves it as a file in the HMI data/Logs/Screenshots folder.
CTRL + X	Cuts out the selected text. The text is located in the clipboard.

Notes

Keys	Function
CTRL + Y	Reactivates changes that were undone (only in the program editor).
CTRL + V	Inserts text from the clipboard: either pastes the text from the clipboard at the actual cursor position or pastes text from the clipboard at the position of the selected text.
CTRL + ALT + S	Creates a complete standard archive (.ARC) on an external data carrier (USB-FlashDrive). **Note:** For diagnostics purposes only.
CTRL + ALT + D	Backs up the log files on the USB-FlashDrive. If a USB-FlashDrive is not inserted, then the files are backed-up in the manufacturer's area of the System CF-Card.

Keys	Function
CTRL + M	Feed over ride to maximum in Simulation
CTRL + ▲	Feed over ride is increased in Simulation
CTRL + ▼	Feed over ride to decreased in Simulation
ALT + S	Opens the Editor to enter Asian characters.
DEL + CTRL	Editing box - Deletes the first word to the of the cursor. Navigation - Deletes all characters

A061: END

Notes

A041

Sinumerik Operate

Machine and Setting Data

Module Description:

Machine and Setting data's are used in many ways to influence the operation of a machine.
During commissioning some of the machine data is automatically set while other machine data has to be manually set.
Over time, as the machine mechanically wears it may be necessary to edit certain machine data such as compensation data's.
This module shows the different machine data areas and how to activate the machine data.

Module Objective:

Upon completion of this module you will be able to locate the different Machine/Setting data areas and be able to carry out adjustments if required.
You will also be able to create a customised user list containing machine and setting data which are specific to a particular function/area of the machine.

Sinumerik

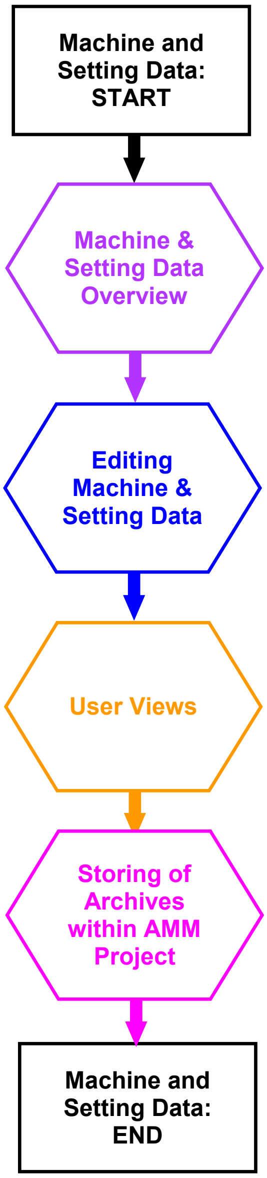
Content:

Machine & Setting Data Overview

Editing Machine & Setting Data

User Views

Storing of Archives within AMM Project


Machine and Setting Data: START
Machine & Setting Data Overview
Editing Machine & Setting Data
User Views
Storing of Archives within AMM Project
Machine and Setting Data: END

Notes

Machine & Setting Data Overview

General assignment

The Machine/Setting data are divided into the following ranges:

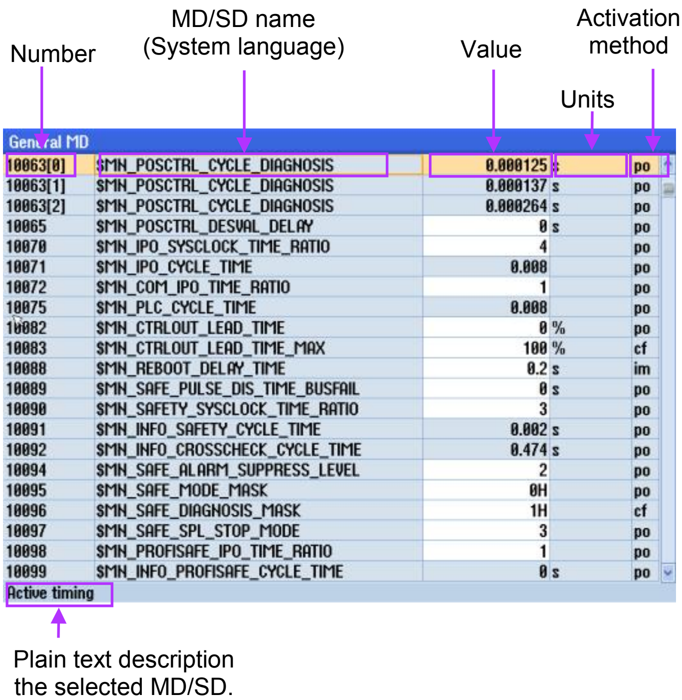
Range	Description
9000 - 9999	Display machine data
10000 - 19999	General machine data
20000 - 29999	Channel-specific machine data
30000 - 39999	Axis-specific machine data
41000 - 41999	General setting data
42000 - 42999	Channel-specific setting data
43000 - 43999	Axis-specific setting data
51000 - 51999	General cycle machine data
52000 - 52999	Channel-specific cycle machine data
53000 - 53999	Axis-specific cycle machine data
54000 - 54999	General cycle setting data
55000 - 55999	Channel-specific cycle setting data
56000 - 56999	Axis-specific cycle setting data

Each range contains the individual machine/setting data. Machine data is often referred to as **MD**, while setting data is referred to as **SD.**

Machine/Setting Data Location:

The following soft-key selection is used to locate the MD/SD area.

The data is listed in numerical order with the MD/SD number in the left column. The following columns show the description/name, value, units and activation requirement.



MD/SD identification:

The MD/SD name contains identifiers so that the system can identify that particular data.

1st character
$ = System variable (Present on all MD & SD)

2nd character (Defines between MD or SD)
M = Machine data
S = Setting data

3rd character (Defines type of MD/SD)
M = Display Machine Data
N = General Machine Data / Setting data
C = Channel Machine Data / Setting data
A = Axis Machine Data / Setting Data

4th character
S = Cycle Data (Defines MD/SD as cycle data)

Examples:
9900 $MM_MD_TEXT_SWITCH Display MD
54600 $SNS_MEA_WP_BALL_DIAM General Cycle SD
55623 $SCS_MEA_EMPERIC_VALUE Channel Cycle SD
30200 $MA_NUM_ENCS Axis MD

Notes

The following data ranges are used for the machine data:

$MM_	Display machine data
$MN_	General machine data
$MC_	Channel-specific machine data
$MA_	Axis-specific machine data
$MNS_	General cycle machine data
$MCS_	Channel-specific cycle machine data
$MAS_	Axis-specific cycle machine data

The following data ranges are used for the setting data:

$SN_	General setting data
$SC_	Channel-specific setting data
$SA_	Axis-specific setting data
$SNS_	General cycle setting data
$SCS_	Channel-specific cycle setting data
$SAS_	Axis-specific cycle setting data

Indexes:

Certain MD/SD have indexes. The function and number of indexes depends on the data in question.

Example:
General MD **10000 $MN_AXCONF_MACHAX_NAME_TAB**

This MD is used for defining the names of the axes in the system. Each axis name is assigned to an index the MD.

General MD

10000[0]	$MN_AXCONF_MACHAX_NAME_TAB	X1
10000[1]	$MN_AXCONF_MACHAX_NAME_TAB	Y1
10000[2]	$MN_AXCONF_MACHAX_NAME_TAB	Z1
10000[3]	$MN_AXCONF_MACHAX_NAME_TAB	SP1
10000[4]	$MN_AXCONF_MACHAX_NAME_TAB	A1
10000[5]	$MN_AXCONF_MACHAX_NAME_TAB	C1
10000[6]	$MN_AXCONF_MACHAX_NAME_TAB	U1
10000[7]	$MN_AXCONF_MACHAX_NAME_TAB	V1

Further examples:
30240[0] $MA_ENC_TYPE = The type of encoder connected to the **1st** measuring system input.
30240[1] $MA_ENC_TYPE = The type of encoder connected to the **2nd** measuring system input.

Help Function:

The function of each MD/SD is described in detail via the “Help” key.
The MD/SD first needs to be selected and then the “Help” key pressed. The key is found on the NC keyboard

A detailed description will be displayed. This will include units, limits etc as well as the functionality.

10000	AXCONF_MACHAX_NAME_TAB				N01, N11	K2, F1, G2, F2, K5, M1
-	PowerOn					

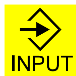
List of the machine axis identifiers.
The name of the machine axis is entered in this MD.
In addition to the fixed, defined machine axis identifiers "AX1", "AX2" ..., user-defined identifiers for the machine axes can also be assigned in this data.
The identifiers defined here can be used parallel to the fixed, defined identifiers for addressing axial data (e.g. MD) and machine axis-related NC functions (reference point approach, axial measurement, travel to fixed stop).

Special cases:

Exit Help

Activation of the MD/SD:

There are four methods available for activating the MD/SD. The method depends on the MD/SD in question.
The right hand column of the screen displays the required method.

po	Active via “Reset (po)” soft-key or controller power off/on.	Reset (po)
cf	Active on selection of “Set MD active (cf)” soft key.	Set MD active (cf)
re	Active on selection of the appropriate PLC interface signal, usually via MCP “Reset” key.	 RESET
im	Active immediately the “Input” key is selected to confirm the value.	 INPUT

Notes

Units for MD/SD

The function of the MD/SD determines the units in use.
The following table shows the abbreviations and descriptions of the units.

Unit	Description
mm/sec	Millimetres/second: Linear velocity/speed
rpm	Revolutions/minute: rotational velocity/speed
m/s²	Meters/second-squared: Linear acceleration
rev/s²	revolutions/second-squared: rotary acceleration
kgm²	Kilogram/meter-squared: Moment of inertia
s	seconds: measurement of time.
Hz	Cycles/second: Frequency.
deg	Degrees: Angular measurement.

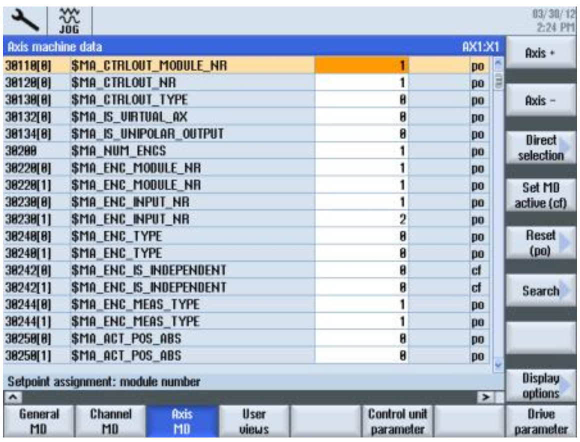
Unit	Description
mm/rev	millimetres/revolution: linear feed-rate determined by a rotary axis.
m/s³	Meters/second-cubed: linear jerk
rev/s³	revolutions/second-cubed: rotary jerk
Kg	Kilogram: Measurement of mass
mm	millimetre: Linear measurement.
Nm	Newton-metre
mH	milli-Henry: Measurement of inductivity
A	Amperes (Amps): Electrical current
V	Volts: Electrical potential

Some MD/SD have no units associated with them. In this case use the "Help" function to determine the data type required.

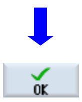
Editing Machine & Setting Data

To locate the machine data screens first ensure that the relevant access level is active (Module A021 Protection Levels) then use the following key sequence:-

The default is for the General Machine screen to be displayed



The machine data which are available for viewing can be restricted via the "Display options" soft-key



Notes

The Machine & Setting Data is separated into different groups.

General MD and general Cycle MD.

Channel-specific MD and channel-specific Cycle MD.

Axis-specific MD and axis-specific Cycle MD.

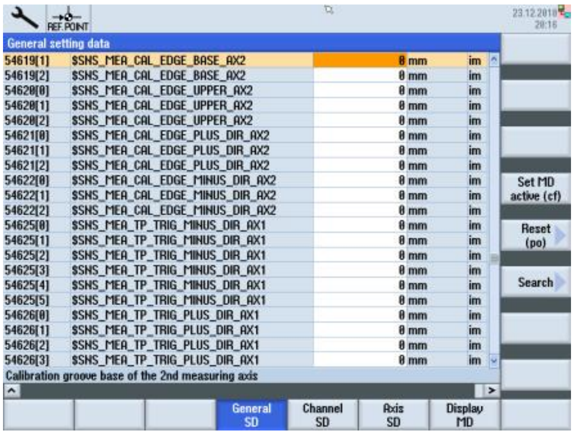
User Views- Allows the user to create list of specific machine data.

Control Unit MD.

Supply MD (ALM/SLM-only DriveCLiQ types).

Drive MD (Motor Modules)

For further data the “Menu Extension” key must be selected. This allows access to the Setting Data and also the Display Machine Data.



General SD and general Cycle SD.

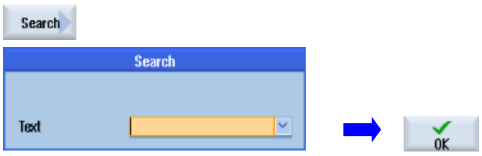
Channel-specific SD and channel-specific Cycle SD.

Axis-specific SD and axis-specific Cycle SD.

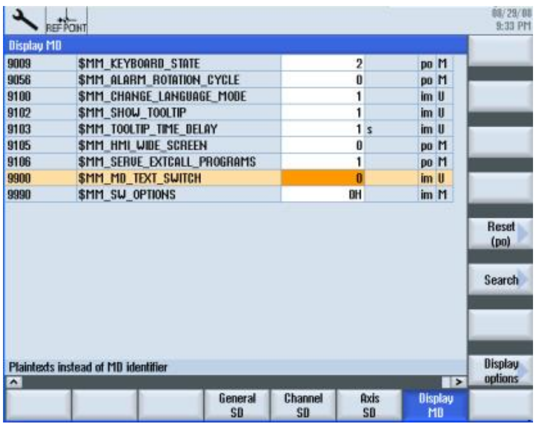
Display MD.

The correct machine data should first be located. This is done by either using the page/cursor keys or the search function.

The search function is activated with the “Search” soft key. If the Machine Data number is known this can be typed into the dialogue box. It is also possible to enter part or all of the MD name in text format.

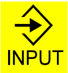


When the correct MD has been found, a value can be entered. In this example we are going to change Display MD 9900 from 0 to 1. This will change the MD/SD description from system language to plain text.

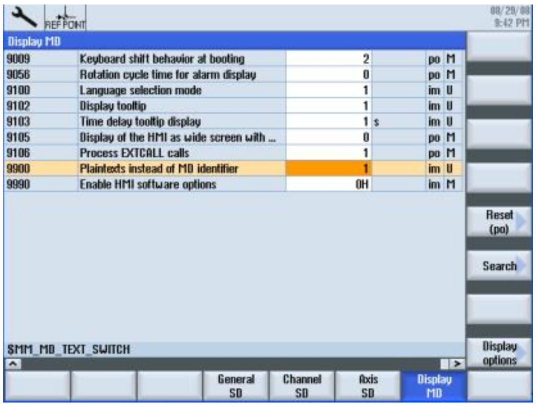


Notes

The correct machine data has to be highlighted. The number 1 is entered and confirmed with the “Input” key.



On pressing the “Input” key the text of the machine data changes immediately (see below). This will apply to all MD/SD.



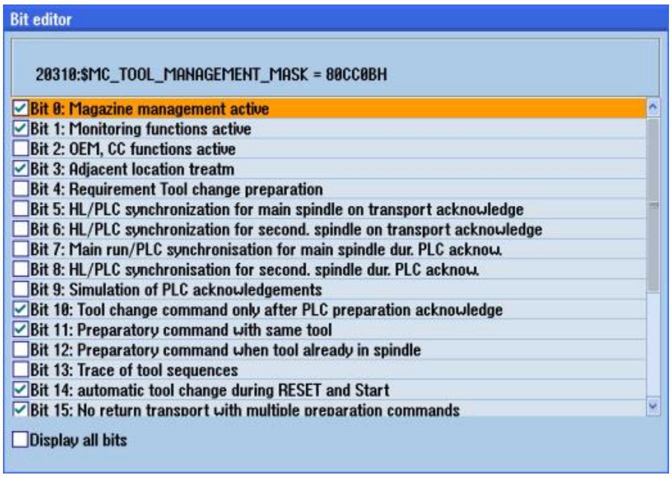
Some MD/SD have no units associated with them. In this case use the “Help” function to determine the data type required.

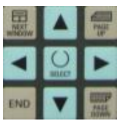
MD/SD with “Bit patterns”

If the letter “H” appear as the last digit in the “Value” column the MD/SD requires a value in Hexadecimal format. A bit pattern can be viewed.
To view the bit pattern, highlight the MD/SD and then use the “Select” key found on the NC keyboard.

Example: Channel MD 20310

A descriptive list of each individual bit is displayed. Use the cursor keys to highlight the required bit, use the select key to activate/deactivate the bit.



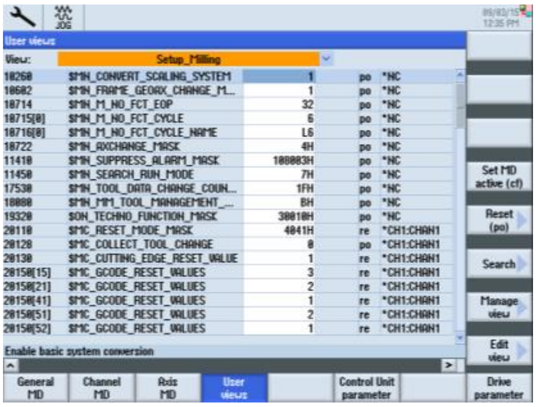


REMEMBER:- Create a data backup.
Any changed MD/SD should be saved in an archive.
See Module A017.

Notes

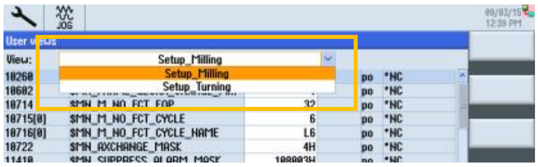
User Views

The “User views” function allows a customised list of machine data to be made. MD/SD can be listed that are of importance to a specific task/function of the machine.



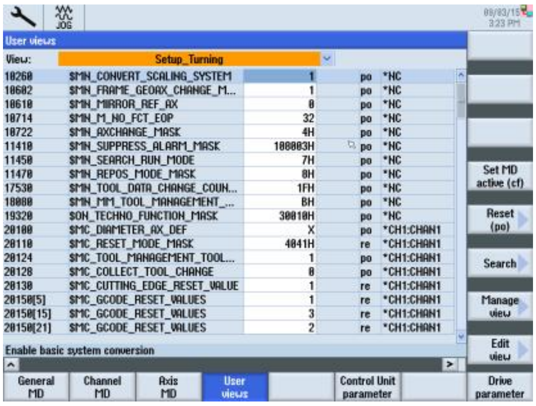
By default on the control there are two “User view” lists. “Setup_Milling” and “Setup_Turning”

These views list all MD/SD required for setting the technology Milling or Turning.



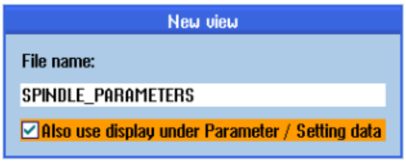
Module A110 Setup Milling and A120 Setup Turning describe the minimum setting for the respective MD and SD that should be applied in each case.
When a bit pattern is described these bits MUST be set, in addition to any that have been set by the OEM.

The “SELECT” button can be to show the Setup_Turning view.



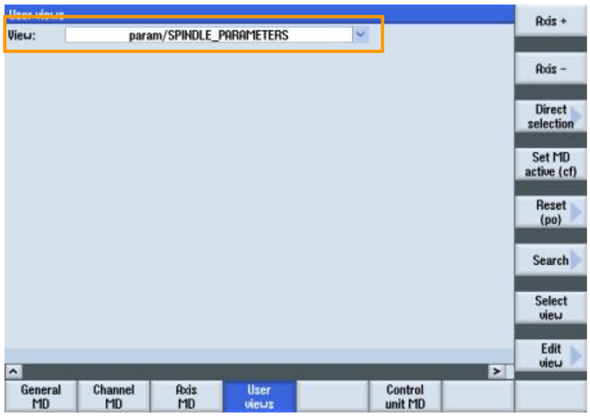
Edit existing machine data list or create new machine data list.

Enter a name for the list e.g. SPINDLE_PARAMETERS. The “User views” soft-key will also be displayed in the “Parameter/Setting data” area if the option is selected.

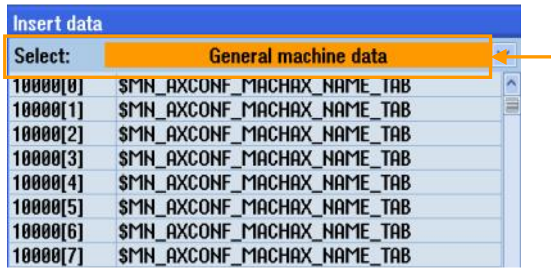


Notes

The new entry will be displayed in the View menu.
It is now possible to add the required MD/SD.

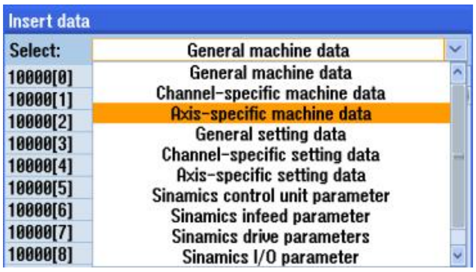


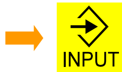
The General machine data will be displayed by default.
A list of all types of data can be viewed on the “Select” list



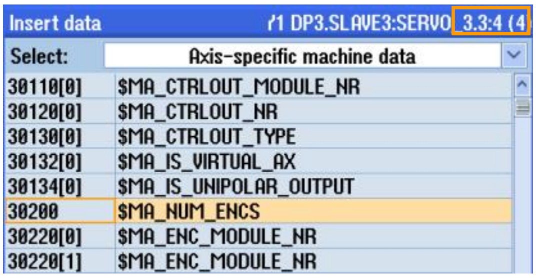
In the following example Axis MD for the spindle will be added.

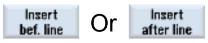
The required MD/SD type is selected from the list and confirmed with the “Input” key.



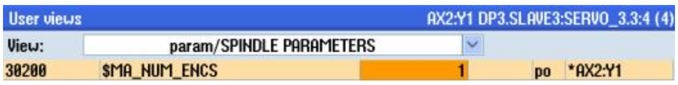


Select the data to be added to the list.



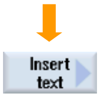


The selected MD will appear on the screen.



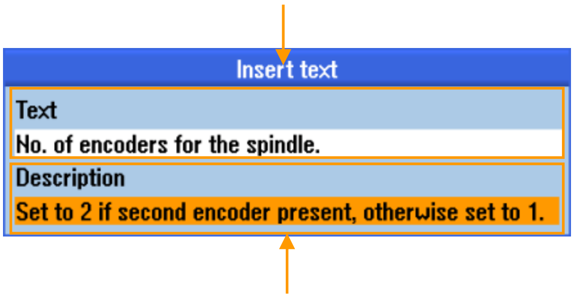
Repeat the procedure to add further data to the list.

Text for information purposes can be added:



The text entered in the “Text” area will be displayed on the list.
The text entered in the “Description” area is displayed at the bottom of the screen.

Text to be displayed on the list.
(Maximum 30 displayable characters)

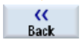


Detailed text to be displayed at the bottom of the screen. (Maximum 80 characters)

Notes

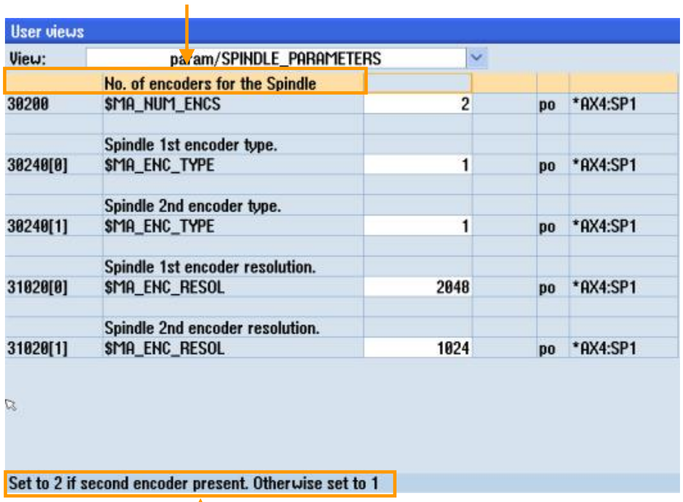
The position of the text line can be selected when editing is complete.

Or

Example of a list of spindle specific MD complete with descriptive texts:

Short text description.

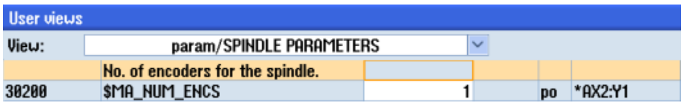


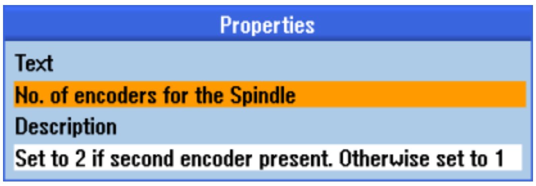
Detailed text description.

Modifying a MD/SD entry:

The entries on the list can be modified by selecting the “Properties” soft-key.

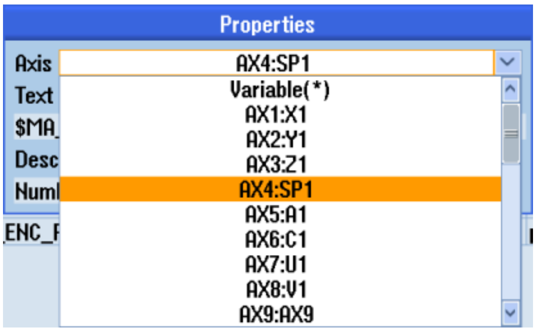
Select the entry to be modified and press the “Properties” soft-key.





To change the order that the items appear on the list, first select the item to be moved and then move it up/down the list via the “Up” or “Down” soft-keys.

Each press will move the item one position.



The changes that can be made will depend on the MD/SD in question.
In this example, a different axis for the MD has been selected.

Or

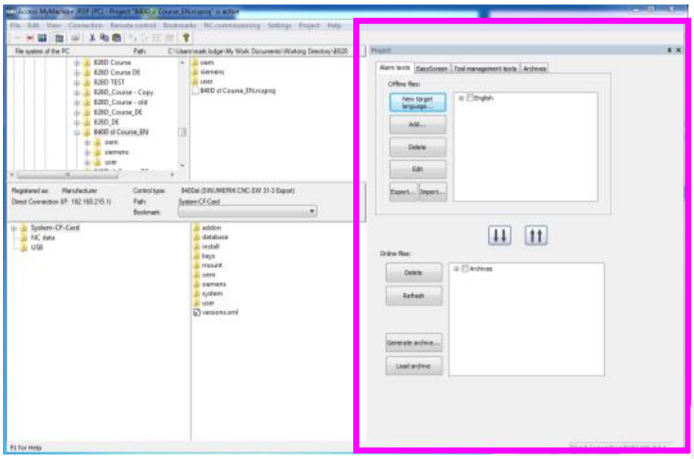

Any unwanted items are removed via the “Delete” Soft-key.

Notes

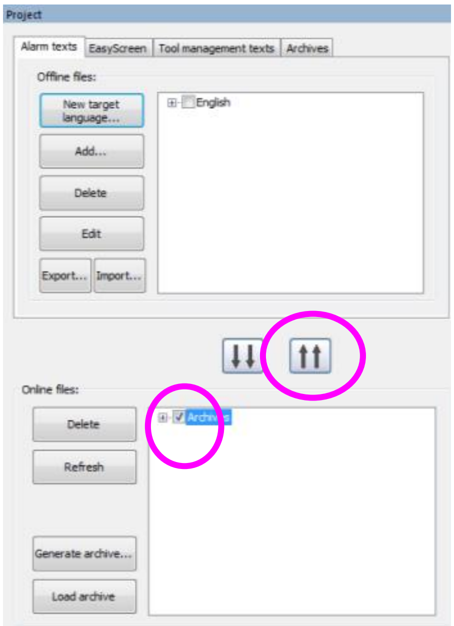
Storing of Archives within an AMM Project

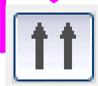
Start the Access MyMachine program and make an online connection to the controller.

Make sure the project, created previously in module A002, appears on the right-hand side.

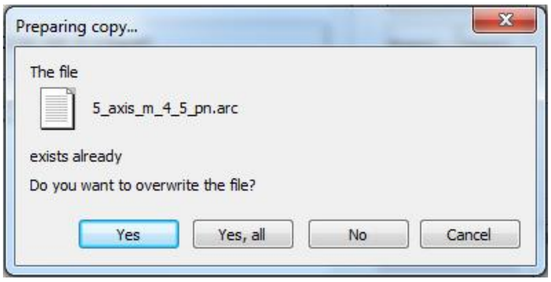


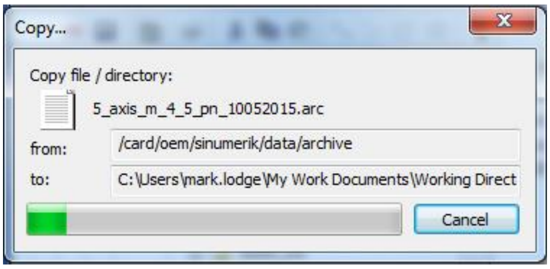
To upload the Archive files from controller into the project, tick all boxes in the bottom window.



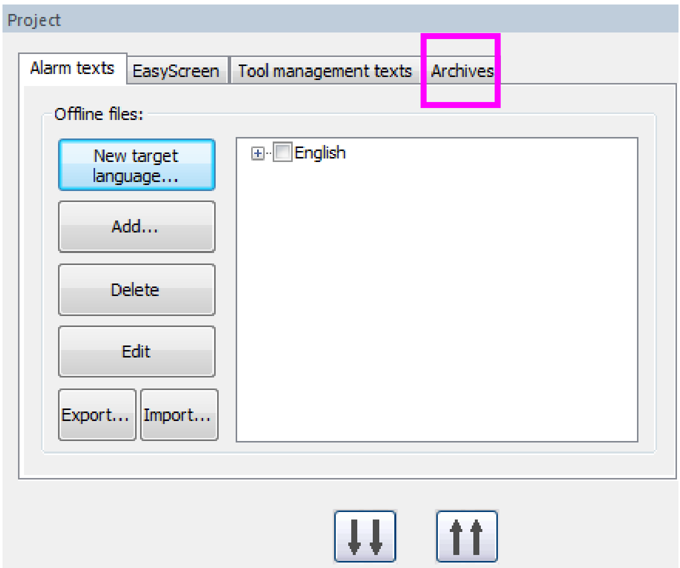


Followed by the "Upload" icon.



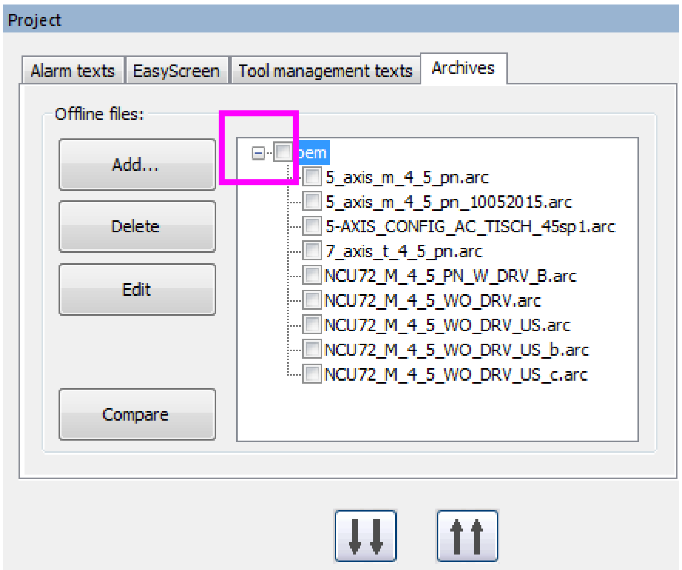


The uploaded files can be seen in the upper window shown under the "Archives" pull down menu.

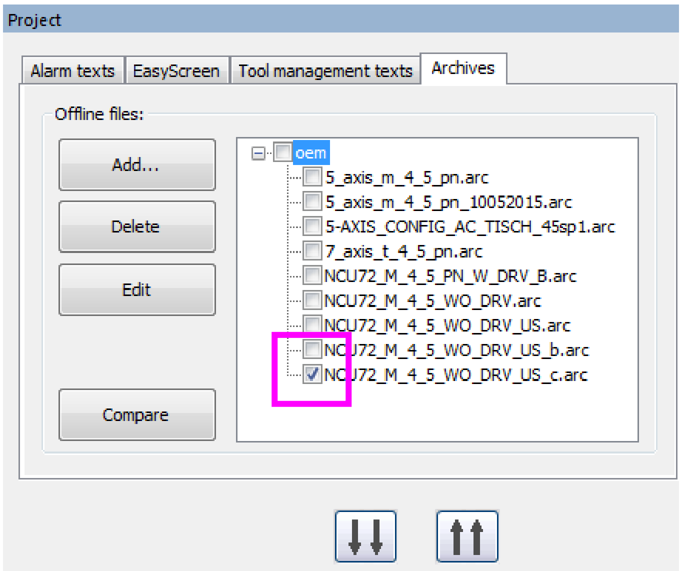


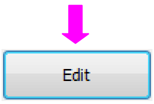
Notes

Use the "+" icon to the right of the OEM text, the contents of the Archives in the "project" can be seen.

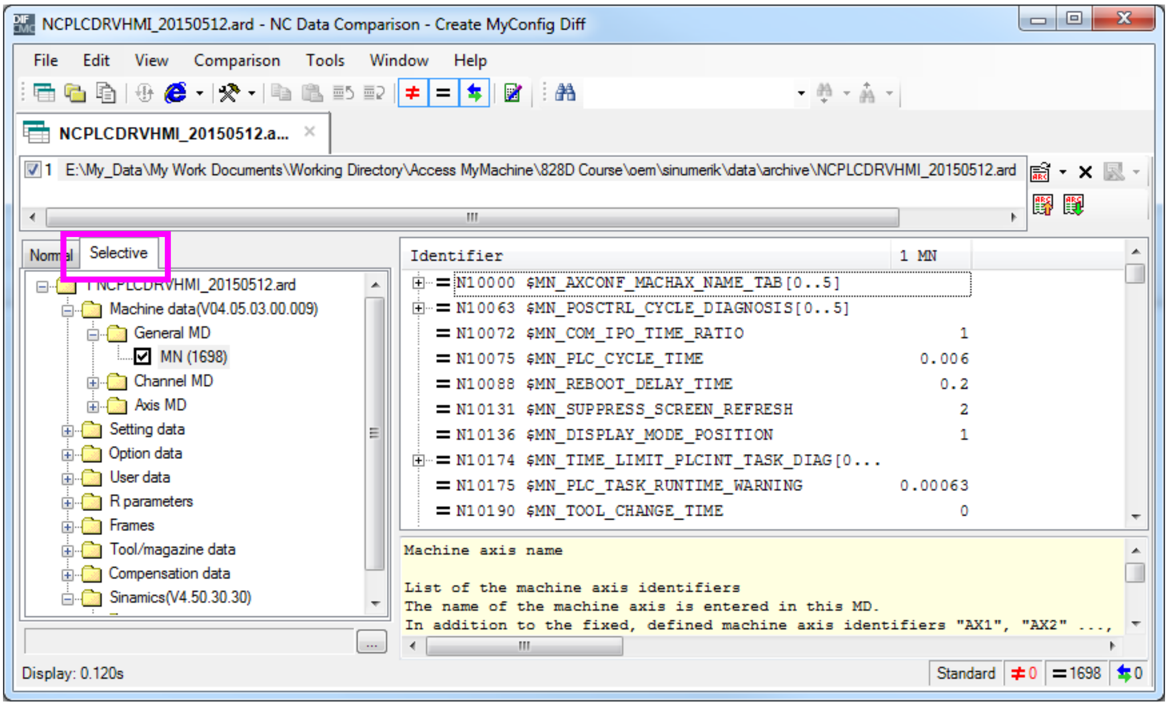


There are several tasks that can be performed from within AMM software. An archive can be opened and the data looked at.
Tick one of the Archive file and click the "Edit"



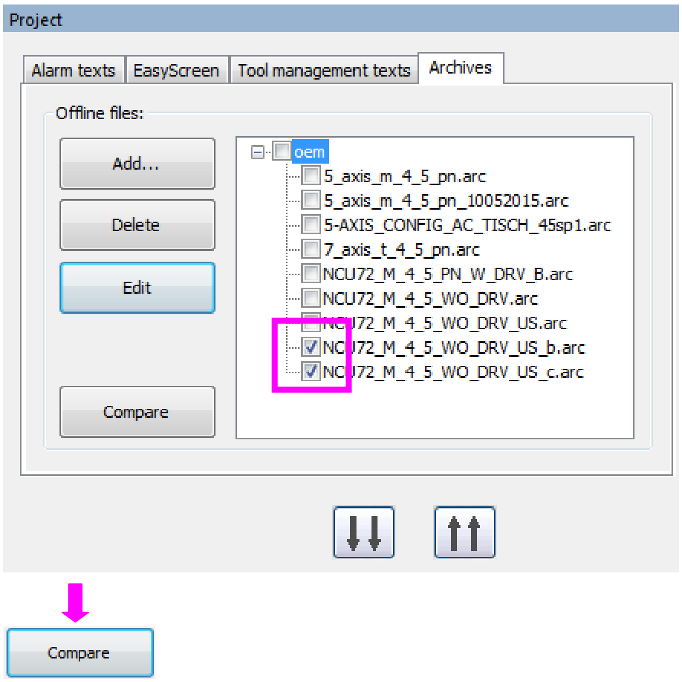


This will open the file within "Create MyConfig Diff".
Select the menu , "Selective"

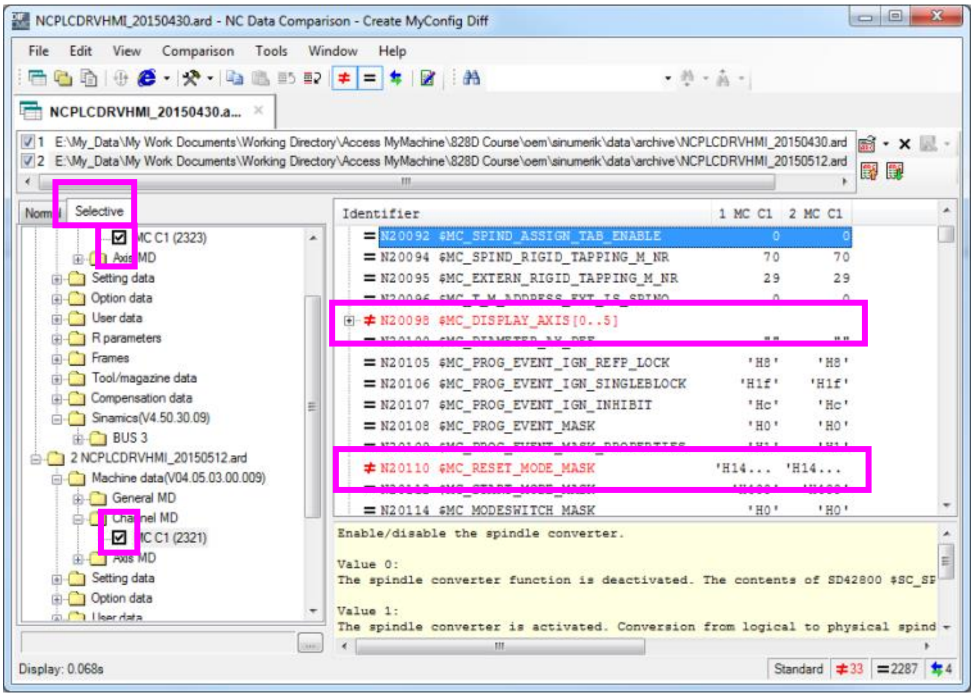


Notes

Two archives can be opened and the data compared with each other.
Tick two Archive files and click the "Compare"



This will open the two files within "Create MyConfig Diff".
Select the menu , "Selective", then open and tick the same areas within each file.
Data that is different will be highlighted in RED text.



A041: END

Notes

Coordinate systems and frames

Module Description:

This Module describes the various types of coordinate systems in the frame chain and their correlation. Important frame types are explained in detail with their system frame variables. Through chaining of individual frames you get to learn how to create a swivel frame with tool carrier, translations and rotations.

Module Objective:

After completion of this module you will understand the correlation between frames and swivel cycle “CYCLE800”. You get to learn how to create a swivel frame conventionally with various language commands like **TOROT**, **PAROT** and **TCARR**, to be able to machine oblique surfaces with any orientation in space on a 3+2 axis milling machine. In addition you get to learn how to store a rotation or translation directly into a system frame.

Content:

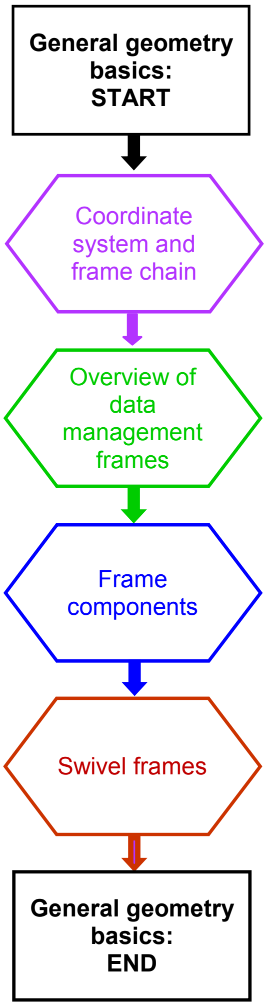
Coordinate systems and frame chain

Overview of the data management frames

Programmable frame components

Swivel frames

840D sl SINUMERIK Operate


General geometry basics: START
Coordinate system and frame chain
Overview of data management frames
Frame components
Swivel frames
General geometry basics: END

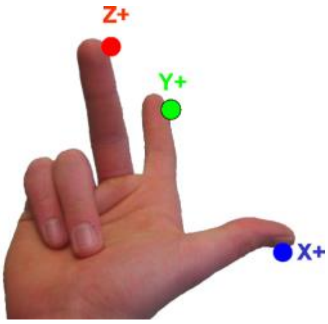
Notes

Coordinate systems and frame chain

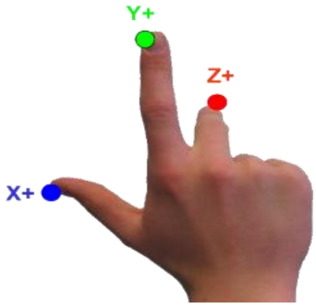
Cartesian coordinate systems

General explanation

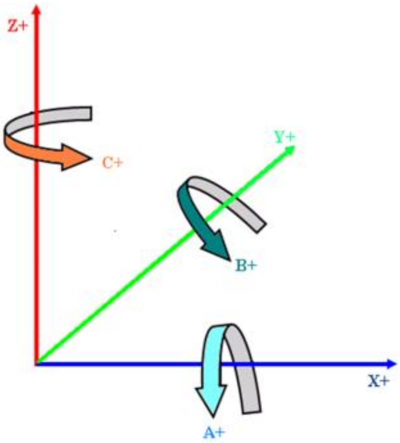
DIN 66217 stipulates that machine tools must use right-angled, rectangular (Cartesian) coordinate systems. The positive directions of the coordinate axes are determined using the "Right Hand Rule". The coordinate system is related to the workpiece and programming takes place independently of whether the tool or the workpiece is being traversed. When **programming**, it is **always** assumed that the **tool** traverses **relative to** the **coordinate system of the workpiece (WCS)**, which is intended to be stationary.



Picture1.1: Right hand rule
Vertical milling machine



Picture1.2: Right hand rule
Horizontal milling machine



X, Y, Z
Axes perpendicular to one another

A, B, C
Rotary axes rotating around X, Y, Z

When looking into positive direction of a linear axis a rotation around this axis in clockwise direction is equal to the positive rotation sense of the particular rotary axis.

Picture 1.3: Clockwise, rectangular Cartesian coordinate system

The following coordinate systems are defined:

MCS Machine Coordinate System

BCS Basic Coordinate System

BZS Basic Zero System

SZS Settable Zero System

WCS Workpiece Coordinate System

Notes

Interrelationships between coordinate systems

General explanation

The coordinate systems are determined by the kinematic transformation and the FRAMES.

A kinematic transformation is used to derive the BCS from the MCS. If no kinematic transformation is active, the BCS is the same as the MCS.

The basic frame maps the BZS onto the BCS.

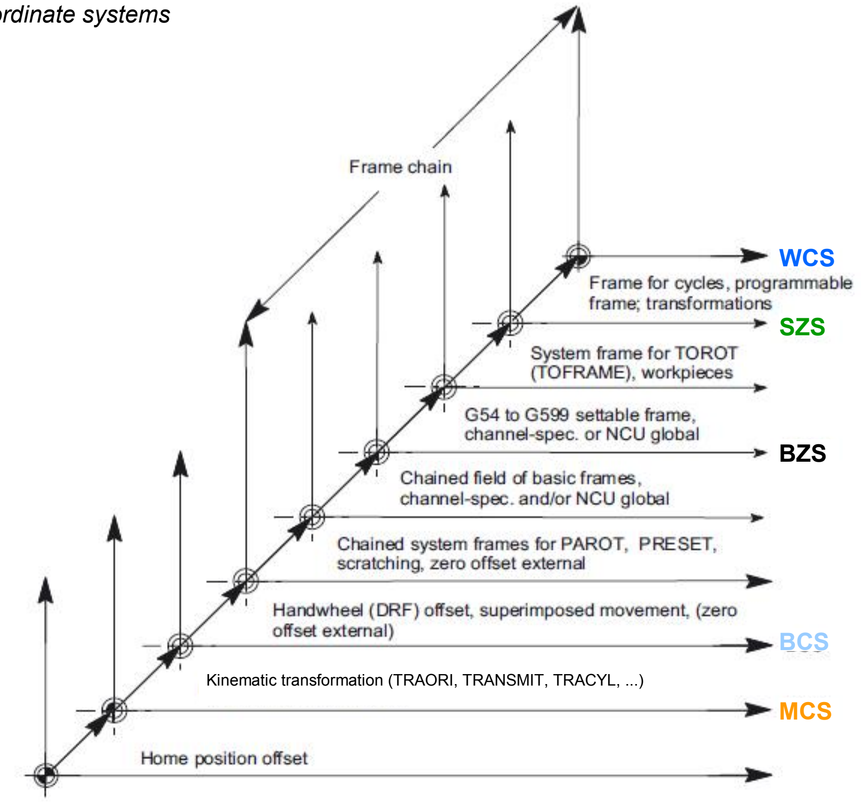
An activated adjustable FRAME G54...G599 SZS is derived from the BZS.

The WCS, which is the basis for programming, is defined by the programmable frame.

WCS: Workpiece Coordinate System
SZS: Settable Zero System
BZS: Basic Zero System
BCS: Basic Coordinate System
MCS: Machine Coordinate System

Picture 1.4: Interrelationship between coordinate systems

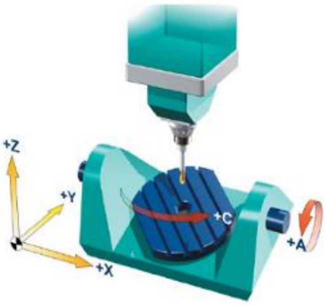
WCS: Workpiece Coordinate System
SZS: Settable Zero System
BZS: Basic Zero System
BCS: Basic Coordinate System
MCS: Machine Coordinate System



Notes

Machine Coordinate System (MCS)

The machine coordinate system (MCS) is made up of all physically available machine axes.



Picture 1.5: MCS with machine axes X, Y, Z, A, C (5-axis machine kinematic type P)

Kinematic transformation

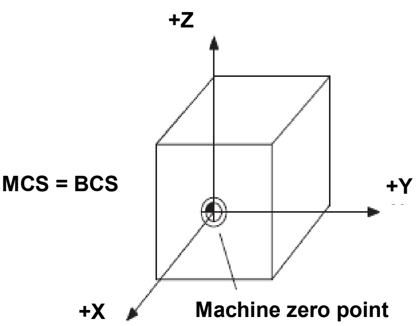
The workpiece is always programmed in a two- or three-dimensional, right-angled coordinate system (WCS). However, such workpieces are being programmed ever more frequently on machine tools with rotary axes or linear axes not perpendicular to one another. Kinematic transformation is used to represent coordinates programmed in the workpiece coordinate system (rectangular) in real machine movements.

The kinematic transformation is specified in the tool carrier parameters TCARR as kinematic chain in the form of a data record (see Section 4 in this module; additional information on this topic can be found in module M102 CYCLE800).

Basic coordinate system (BCS)

The basic coordinate system (BCS) consists of three mutually perpendicular axes (geometry axes) as well as other special axes, which are not interrelated geometrically.

Machine tools without kinematic transformation:
BCS and MCS always coincide when the BCS can be mapped onto the MCS without kinematic transformation (up to three axes are mapped onto the MCS). On such machines, machine axes and geometry axes can have the same names.

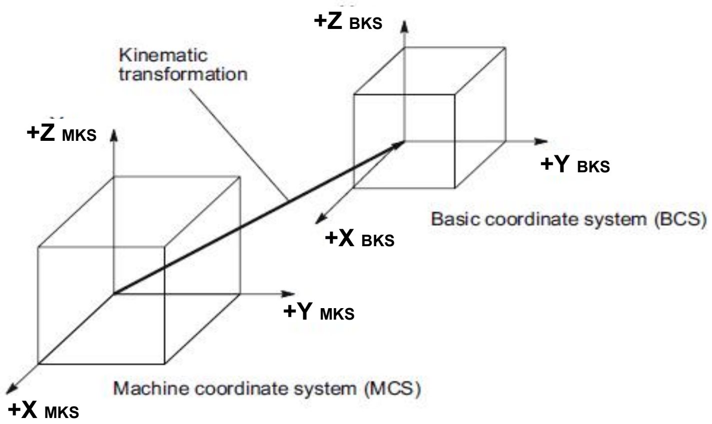


Picture 1.6: MCS=BCS without kinematic transformation

Notes

Machine tools with kinematic transformation:

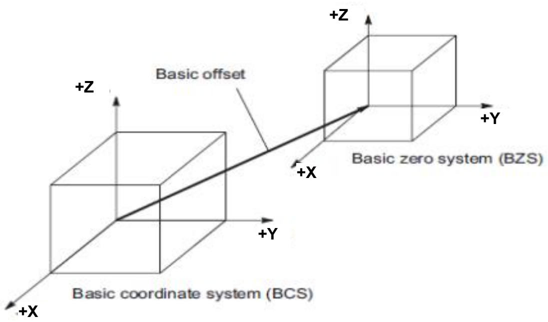
The BCS and MCS do not coincide when the BCS is mapped onto the MCS with kinematic transformation (e.g., TRANSMIT / face transformation, TRAORI / 5-axis transformation or more than three axes can be mapped onto the MCS). On such machines the machine axes and geometry axes must have different names.



Picture 1.7: Kinematic transformation between the MCS and BCS

Basic offset

The basic offset describes the coordinate transformation between BCS and BZS. It can be used, for example, to define the palette window zero (see picture 1.10).

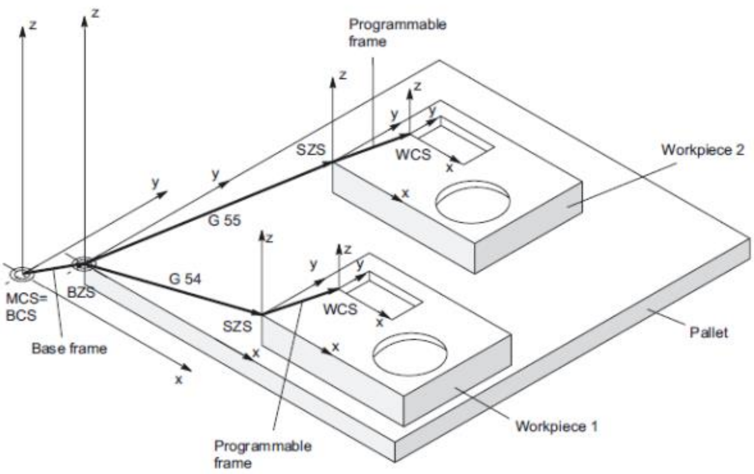


Picture 1.9: Basic offset between BCS and BZS

The basic offset comprises the following frames:

- External zero offsets ($AA_ETRANS[<Axis>])
- Handwheel (DRF) offset ($AC_DRF[<Axis>])
- Superimposed motion ($AA_OFF[<Axis>])
- Chained system frames (e.g. $P_PARTFR Rotary table reference)
- Chained basic frames ($P_NCBFR[n], $P_CHBFR[n])

Notes



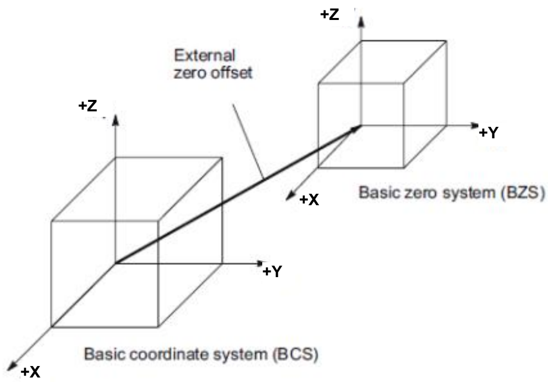
Picture 1.9: Example of the use of the basic offset

External zero offsets (additive offsets)

The "zero offset external" is an axial offset. Unlike with frames, no components for rotation, scaling and mirroring are possible.

The offset values are set via:

- PLC (by describing system variables)
- The operator panel (from menu "Current zero offsets")
- NC Program (By assigning to system variable $AA_ETRANS[axis])



Picture 1.8: Zero offset external between BCS and BZS

Note:

The system frame for the external zero offset can be configured through the machine data: ***MD28080 $MC_MM_SYSTEM_FRAME_MASK*** (Attention! System frames SRAM). If not configured the external zero shift is interpolated as usual as an overlaid movement of the axis.

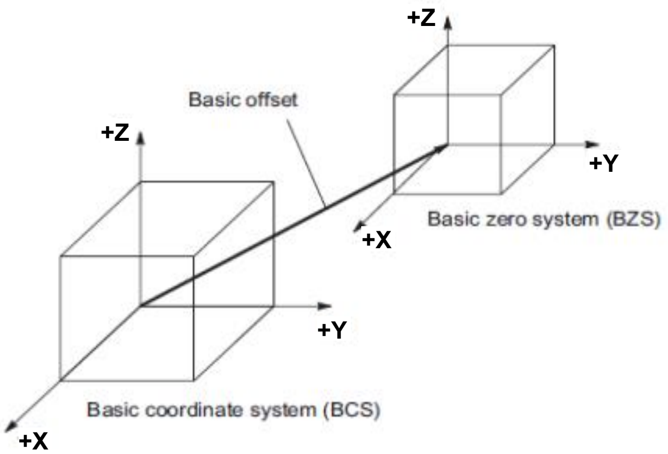
Notes

DRF offset

The DRF offset enables the adjustment of an additional incremental zero offset for geometry and additional axes in the basic coordinate system through handwheel. The DRF offset can be read via the axis-specific system variable: $AC_DRF [<Axis>]

Overlaid movements

The "Superimposed motion" of the programmed axis can only be accessed from synchronized actions via the system variable $AA_OFF[axis].



Picture 1.9: Basic offset between BCS and BZS

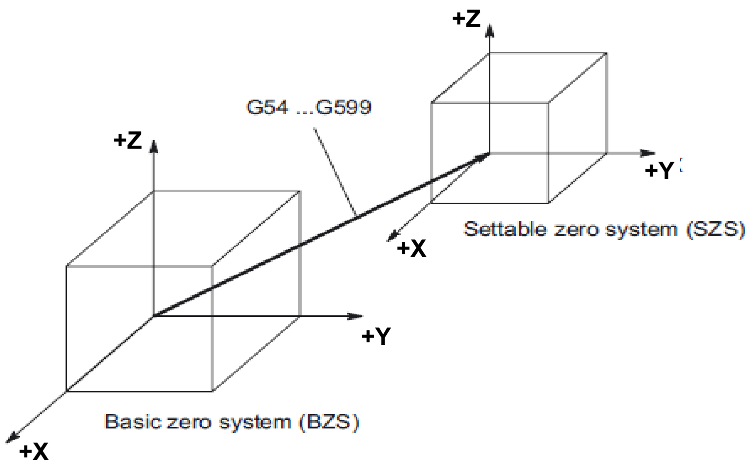
<u>Basic zero system (BZS)</u>

The basic zero system (BZS) is the basic coordinate system with a basic offset. Activation of the basic offset takes place with the **G500** instruction.

<u>Settable zero system (SZS)</u>

The "settable zero system" (SZS) is the workpiece coordinate system WCS with programmable FRAME (as seen from the perspective of the WCS). The workpiece zero is specified by the settable FRAMES **G54...G599**.

Programmable offsets act from the "settable zero system". All programmable offsets refer to the "settable zero system".



Picture 1.11: Settable FRAME G54 ... G599 between BZS and SZS

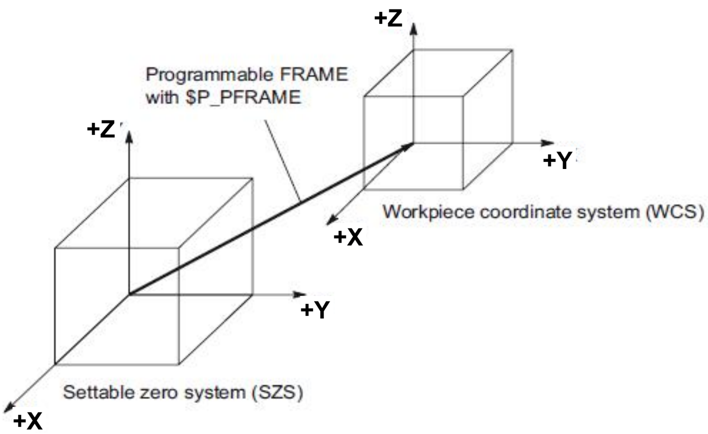
Notes

Workpiece coordinate system (WCS)

The “workpiece coordinate system” (WCS) is the programming basis.

The workpiece coordinate system comprises the following programmable frames:

- Tool reference $P_TOOLFRAME (TOROT, TOFRAME)
- Workpiece reference $P_WPFRAME
- Programmed WO $P_PFRAME (TRANS, ROT, SCALE, MIRROR)
- Cycle reference $P_CYCFRAME (TRANS, ROT, SCALE, MIRROR)



Picture 1.11: Programmable FRAME between SZS and WCS

Overview of various system frames on the HMI

Work offset - Overview [mm]

	X	Y	Z	A	C
DRF	0.000	0.000	0.000	0.000	0.000
Rotary table ref.	0.000	84.531	163.300	0.000	0.000
Basic reference	0.000	0.000	0.000	0.000	0.000
Total basic WO	0.000	0.000	0.000	0.000	0.000
G54	0.000	0.000	0.000	0.000	0.000
Tool reference	0.000	0.000	0.000	0.000	0.000
Workpiece ref.	0.000	0.000	0.000	0.000	0.000
Programmed WO	0.000	0.000	0.000	0.000	0.000
Cycle reference	0.000	0.000	0.000	0.000	0.000
Total WO	0.000	84.531	163.300	0.000	0.000

DRF: Hand wheel offset
Rotary table reference: $P_PARTFRAME (TCARR and PAROT)
Basic reference: Active channel basic frame $P_CHBFR[n] (BZS G500)
Total basic WO: Complete basic frame $P_ACTBFRAME
G54: Active settable zero frame $P_UIFR[n] (SZS)
Tool reference: $P_TOOLFRAME (TOROT or TOFRAME)
Workpiece ref.: $P_WPFRAME
Programmed WO: $P_PFRAME (TRANS, ATRANS, ROT, AROT, ...)
Cycle reference: Cycle frame $P_CYCFRAME
Total WO: Complete zero frame $P_ACTFRAME

Notes

WCS actual-value display in WCS or SZS

The actual values of the axes in the machine coordinate system (MCS) or the WCS can be displayed on the HMI operator interface. For displays in WCS, the actual values can also be displayed in relation to the SZS.

The corresponding parameterization takes place through the following machine data:

- ***MD51037 $MNS_ENABLE_COORDINATE_ACS*** (Enable settable coordinate system)
- ***MD51038 $MNS_SET_ACT_VALUE*** (Selection actual-value display)

Value MD51037	Meaning
0	Actual-value display in relation to the WCS
1	Actual-value display in relation to the SZS

Value MD51038	Meaning
0	**"Set WO"** in Jog mode not available
1	Is used if a settable zero offset e.g. G54 is active. **"Set WO"** for G500 not available (System frame is no longer used).

Example:

Actual-value display in relation to the WCS or SZS

NC Code	Actual value display: Axis X (WCS)	Actual value display: Axis X (SZS)
N10 X100	100	100
N20 X0	0	0
N30 $P_PFRAME=CTRANS(X,10)	0	10
N40 X100	100	110

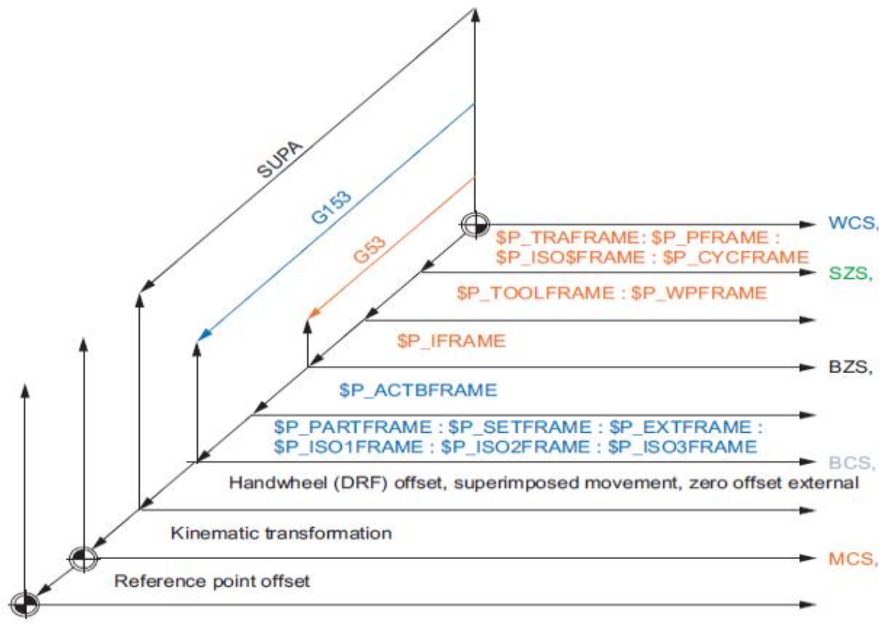
<u>Note:</u>

When "Actual-value display in relation to the SZS" is active, the WCS is still displayed on the HMI operator interface as the coordinate system to which the actual-value display relates.

Notes

Suppression of frames

Command	Meaning
G53	Non-modal suppression of the following frames: • System frame for cycles • Programmable frame • System frame for transformations, workpieces, TOROT and TOFRAME • Active settable frame
G153	Non-modal suppression of the following frames: • System frame for cycles • Programmable frame • System frame for TOROT and TOFRAME, workpieces • Active settable frame • All channel-specific and NCU global basic frames • System frames for PAROT, PRESET, scratching, ext. ZO
SUPA	Implicit preprocessing stop and non-modal suppression of frames analog G153 and additional • Hand wheel offsets (DRF) • [ext. zero offset] • Overlaid motion
DRFOF	Deactivate (clear) the hand wheel offsets (DRF)



Frame suppressions **SUPA**, **G153** and **G53** lead to the WCS, SZS and possibly the BZS jumping when frame suppression is active. The behavior for position display and predefined position variables can be changed with the following machine data: ***MD24020 $MC_FRAME_SUPPRESS_MODE***

Notes

Overview of the data management frames

Frame-types

The following frame types are available:

- System frames (see diagram)
- Basic frames ($P_NCBFR[n], $P_CHBFR[n])
- Settable work offset frames ($P_UIFR[n])
- Programmable frames ($P_PFRAME[n])

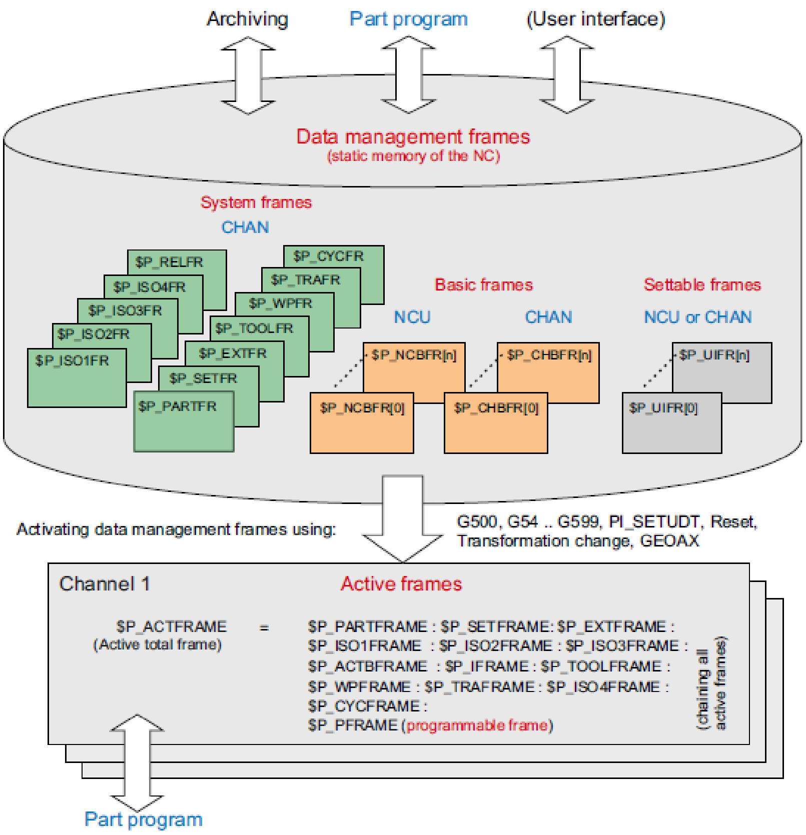
Apart from the programmable frame, all types have a frame in the data management (data management frame) and an active frame. For a programmable frame, there is only one **active** frame.

Writing frames

Data management frames and active frames can be written from the part program. Only data management frames can be written via the user interface.

Archiving frames

Only data management frames can be archived.



Notes

System frames
System frames are only described by system functions, such as PRESET, Set WO, zero offset external and oblique processing .There are up to seven system frames per channel.

Bit	Default	System frame
0	1	PRESET and Set WO
1	0	Zero offset external via system frames
2	0	TCARR and PAROT with orientable tool carrier (rotary table)
3	0	TOROT and TOFRAME
4	0	Frame for workpiece reference points
5	1	Frame for cycles
6	0	Frame for selection and deselection of transformations
7	0	For ISO G51.1 Mirror $P_ISO1FR
8	0	For ISO G68 2D ROT $P_ISO2FR
9	0	For ISO G68 3D ROT $P_ISO3FR
10	0	For ISO G51 Scaling $P_ISO4FR
11	0	For relative coordinate systems $P_RELFR

The valid system frames in the channel can be defined via machine data:

MD28082 $MC_MM_SYSTEM_FRAME_MASK (System frames SRAM)
(Attention !!! This MD reconfigures the memory SRAM)

The system frame for PRESET and Set WO and the system frame for cycles are the default. Channel-specific system frames are configured as bit codes, in accordance with the table below:

System frames in data management

The system frames are stored in the static NC memory and can, therefore, be archived and reloaded. System frames in data management can be read and written in the program using the following variables:

System frames in the data management are either activated directly with the system function **TOROT, PAROT,** directly, or with **G54** to **G599,** e.g. activated.

System variable	Description
$P_SETFR	System frame for PRESET and scratching (Set-Frame)
$P_EXTFR	System frame for zero offset external (Ext.-Frame)
$P_PARTFR	System frame for TCARR and PAROT (Part-Frame)
$P_TOOLFR	System frame for TOROT and TOFRAME (Tool-Frame)
$P_WPFR	System frame for workpiece ref. points (Workpiece-Frame)
$P_CYCFR	System frame for cycles (Cycle-Frame)
$P_TRAFR	System frame for transformations (Transformation-Frame)

Example:
$MC_MM_SYSTEM_FRAME_MASK = 'B001101' means:
There are three system frames, one for the actual value setting, one for ***PAROT*** and one for ***TOROT*** and ***TOFRAME.***
The system frame mask is used to specify whether a system frame is available for the corresponding function. If a frame is not configured, the function may be rejected with an alarm.

Notes

Active system frames

The active system frames are the frames, which are active in the main run. An appropriate current system frame exists for each current system frame in the data management. Only with the activation of the data management frame are the values taken into account with regard to the preprocessing.

The following current system frames exist:

$P_SETFRAME
In the part program, the variable $P_SETFRAME can be used to read and write the current system frame for PRESET and scratching.

$P_EXTFRAME
In the part program, the variable $P_EXTFRAME can be used to read and write the current system frame for the zero offset external.

$P_PARTFRAME (rotary table reference)
In the part program, the variable $P_PARTFRAME can be used to read and write the current system frame for **TCARR** and **PAROT** for orientable tool holders.

$P_TOOLFRAME (tool reference)
In the part program, the variable $P_TOOLFRAME can be used to read and write the current system frame for **TOROT** and **TOFRAME**.

$P_WPFRAME (tool reference)
In the part program, the variable $P_WPFRAME can be used to read and write the current system frame for setting work-piece reference points.

$P_CYCFRAME
In the part program, the variable $P_CYCFRAME can be used to read and write the current system frame for cycles.

$P_TRAFRAME
In the part program, the variable $P_TRAFRAME can be used to read and write the current system frame for transfor-mations.

$P_ISO1FRAME, $P_ISO2FRAME, $P_ISO3FRAME, $P_ISO4FRAME
One can read and write the current system frames for special ISO language commands in the part program through the variables.

$P_RELFRAME
In the part program, the variable $P_RELFRAME can be used to read and write the current system frame for relative coor-dinate systems.

$P_ACSFRAME
The currently resulting frame that is defined by the ENS-(ACS) coordinate system, can be read and written through the $P_ACSFRAME variable.

Note:

All the above variables return a zero frame if the system frame is not configured through
MD28082 $MC_MM_SYSTEM_FRAME_MASK .

Notes

Basic frames $P_NCBFR[n], $P_CHBFR[n]

The number of basic frames in the channel can be configured via the machine data: *MD28081 $MC_MM_NUM_BASE_FRAMES*
(Attention !!! This MD reconfigures the memory SRAM)

The minimum configuration is designed for at least one basic frame per channel. A maximum of 16 basic frames per channel is possible. In addition to the 16 basic frames, there can also be 16 NCU-global basic frames in the channel.

System variable **$P_CHBFR[n]** can be used to read and write the basic frame field elements. While writing a basic frame field element, the chained total frame is not activated. Instead, the activation takes place only after a **G500** instruction is executed.

The variable is used primarily for storing write operations to the basic frame on HMI or PLC. These frame variables are saved by the data backup.

Example of a channel-specific base frame in the HMI menu:

Settable work offset frames $P_UIFR[0], $P_UIFR[n]

The number of NCU global settable frames is set through the following machine data:
MD18601 $MN_MM_NUM_GLOBAL_USER_FRAMES
(Attention !!! This MD reconfigures the memory SRAM)

Work offset – basic [mm]

				X	Y	Z	A	C
1. Channel Basic WO				0.000	0.000	0.000	0.000	0.000
Fine				0.000	0.000	0.000	0.000	0.000

The number can be between 0 and 100. If the MD has a value greater than zero, there are only NCU global settable frames, otherwise the following machine data specifies the number of channel-specific settable frames:
MD28080 $MC_MM_NUM_USER_FRAMES
(Attention !!! This MD reconfigures the memory SRAM)

System variable **$P_UIFR[n]** can be used to read and write the frame field elements. The frame is not activated simultaneously when writing a field element, but rather activation only takes place on execution of a **G54** to **G599** instruction. For NCU global frames, the changed frame only becomes active in those channels of the NCU, which execute a **G54** to **G599** instruction. The variable is used primarily for storing write operations from HMI or PLC. These frame variables are saved by the data backup.

Example of a channel specific settable frame in the HMI menu:

Work offset – G54 ... G519 [mm]

				X	Y	Z	A	C
G54				10.000	90.000	40.000	0.000	0.000
Fine				0.000	0.000	0.000	0.000	0.000
G55				0.000	0.000	0.000	0.000	0.000
Fine				0.000	0.000	0.000	0.000	0.000
G56				0.000	0.000	0.000	0.000	0.000
Fine				0.000	0.000	0.000	0.000	0.000
G57				0.000	0.000	0.000	0.000	0.000
Fine				0.000	0.000	0.000	0.000	0.000

Notes

Settable Coarse and Fine offset

The total work offset (Total WO) of a settable frame consists of two offset components, the coarse offset and the fine offset.

The fine offset is parameterized via the machine data:
MD18600 $MN_MM_FRAME_FINE_TRANS
(Attention !!! This MD reconfigures the memory SRAM)

Value	Meaning
0	The fine offset cannot be entered or programmed.
1	Fine offset possible for settable frames, basic frames and the programmable frame via frame instruction or program.

Total translation

Work offset – active [mm]				X	Y	Z	A	C
G54				50.500	0.000	0.000	0.000	0.000
Total WO				50.500	0.000	0.000	0.000	0.000

Coarse
or
absolute Translation

Fine
or
additive translation

Work offset – details: G54 [mm]	Coarse	Fine			
X	50.000	0.500	0.000	1.000	☐
Y	0.000	0.000	0.000	1.000	☐
Z	0.000	0.000	0.000	1.000	☐
A	0.000	0.000		1.000	☐
C	0.000	0.000		1.000	☐
SP1	0.000	0.000		1.000	☐

Settable axial frame

An axial frame contains the frame values of an axis.
Example of the data structure of an axial frame in the HMI menu "Work offset - details":

Work offset – details: G54 [mm]	Coarse	Fine			
X	50.000	0.500	0.000	1.000	☐

G54 [mm]“:

Settable channel-specific frame

A channel-specific frame contains the frame values for all channel axes (geometry, special and machine axes).

- Geometry axes: X, Y, Z

Notes

Example of the data structure of a channel-specific frame in the HMI menu "Work offset - details":

Work offset – details: G54 [mm]	Coarse	Fine			
X	10.000	0.000	0.000	1.000	☐
Y	90.000	0.000	0.000	1.000	☐
Z	40.000	0.000	0.000	1.000	☐
A	0.000	0.000		1.000	☐
C	0.000	0.000		1.000	☐
SP1	0.000	0.000		1.000	☐

Example of the data structure of a settable system frame on the HMI menu “Work offset -active [mm]”:

Work offset – active [°]				X	Y	Z	A	C
Machine act value				0.000	0.000	1000.000	0.000	0.000
G54				10.000	90.000	40.000	0.000	0.000
Total WO				10.000	90.000	40.000	0.000	0.000

Effect
On activation of a frame a static coordinate transformation, based on the frame values and a defined calculation rule, is being executed for the axes included in the frame.
Example:
Writing of a course translation into a settable work offset, e.g. G54: $P_UIFR[1] = CTRANS(X,10,Y,90, Z,40)

Example of the data structure of a settable system frame in the HMI menu "Work offset - G54 ... G57 [mm]":

Work offset – G54 ... G519 [mm]				X	Y	Z	A	C
G54				10.000	90.000	40.000	0.000	0.000
Fine				0.000	0.000	0.000	0.000	0.000
G55				0.000	0.000	0.000	0.000	0.000
Fine				0.000	0.000	0.000	0.000	0.000
G56				0.000	0.000	0.000	0.000	0.000
Fine				0.000	0.000	0.000	0.000	0.000
G57				0.000	0.000	0.000	0.000	0.000
Fine				0.000	0.000	0.000	0.000	0.000

Frame chaining

The single frame components can be combined into a settable system frame for example in G54 using the chain operator **":"** as follows:
$P_UIFR[1] = CTRANS(X,10,Y,90,Z,40) : CROT(Z,30)

Example of the data structure of a settable system frame on the HMI:

"Work offset - details: G54 [°]":

Work offset – details: G54 [°]	Coarse	Fine			
X	10.000	0.000	0.000	1.000	☐
Y	90.000	0.000	0.000	1.000	☐
Z	40.000	0.000	30.000	1.000	☐
A	0.000	0.000		1.000	☐
C	0.000	0.000		1.000	☐
SP1	0.000	0.000		1.000	☐

Notes

_Programmable frames $P_PFRAME_

Programmable frames are available only as active frames.

This frame is reserved for the programmer and is generally only active during program run.

The programmable frame can be maintained during **RESET** with setting of the following machine data *MD24010 $MC_PFRAME_RESET_MODE*

Value MD24010	Meaning
0	Programmable frame deactivated after RESET.
1	Programmable frame remains activated after RESET.

This functionality is important, most of all if after a **RESET** one still wants to retract out of an oblique hole.

Programmable frame for coarse– and fine offset

The translation component of the programmable frame is split into an absolute component and a component for the total of all additively programmed translations.

The absolute component can be changed using **TRANS, CTRANS** or by writing the translation components , in which the additive component is set to zero

The absolute component of the translation is stored in the **coarse offset** component and the additive translation component is stored in the **fine offset** component. To this end, the programmable frame or the fine offset is expanded.

The fine component is transferred on saving the programmable frame in a local frame variable (LUD or GUD) and on rewriting.

The table below shows the effect of various program commands on the absolute and additive translation.

NC-commands/ System variables	Coarse or absolute translation	Fine or additive translation
N10 TRANS X10	10	0
N20 ATRANS X10	20	0
N30 CTRANS(X,10)	10	0
N40 CTRANS( )	0	0
N50 CFINE(X,10)	0	10
N60 $P_PFRAME[X,TR] = 10	10	0
N70 $P_PFRAME[X,FI] = 10	unchanged	10

Notes

Example 1:

Programming an axial zero offset (coarse) of 50 mm in X
$P_PFRAME[X, TR]=50

Example of the data structure of a programmable frame in the HMI:
“Work offset-active [mm]“:

Work offset – active [mm]

				X	Y	Z	A	C
G54				0.000	0.000	0.000	0.000	0.000
Programmed WO				50.000	0.000	0.000	0.000	0.000
Total WO				50.000	0.000	0.000	0.000	0.000

Example of the data structure of the programmed complete frame in the HMI menu "Work offset - details: Programmed work offset [mm]":

Work offset – details: Programmed WO [mm]

	Coarse	Fine			
X	50.000	0.000	0.000	1.000	☐
Y	0.000	0.000	0.000	1.000	☐
Z	0.000	0.000	0.000	1.000	☐
A	0.000	0.000		1.000	☐
C	0.000	0.000		1.000	☐
SP1	0.000	0.000		1.000	☐

Example 2:

Frame components or complete frames can be combined to form a complete frame using the chain operator **":"**, for example as programmable work offset as follows:

$P_PFRAME = CTRANS(X, 50) : CFINE(X, 0.1)

Example of the data structure of the programmed complete frame in the HMI menu "Work offset - active [mm]":

Work offset – active [mm]

				X	Y	Z	A	C
G54				0.000	0.000	0.000	0.000	0.000
Programmed WO				50.100	0.000	0.000	0.000	0.000
Total WO				50.100	0.000	0.000	0.000	0.000

Example of the data structure of the programmed complete frame in the HMI menu "Work offset - details: Programmed WO [mm]":

Work offset – details: Programmed WO [mm]

	Coarse	Fine			
X	50.000	0.100	0.000	1.000	☐
Y	0.000	0.000	0.000	1.000	☐
Z	0.000	0.000	0.000	1.000	☐
A	0.000	0.000		1.000	☐
C	0.000	0.000		1.000	☐
SP1	0.000	0.000		1.000	☐

Note:

Programmable frames are available only as active frames.

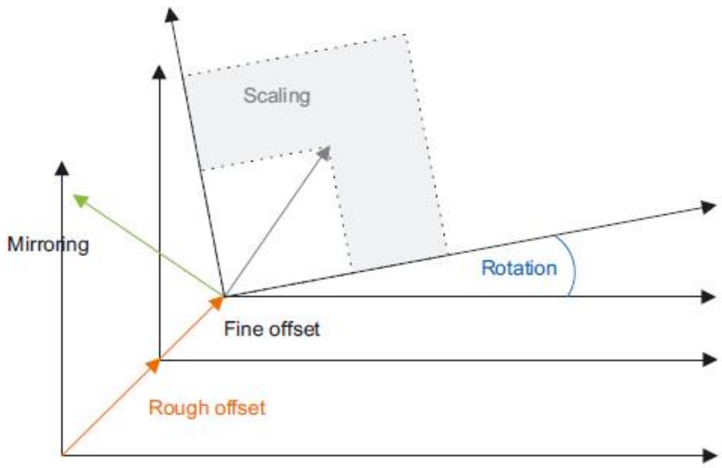
Notes

Programmable frame components

Offset, rotation, scaling and mirroring

General explanation

A frame is a closed calculation rule that translates Cartesian coordinate system in relation of the Settable zero system (SZS) into another and is a data structure that contains values for offset (TRANS), fine offset (FINE), rotation (ROT), mirroring (MIRROR) and scaling (SCALE) for the geometry and special axes.



A frame consists of the following components:

Frame components	Programmable with:	
Translation	Coarse	TRANS (absolute translation component) ATRANS (additive translation component) CTRANS (writing a course offset to a system frame) G58 (programmable axial zero offset ISO)
	Fine	CFINE (writing a fine offset to a system frame) G59 (programmable axial zero offset ISO)
Rotation		ROT / ROTS (absolute rotation component) AROT / AROTS (additive rotation component) CROT (writing a rotation to a system frame)
Scaling		SCALE (absolute scaling component) ASCALE (additive scaling component)
Mirroring		MIRROR (absolute mirror component) AMIRROR (additive mirror component)

Note:

The rough and fine offsets, scaling and mirroring can be programmed for geometry and special axes. A rotation can only be programmed for geometry axes.

Notes

Translation (offset) TRANS, ATRANS and CTRANS

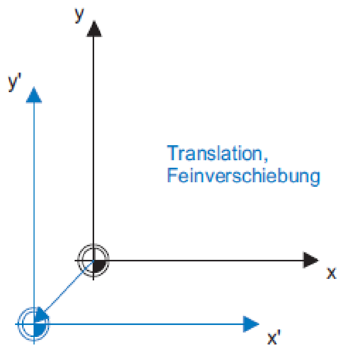
***Explanation of the coarse and fine offset*:**

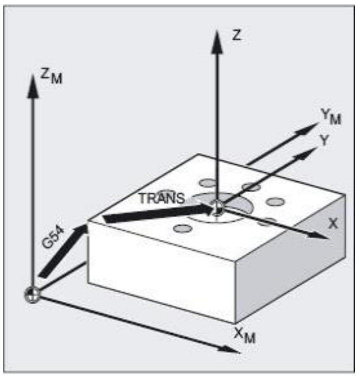
The coarse offset is normally specified by the machine operator via the actual value setting function in JOG mode. The fine offset can be set by the machine operator within certain input limits in the work offset table (e.g. G54).

The programmable total offset of all geometry axes and special axes is specified in the program (WCS) with **TRANS**.

The fine offset is written in the program with the command **CFINE (x, .., y, ..)** and the coarse offset with the command **CTRANS(...)** in the system frame. Coarse and fine offset are added to form the total offset.

$P_UIFR[1] = CTRANS(x, 10) : CFINE(x, 0.1)
$P_UIFR[1] = CFINE(x, 0.5, y, 1.0, z, 0.1)





Programming

The translation is programmed via the following NC program commands:

NC command	Description
TRANS X.. Y.. Z..	Programmable absolute offset in relation to the active settable work offset G500, G54-G599.
ATRANS X.. Y.. Z	Programmable additive offset based on the last programmed absolute offset.
TRANS	Deletes the programmable offset for all axes. The last called storable WO G54 to G599) is then the reference again for the WCS.
CTRANS	Writes a coarse offset to a system frame, e.g. storable WO G54-G599. (See also Section 2.7-2.8 "Programmable frames")
CFINE	Writes a fine offset to a system frame, e.g. storable WO G54-G599. (See also Section 2.7-2.8 "Programmable frames")

Notes

__Parameterization__

A fine offset can only be programmed if:
MD18600 $MN_FRAME_FINE_TRANS = 1
(Attention !!! This MD reconfigures the memory SRAM)

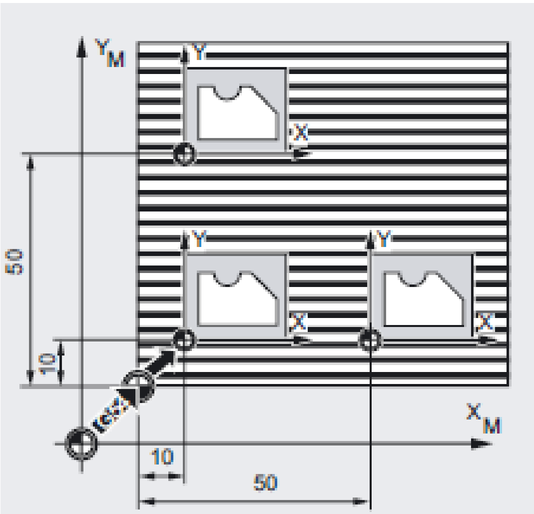
If this is not the case, every assignment of a fine offset to settable frames and to the basic frame is rejected with the alarm "FRAME: Fine offset not possible".

Activation of the coarse and fine offsets

A fine offset changed by the operator only becomes active after the appropriate frame is activated, i.e. the activation is via **G500, G54...G599.** An activated fine offset of a frame is active as long as the frame is active. In the display of the offset of the current frame, the additive offset of the coarse and fine offset is output (see also Section 2.5).

Application example:

The same contour is shown here several times at different positions. The machining of the contour is in a subprogram. With the aid of the offset, the WCS can be moved to the respective reference point and then the subprogram called.




```
N10 G17 G54             ;Working plane X/Y, WO
N20 G0 X0 Y0 Z2         ;Approach starting point
N30 TRANS X10 Y10       ;Abs. offset
N40 L10                 ;Subprogram call
N50 TRANS X50 Y10       ;Abs. offset
N60 L10                 ;Subprogram call
N50 TRANS X10 Y50       ;Abs. offset
N60 L10                 ;Subprogram call
N70 M30                 ;End of program
```

Notes

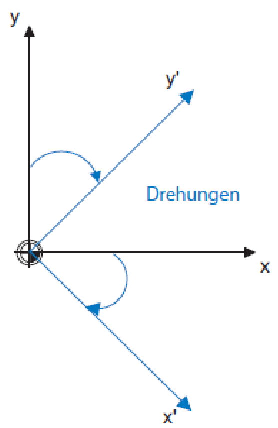
Rotations ROT, AROT, CROT

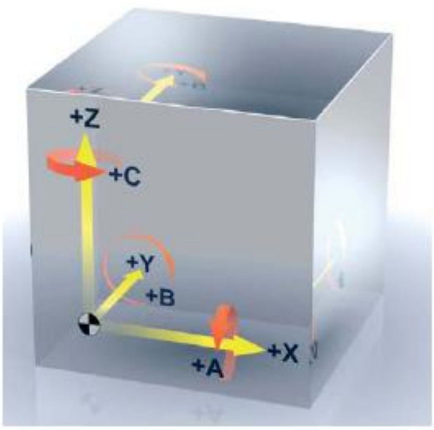
Explanation of the function

The direction of rotation around the coordinate axes (geometry axes) is defined by a right-hand, rectangular coordinate system with the axes X, Y and Z.

The direction of rotation is positive, if the rotational motion is clockwise when looking in the positive direction of the coordinate axis.

A, B and C designate the rotations whose axes are parallel to X, Y and Z.





Order of rotation

The corresponding order of rotation in the frame is parameterized through the machine data *MD10600 $MN_FRAME_ANGLE_INPUT_MODE* and is independent of the order of the programmed axes identifiers in the NC block. It is possible to program up to three axes rotations in one NC block.

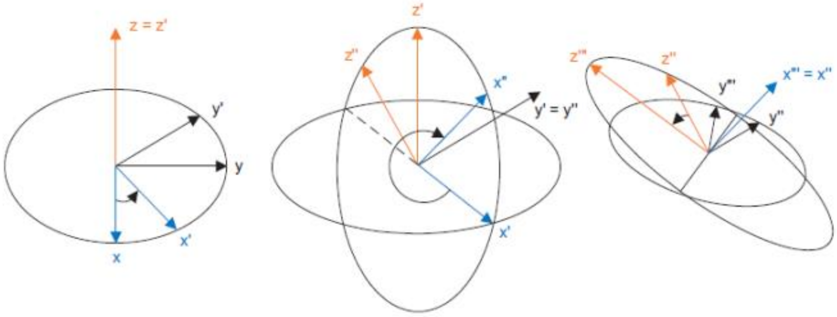
Value MD10600	Meaning
0	Programming Euler angles
1	Programming RPY angles (default)

Notes

RPY angles (roll, pitch, yaw)

The rotations with RPY angles are performed in the order Z, Y', X''.

Rotation around Z   Rotation around Y'   Rotation around X"



The angles are only unambiguous in the following ranges:

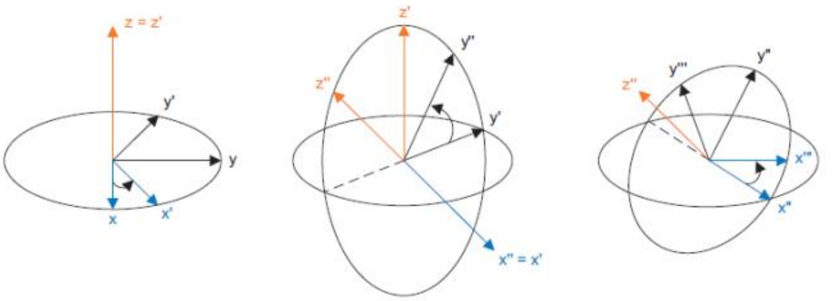
-180 <= x <= 180
-90 < y < 90
-180 <= z <= 180

EULER angles

The rotations with EULER angles
are performed in the order Z, X', Z''.

The angles are only unambiguous in the following ranges:

Rotation around Z   Rotation around X'   Rotation around Z"



0 <= x < 180
-180 <= y <= 180
-180 <= z <= 180
In these ranges, the written angles can also be read back unambiguously. If rotations are entered that are larger than the specified angles, they are converted to a display that fits in the specified range.
When writing and reading frame rotation components, these limits should be observed so that when writing and reading or repeated writing the same results are received.

Example of RPY		
$P_UIFR[1]= CROT(X,10,Y,90,Z,270)	Returns when reading back:	$P_UIFR[1]= CROT(X,10,Y,90,Z,-90)
$P_UIFR[1]= CROT(X,190,Y,0,Z,-200)	Returns when reading back:	$P_UIFR[1]= CROT(X,-170,Y,0,Z,160)

Notes

Programming

A rotation is programmed via the following NC program commands:

NC command	Description
ROT	Absolute rotation in space in relation to the active work offset G500, G54 to G599.
RPL	Rotation in the plane: Angle through which the coordinate system is rotated, is specified by the working planes G17-G19.
AROT	Additive rotation in space in relation to the active work offset G500, G54 to G599.
X Y Z	For ROTS and AROTS, the solid angle is defined. For ROT and AROT, the rotation around the geometry axes is defined. The order of the rotations is defined in the machine data MD10600. Whereby, rotation according to the RPY principle (= roll, pitch, yaw) is the default setting.
ROTS	Absolute rotation with solid angle in relation to the active work offset G500, G54 to G599.
AROTS	Additive rotation with solid angle in relation to the active work offset G500, G54 to G599.
CRPL	Constant rotation in the plane (see Parameters page 26).
CROTS, CROT	For writing an absolute rotation as solid angle or axis-by-axis (RPY, Euler) to a system frame.

:

Examples:

Programmable rotation absolute around the geometry axes X and Y of 10° (only active during program run):

ROT X10 Y10

Programmable rotation absolute in the active plane of 90° (with active G17 the XY-plane is rotated around Z) (only active during program run):

CRPL(0,90)

Writing a rotation around the geometry axes X and Y of 10° to the storable work offset G54:

$P_UIFR[1]=CROT(X,10,Y,10)

Writing a single rotation around the geometry axis X of 10° to the storable work offset G54 (system frame):

$P_UIFR[1,X,RT]=10

Notes

Constant Rotation Plane - CRPL

With the predefined "Constant Rotation Plane" function, a rotation in any plane can be programmed for every frame: FRAME CRPL(INT,REAL)

Parameters:

INT	0:	Rotation in the active plane G17, G18, G19
	1:	Rotation around z
	2:	Rotation around y
	3:	Rotation around x
REAL		Angle of rotation in degrees
	RPY:	-180 <= x <= 180
		-90 <= y <= 90
		-180 <= z <= 180
	Euler:	-180 <= x <= 180
		0 <= y <= 180
		-180 <= z <= 180
		The specified angles must be observed by the user for the reconversion. If these limits are not observed, an unambiguous reconversion is not possible. The input is not aborted with an alarm.

The advantage of this method is that no axis identifier has to be specified for a geometry coordinate axis around which a rotation is to be performed.

Chaining with frames

CRPL() can be chained with frames and the known frame functions CTRANS(), CROT(), CMIRROR(), CSCALE(), CFINE().

Examples:

$P_PFRAME = CTRANS(X,50,Y,90,Z,40) : CRPL(1,30)
$P_PFRAME = $P_PFRAME : CRPL(1,30)
$P_PFRAME = CROT(X,10) : CRPL(2,30)
$P_PFRAME = CRPL(3,30) : CMIRROR(Y)

Work offset – details: 1. Channel Basic WO [mm]

	Coarse	Fine			
X	50.000	0.000	0.000	1.000	☐
Y	90.000	0.000	0.000	1.000	☐
Z	40.000	0.000	30.000	1.000	☐
A	0.000	0.000		1.000	☐
C	0.000	0.000		1.000	☐
SP1	0.000	0.000		1.000	☐

Notes

Frame rotations with ROT, AROT, CROT (axis-by-axis)

If a frame is to be defined that describes a rotation around more than one axis, this is performed by chaining the individual rotations around the geometry axes. Whereby, a new rotation is always performed in the already rotated coordinate system. This applies to programming in a block, e.g. ***ROT X... Y... Z...,*** as well as for the formation of a frame in several blocks, e.g. in the form:

N10 ROT X...
N20 AROT Y...
N30 AROT Z…

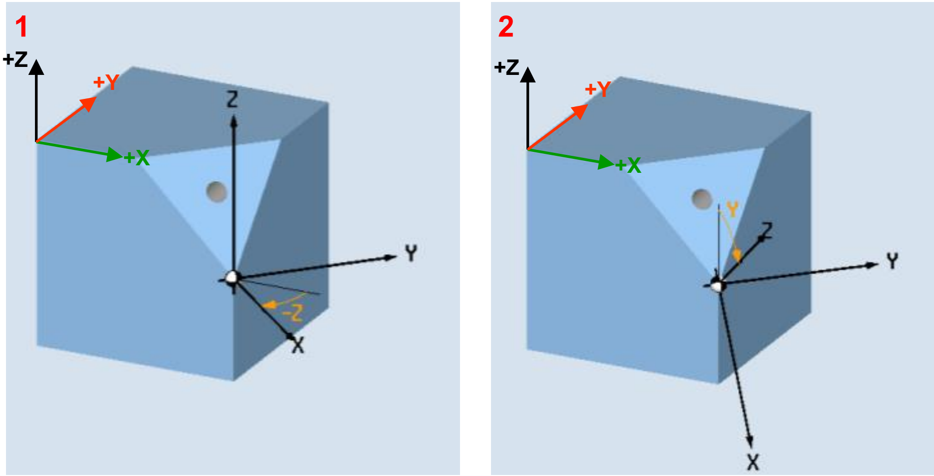


Fig. 3.1: Axis-by-axis around the geometry axes

Programming example (Fig. 3.1):

N10 ROT Z-45 Y54.736 X0
or
N10 ROT Z-45 ;(1)
N20 AROT Y54.736 ;(2)

Note:

The order of the rotations is specified with the machine data
MD10600 $MN_FRAME_ANGLE_INPUT_MODE,
and is independent of the order of the axis addresses in the NC block.

Frame rotations with ROTS, AROTS, CROTS (solid angle)

In workpiece drawings, solid angles are often used to describe inclined surfaces, i.e. the angles formed by the intersection lines with the main planes (XY, YZ, ZX plane) (see Fig. 3.2). The conversion of these solid angles to the angles of rotation of a chain is very complicated.

To simplify this, the rotations can be described directly as solid angles with the **ROTS, AROTS** and **CROTS** commands.

Notes

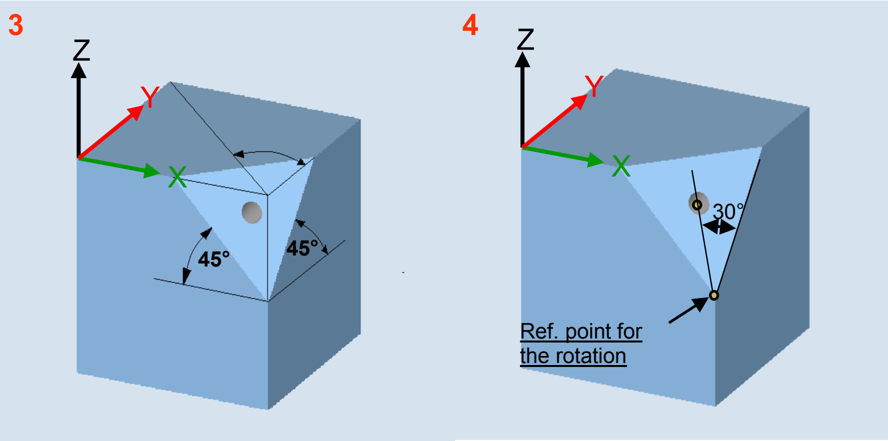


Fig. 3.2: Rotation with specification of the solid angles:

Programming example (Fig. 3.2):

N10 ROTS X45 Y45 **(3)**
N20 AROT Z30 **(4)**

The orientation of a plane in space is clearly defined through the specification of two solid angles **(3)**. The third solid angle results from the first two. A maximum of two solid angles can be programmed. If a third angle is specified, an alarm is output.

The rotations that are performed with **ROTS** and **AROTS** in this case are identical to those for **ROT** and **AROT**.

An extension of the available functionality is only obtained after two solid angles are specified.

The two programmed axes define a plane. The axis that is not programmed defines the associated third axis of an orthogonal coordinate system. For the two programmed axes, it is therefore clearly defined which is the first and which is the second axis. (Definition corresponds to the planes **G17/G18/G19**).

The example in Fig. 3.2 shows that X and Y have been programmed. Y specifies the angle that the X axis must be rotated around the Y axis so that the X axis is moved to the intersection line that the inclined plane forms with the XZ plane. This applies accordingly for the X value.

With the specification of the solid angles, the orientation of the two-dimensional coordinate system within the plane (i.e. the angle of rotation around the surface normal vector) is not defined. The position of the coordinate system is therefore defined so that the rotated first axis is in the plane formed by the first and third axes of the non-rotated coordinate system.

This means:

- When programming X and Y, new X axis is in the old ZX plane.
- When programming Z and X, new Z axis is in the old YZ plane.
- When programming Y and Z, new Y axis is in the old XY plane.

If a position of the coordinate systems is required that differs from this default setting, an additional rotation must be performed
(4) with **AROT...**.

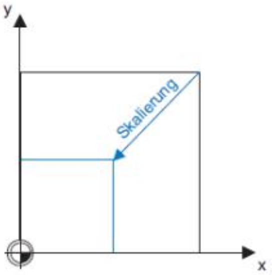
The programmed solid angles depend on the machine data:
MD10600$MN_FRAME_ANGLE_INPUT_MODE (RPY or Euler angles)

Notes

Scaling and mirroring with SCALE and MIRROR

Scaling

The programmable scaling (scale factor) for all geometry axes and special axes is programmed with **SCALE**. If a new scaling is to be based on another scaling, rotation, offset or mirroring, **ASCALE** must be programmed.

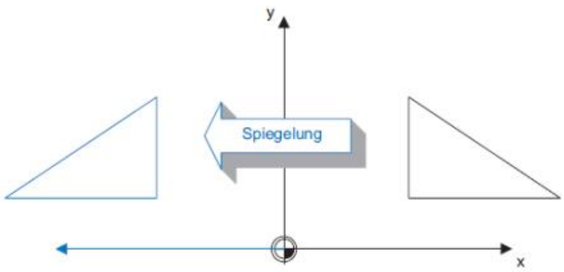


Programming

The following program commands are used for scaling:
$P_UIFR[1] = CSCALE(x,1,y,1)
SCALE x = 1y = 1
$P_UIFR[1,x,sc] = 1

Mirroring

The programmable mirroring for all geometry axes is programmed with **MIRROR**. If a new mirroring is to be based on another mirroring, rotation, offset or scaling, **AMIRROR** must be programmed.



Programming

The following program commands are used to program mirroring:
$P_UIFR[1] = CMIRROR(x,1,y,1)
MIRROR x = 1y = 1
$P_UIFR[1,x,mi] = 1

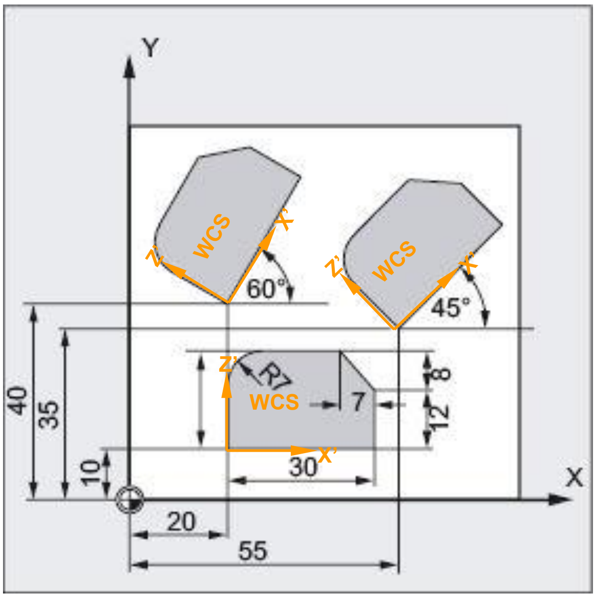
Parameterization

Which reference axis is to be mirrored can be set with the following machine data:
MD10610 MIRROR_REF_AX

Notes

Example of a frame rotation in the plane

With this workpiece, the same contour appears several times at different positions and in different orientations in the same program. In addition to the offset, rotations must also be programmed because not all contours are aligned parallel to the basic coordinate system.



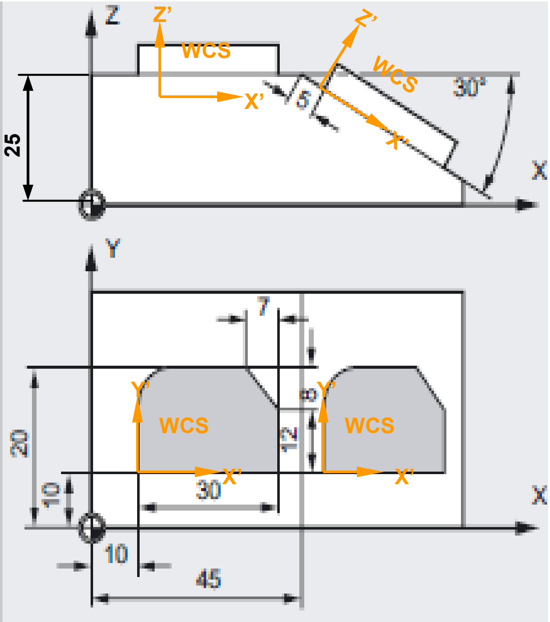

```
N10 G17 G54                 ;X/Y working plane, storable WO
N20 TRANS X20 Y10           ;Absolute offset of the WCS
N30 L10                     ;Subprogram call
N40 TRANS X55 Y35           ;Absolute offset of the WCS
N50 AROT RPL=45             ;Rotation of the XY plane through 45°
N60 L10                     ;Subprogram call
N70 TRANS X20 Y40           ;Absolute offset of the WCS
                            ;(deletes all previously programmed frames)
N80 AROT RPL=60             ;Additive rotation of the XY plane through 60°
N90 L10                     ;Subprogram call
N100 G0 X100 Y100 Z100      ;Retraction movement
N110 M30                    ;End of program

L10.SPF
N10 G0 X-10 Y-10
N20 G1 Z-5 F500
N30 G41 X0
N40 G1 Y20 RND=7
N50 X23
N60 X30 Y12
N70 Y0
N80 X0
N90 G40 X-10 Y-10
N100 G0 Z10
```

Notes

Example of a frame rotation in space

In this example, the contour is to be milled on an inclined plane below 30°. To do this, the workpiece coordinate system WCS must first be aligned with the inclined plane. The second step would then be to align the tool with the rotated workpiece coordinate system (WCS).




```
N10 G17 G54                  ;X/Y working plane, storable WO
N11 TRANS X10 Y10 Z25        ;Absolute offset of the WCS
N12 L10                      ;Subprogram call

; ***Align WCS***
N13 ATRANS X35               ;Additive offset of the WCS
N14 AROT Y30                 ;Additive rotation of the WCS
N15 ATRANS X5                ;Additive offset of the WCS

;***Align tool with AC table kinematics type P***
N16 TCARR=1                  ;Select the orientable
                             ;tool carrier
N17 TCOFR                    ;Calculate the rotary axis positions
N18 PAROT                    ;only for types P and M
N19 G0 A=$P_TCANG[1]         ;Travel to calculated A axis position
       C=$P_TCANG[2]         ;Travel to calculated C axis position
N20 L10                      ;Subprogram call
N21 G0 Z100                  ;Retract
N22 TCARR=0                  ;Deselect the orientable
                             ;tool carrier
N23 PAROTOF                  ;Deselect the frame rotation of the
                             ;workpiece reference
N22 M30                      ;End of program
```

Note:

The functions of the TCARR, TCOFR, PAROT and PAROTOF commands are explained in detail in Section 4 in this module.

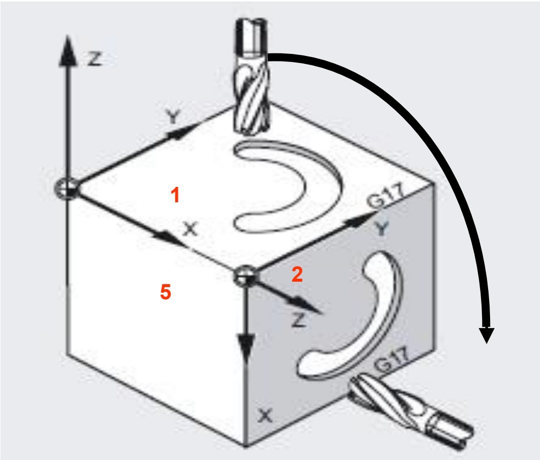
Notes

Example of multiface machining

In this example, two identical circumferential grooves are to be milled on two workpiece faces at right angles with a subprogram.

Whereby the WCS is offset and rotated in such a way that the infeed axis, the working plane and the workpiece zero in the new coordinate system on the right face 2 matches the top workpiece face 1.

The requirements for the subprogram correspond to those of the top face 1: Working plane G17, coordinate plane X/Y, infeed direction in Z.




```
N08 T1 M6
N09 S8000 M3 F1000
N10 G17 G54          ;X/Y working plane, storable WO

;Face 1:
N11 L10                    ;Subprogram call

;Face 2:
; ***Align WCS***
N12 TRANS X100       ;Additive offset of the WCS in X
N13 AROT Y90         ;Additive rotation of the WCS around the Y axis by 90°

;***Align tool with AC table kinematics type P***
N14 TCARR=1          ;Select the orientable tool carrier
N15 TCOFR            ;Calculate the rotary axis positions
N16 PAROT            ;Rotate the workpiece reference only for types P and M
N17 G0 A=$P_TCANG[1]  C=$P_TCANG[2]
N18 L10               ;Subprogram call for groove
N19 PAROTOF
N20 TCARR=0
N30 M30
```

Notes

Alternatively, the following offsets result for faces 3, 4 and 5:

```
;Face 3:
; ***Align WCS***
N12 TRANS X100 Y100
N13 AROT Z90
N14 AROT Y90
;***Align tool with AC table kinematics type P***
N15 TCARR=1
N15 TCOFR
N16 PAROT
N17 G0 A=$P_TCANG[1]  C=$P_TCANG[2]
N18 L10
N19 PAROTOF
N20 TCARR=0
```

```
;Face 4:
; ***Align WCS***
N12 TRANS Y100
N13 ROT Z180
N14 AROT Y90
;***Align tool with AC table kinematics type P***
N15 TCARR=1
N15 TCOFR
N16 PAROT
N17 G0 A=$P_TCANG[1]  C=$P_TCANG[2]
N18 L10
N19 PAROTOF
N20 TCARR=0

;Face 5:
; ***Align WCS***
N12 AROT Z270
N13 AROT Y90
;***Align tool with AC table kinematics type P***
N15 TCARR=1
N15 TCOFR
N16 PAROT
N17 G0 A=$P_TCANG[1]  C=$P_TCANG[2]
N18 L10
N19 PAROTOF
```

Notes

Swivel frames

Tool carrier TCARR and rotary table reference

Explanation of the function:

For kinematics of type P and type M, when a tool carrier is selected, an additive frame is activated (table offset of the orientable tool carrier) which takes into account the offset of the zero point as a result of the rotation of the table (rotary table reference in the WO).

The work offset is entered in a system frame ($P_PARTFR). Whereby, the translatory component of this frame is overwritten. The other contents of this frame are retained.

In order to be able to use this system frame, bit 2 must be set in the machine data
MD28082 $MC_MM_SYSTEM_FRAME_MASK.

A frame offset as a result of a tool carrier change takes effect immediately when **TCARR=...** is selected. In contrast, a tool carrier change for kinematics type M (tool and workpiece can be rotated) only takes effect immediately when a tool is active.

The activation does not cause a frame rotation or an already active rotation is not changed. In case P (table kinematics), the rotary axis position is determined from the rotation component of an active frame depending on the G code **TCOFR/**.

The calculated rotary axis positions for the first and second rotary axis are stored in the system variables **$P_TCANG[1]** and **$P_TCANG[2]**.

Through the activation of a frame, the position in the workpiece coordinates system changes without causing a compensating movement of the machine.

The relationships are shown in the following figure:

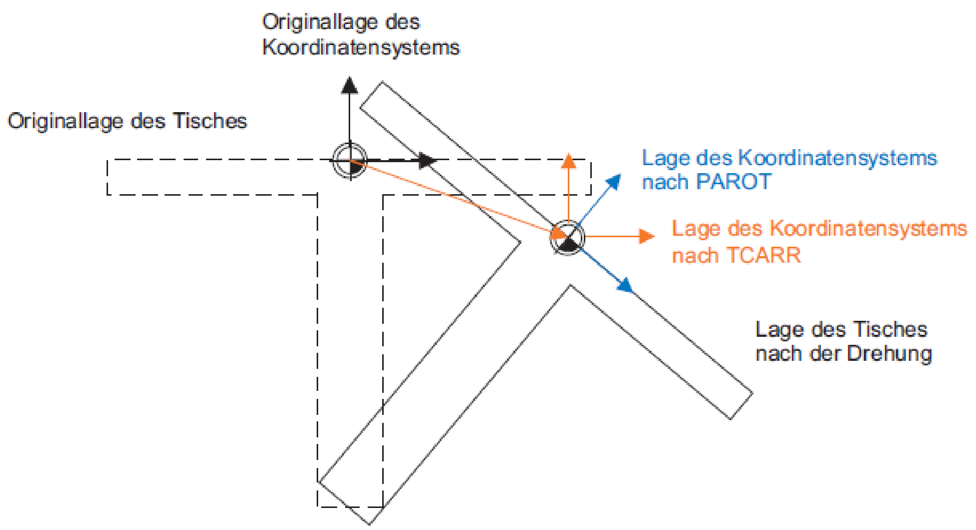


Fig. 4.2: Frame rotation with PAROT table kinematics type P

Orginal position of the table

Notes

Frame rotation of the workpiece with PAROT

Explanation of the function:

Depending on the machining task, when using rotatable tool carriers or tables, not only a work offset (whether as frame or as tool length), but also a rotation must be taken into account. However, the activation of an orientable tool carrier does not immediately result in a rotation of the coordinate system.

For rotatable tables (kinematics types P and M), the activation with TCARR does not cause a rotation of the coordinate systems at first, i.e. the zero point of the coordinate system is offset in relation to the machine and remains fixed in relation to the workpiece zero, but the orientation remains unchanged (fixed in space), (see Fig. 4.2 "Position of the coordinate system after TCARR").

If a coordinate system is required that is fixed in relation to the workpiece, i.e. compared to the original position not only offset, but also rotated corresponding to the table rotation, an appropriate rotation can be activated with **PAROT** (see Fig. 4.2 "Position of the coordinate system after PAROT").

TCARR

For kinematics of the type P or type M, TCARR enters the table offset of the orientable tool carrier as translation in the system frame ($P_PARTFR) (see Fig. 4.3: Rotary table reference).

PAROT

PAROT converts the system frame so that workpiece-related WCS results (see Fig. 4.3: Workpiece reference). The rotation activated by PAROT is included in the programmable frame ($P_PFRAME) by changing the rotation component in relation to the active tool carrier (TCARR).

Work offset – Overview [mm]

	X	Y	Z	A	C
Machine act value	0.000	0.000	1000.000	0.000	84.531
DRF	0.000	0.000	0.000	0.000	0.000
Rotary table ref.	0.000	84.531	163.300	0.000	0.000
Basic reference	0.000	0.000	0.000	0.000	0.000
Total basic WO	0.000	0.000	0.000	0.000	0.000
G54	0.000	0.000	0.000	0.000	0.000
Tool reference	0.000	0.000	0.000	0.000	0.000
Workpiece ref.	0.000	0.000	0.000	0.000	0.000
Programmed WO	0.000	0.000	0.000	0.000	0.000
Cycle reference	0.000	0.000	0.000	0.000	0.000
Total WO	0.000	84.531	163.300	0.000	0.000

TCOFR/TCOABS

After calling a defined tool carrier, TCOFR calculates the rotary axis positions in relation to the machine kinematics defined in TCARR. Whereby, active or previously programmed frames, such as rotations and offsets, are included in the calculation. In contrast to TCOFR, these are not included with TCOABS. The calculated rotary axis angles are stored in the system variables **$P_TCANG[1] - [2]**.

Notes

Sequence (see Fig. 4.1 page 34)

On a machine, the rotary axis of the table points in the positive X direction. The table is rotated through +45 degrees. A frame is then defined with **PAROT** that also has a rotation of +45 degrees around the X axis (workpiece reference).

Compared to the outside, the coordinate system has not been rotated (see Fig. 4.2 "Position of the coordinate system after TCARR"), but compared to the coupled-motion coordinate system, has been rotated through -45 degrees (see Fig. 4.2 "Position of the coordinate system after PAROT").

If this coordinate system is defined, e.g. with **ROT** X+45, and then the tool carrier selected with active **TCOFR**, an angle of **A+45** degrees is determined for the rotary axis of the tool carrier.

The **PAROT** language command is not rejected when no orientable tool carrier is active. However, such a call does not change the frame.

Parameterization

The system frame for **TCARR** and **PAROT** is configured with bit 2 in the following machine data:
MD28082 $MC_MM_SYSTEM_FRAME_MASK

If the system frame has been configured for **TCARR**, then TCARR and PAROT describe the corresponding system frame, otherwise the base frame designated by ***MD20184 $MC_TOCARR_BASE_FRAME_NUMBER*** is described.

The rotation component which describes the rotation of the table, is entered in the system frame **$P_PARTFRAME** when bit 2 of machine data ***MD28082 $MC_MM_SYSTEM_FRAME_MASK*** is set. The system frames are stored in the SRAM and are therefore retained after a reset.

Effect

It is therefore possible when using ***PAROT*** to rotate the workpiece frame in such a way that there is an inclined plane parallel to the fixed XY plane, perpendicular to the tool axis. Whereby, the tool orientation (table rotation) must be taken into account if, for example, holes are to be drilled in this orientation.

Deselection

The rotation component of the workpiece frame can be deselected with the PAROTOF command irrespective of whether this frame is in the base frame or the system frame. The offset component is deleted when a tool carrier that does not cause an offset is activated or the active tool carrier is deselected with **TCARR=0**.

Note:

With **PAROT**, translations, scaling and mirroring are retained in the active frame, but the rotation component is rotated by the rotation component of an orientable tool carrier that corresponds to the table.

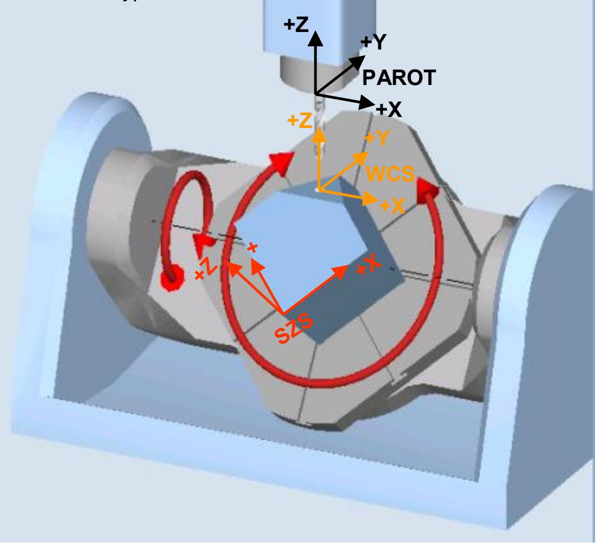
The rotation component of the part frame can be deleted with ***PAROTOF*** irrespective of whether this frame is in the base frame or the system frame. The translation component is deleted when a tool carrier that does not cause an offset is activated or when an active orientable tool carrier is deselected with ***TCARR=0***.

Notes

Example: Frame rotation of the workpiece of kinematics type P

PAROT takes into account the entire orientation change in those cases in which only the table is oriented by two rotary axes. For kinematics in which the workpiece rotates around the tool, the appropriate rotation component caused by a rotary axis is taken into account (see Fig. 4.4).

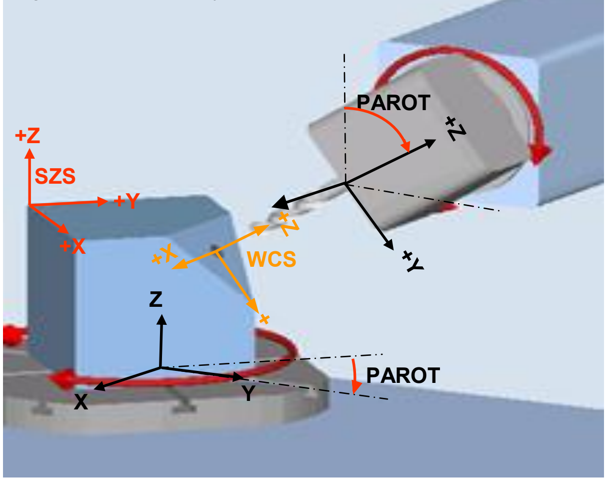
Fig. 4.4:
Kinematics type P



Example: Frame rotation of the workpiece of kinematics type M

PAROT takes into account the entire orientation change in those cases in which the table and the tool are oriented with two rotary axes. For mixed kinematics, the appropriate rotation component caused by a rotary axis is taken into account (see Fig. 4.5).

Fig. 4.5: Kinematics type M



Notes

Example 1: Swivel frame with table kinematics type P

In the following example, a hole is to be drilled on an inclined plane with the solid angles Xα=-45° and Yβ=45° (see Fig. 4.7). The machine has an AC swivel table kinematics type P (see Fig. 4.6).

Fig. 4.6:
Kinematics type P

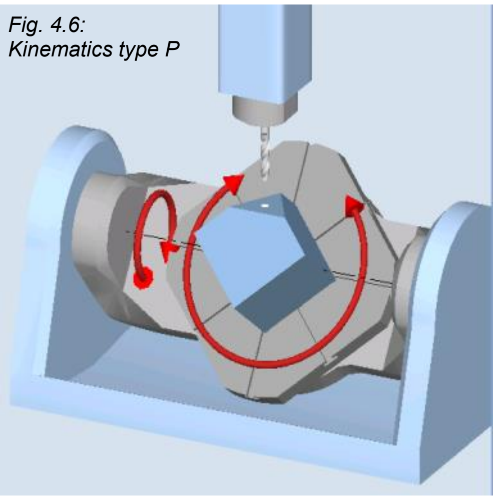
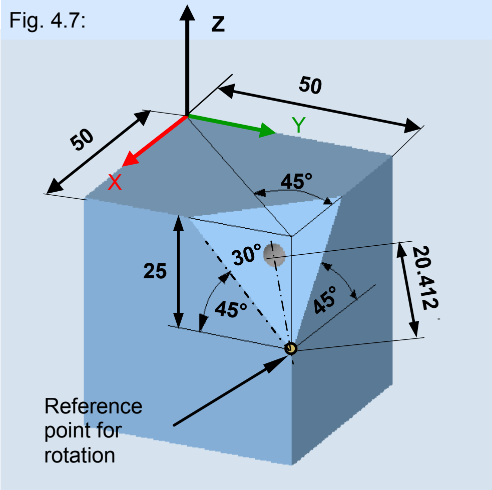


Fig. 4.7:



Orientable tool carrier 1 (swivel data record 1)

;Definition of tool carrier 1:
$TC_CARR1[1] = 0 ; X component of the 1st offset vector (I1)
$TC_CARR2[1] = 0 ; Y component of the 1st offset vector (I1)
$TC_CARR3[1] = 0 ; Z component of the 1st offset vector (I1)
$TC_CARR4[1] = 260 ; X component of the 2nd offset vector (I2)
$TC_CARR5[1] = 200. ; Y component of the 2nd offset vector (I2)
$TC_CARR6[1] = 150.02 ; Z component of the 2nd offset vector (I2)
$TC_CARR7[1] = -1 ; X component of the 1st rotary axis vector (V1)
$TC_CARR8[1] = 0 ; Y component of the 1st rotary axis vector (V1)
$TC_CARR9[1] = 0 ; Z component of the 1st rotary axis vector (V1)
$TC_CARR10[1] = 0 ; X component of the 2nd rotary axis vector (V2)
$TC_CARR11[1] = 0 ; Y component of the 2nd rotary axis vector (V2)
$TC_CARR12[1] = -1 ; Z component of the 2nd rotary axis vector (V2)
$TC_CARR13[1] = 0 ; Angle of rotation of the 1st axis
$TC_CARR14[1] = 0 ; Angle of rotation of the 2nd axis
$TC_CARR15[1] = 0 ; X component of the 3rd offset vector (I3)
$TC_CARR16[1] = 0.02 ; Y component of the 3rd offset vector (I3)
$TC_CARR17[1] = 0 ; Z component of the 3rd offset vector (I3)
$TC_CARR18[1] = -260 ; X component of the 4th offset vector (I4)
$TC_CARR19[1] = -200.02 ; Y component of the 4th offset vector (I4)
$TC_CARR20[1] = -150.02 ; Z component of the 4th offset vector (I4)
$TC_CARR21[1] = A ; Designation of the 1st rotary axis
$TC_CARR22[1] = C ; Designation of the 2nd rotary axis
$TC_CARR23[1] = "P" ; Type of tool carrier

Notes

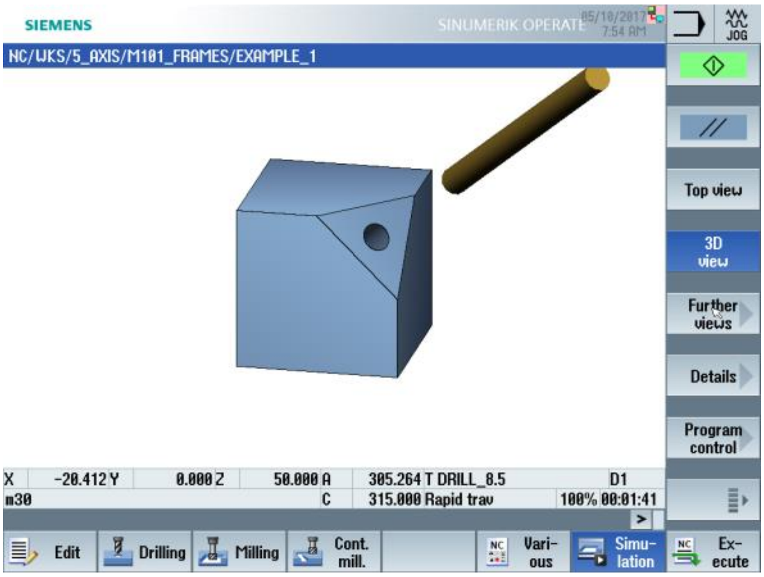
NC program:

```
T1 D1; FACING TOOL_D32
M6
G54 G17 S4000 M3
TRANS X50 Y50 Z-25 ; Translation WO reference point for rotation
AROTS X=-45 Y=45 ; Frame rotation with solid angles
AROT Z=30
TCOFR ; Calculation of the rotary axis positions in the active frame
TCARR=1 ; Selection of the orient. tool carrier
G0 X0 Y0 A=$P_TCANG[1]  C=$P_TCANG[2]  ; Pos. rotary axes
PAROT ; Activate frame rotation of the workpiece
G0 X0 Y0 Z25 ; Position 1st hole
CYCLE61(50,14.442,5,0,-25,0,32,25,2,60,0,500,41,0,1,10) ;Face milling cycle
PAROTOF ; Deactivate frame rotation of the workpiece
TCARR=0  ; Deselection of the orient. tool carrier

T2 D1; DRILL_D8.5
M6
G54 G17 S3000 M3
TRANS X50 Y50 Z-25 ; Translation WO reference point for rotation
AROTS X=-45 Y=45 ; Frame rotation with solid angles
AROT Z=30
TCOFR ; Calculation of the rotary axis positions in the active frame
TCARR=1 ; Selection of the orient. tool carrier
G0 X0 Y0 A=$P_TCANG[1]  C=$P_TCANG[2]  ; Pos. rotary axes
```

```
PAROT ; Activate frame rotation of the workpiece
G0 X-20.412 Y0 Z10 ; Position 1st hole
CYCLE82(100,0,5,-15,,1,0,1,12) ; Drilling cycle
PAROTOF ; Deactivate frame rotation of the workpiece
TCARR=0  ; Deselection of the orient. tool carrier
M30
```

Simulation



Notes

Example 2: Swivel frame with mixed kinematics type M

In the following example, a hole is to be drilled on an inclined plane with the solid angles Xα=-45° and Yβ=45° (see Fig. 4.10). The machine has mixed kinematics type M with a BC head/table kinematics type (see Fig. 4.11).

Fig. 4.10:

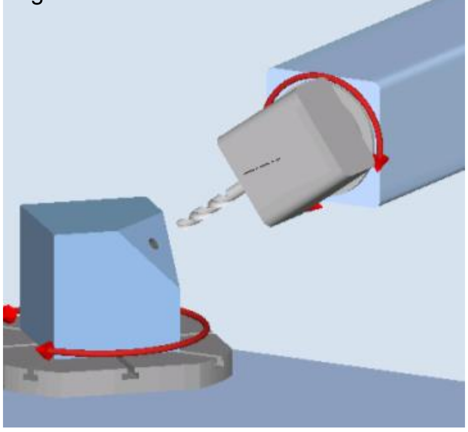
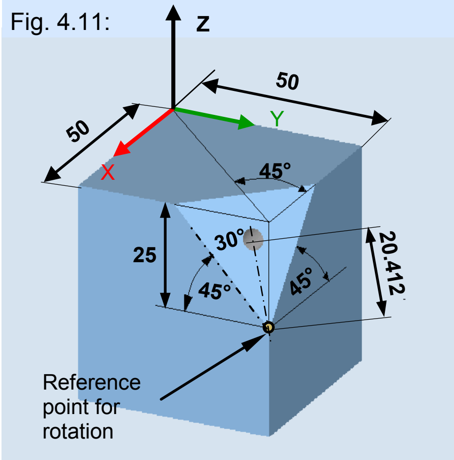


Fig. 4.11:



Orientable tool carrier 1 (swivel data record 1)

;Definition of tool carrier 2:
N30 $TC_CARR1[2] = 0 ; X component of the 1st offset vector (I1)
N40 $TC_CARR2[2] = 0 ; Y component of the 1st offset vector (I1)
N50 $TC_CARR3[2] = -100.5 ; Z component of the 1st offset vector (I1)
N60 $TC_CARR4[2] = 0 ; X component of the 2nd offset vector (I2)
N70 $TC_CARR5[2] = 0. ; Y component of the 2nd offset vector (I2)
N80 $TC_CARR6[2] = 100.5 ; Z component of the 2nd offset vector (I2)
N90 $TC_CARR7[2] = 0 ; X component of the 1st rotary axis vector (V1)
N100 $TC_CARR8[2] = 1 ; Y component of the 1st rotary axis vector (V1)
N110 $TC_CARR9[2] = 0 ; Z component of the 1st rotary axis vector (V1)
N120 $TC_CARR10[2] = 0 ; X component of the 2nd rotary axis vector (V2)
N130 $TC_CARR11[2] = 0 ; Y component of the 2nd rotary axis vector (V2)
N140 $TC_CARR12[2] = -1 ; Z component of the 2nd rotary axis vector (V2)
N150 $TC_CARR13[2] = 0 ; Angle of rotation of the 1st axis
N160 $TC_CARR14[2] = 0 ; Angle of rotation of the 2ndaxis
N170 $TC_CARR15[2] = -300 ; X component of the 3rd offset vector (I3)
N180 $TC_CARR16[2] = -200 ; Y component of the 3rd offset vector (I3)
N190 $TC_CARR17[2] = -400 ; Z component of the 3rd offset vector (I3)
N200 $TC_CARR18[2] = 300 ; X component of the 4th offset vector (I4)
N210 $TC_CARR19[2] = 200 ; Y component of the 4th offset vector (I4)
N220 $TC_CARR20[2] = 400 ; Z component of the 4th offset vector (I4)
N230 $TC_CARR21[2] = B ; Designation of the 1st rotary axis
N240 $TC_CARR22[2] = C ; Designation of the 2nd rotary axis
N250 $TC_CARR23[2] = "M" ; Type of tool carrier

Notes

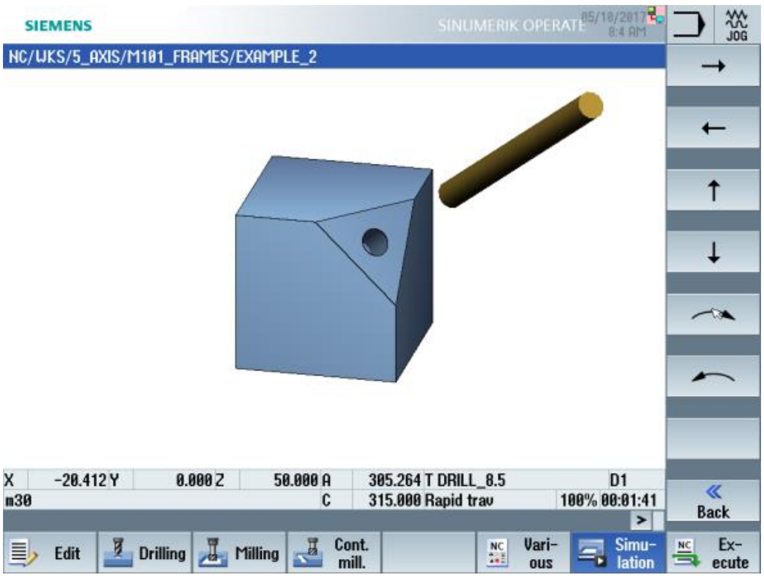
NC program:

```
T1 D1; FACING TOOL_D32
M6
G54 G17 S4000 M3
TRANS X50 Y50 Z-25 ; Translation WO reference point for rotation
AROTS X=-45 Y=45 ; Frame rotation with solid angles
AROT Z=30
TCOFR ; Calculation of the rotary axis positions in the active frame
TCARR=2 ; Selection of the orient. tool carrier
G0 X0 Y0 B=$P_TCANG[1]  C=$P_TCANG[2]  ; Pos. rotary axes
PAROT ; Activate frame rotation of the workpiece
G0 X0 Y0 Z25 ; Position 1st hole
CYCLE61(50,14.442,5,0,-25,0,32,25,2,60,0,500,41,0,1,10) ;Face milling cycle
PAROTOF ; Deactivate frame rotation of the workpiece
TCARR=0  ; Deselection of the orient. tool carrier

T2 D1; BOHRER_D8.5
M6
G54 G17 S3000 M3
TRANS X50 Y50 Z-25 ; Translation WO reference point for rotation
AROTS X=-45 Y=45 ; Frame rotation with solid angles
AROT Z=30
TCOFR ; Calculation of the rotary axis positions in the active frame
TCARR=2 ; Selection of the orient. tool carrier
G0 X0 Y0 B=$P_TCANG[1]  C=$P_TCANG[2]  ; Pos. rotary axes
PAROT ; Activate frame rotation of the workpiece
```

```
G0 X-20.412 Y0 Z10 ; Position 1st hole
CYCLE82(100,0,5,-15,,1,0,1,12) ; Drilling cycle
PAROTOF ; Deactivate frame rotation of the workpiece
TCARR=0  ; Deselection of the orient. tool carrier
M30
```

Simulation



Notes

Frame rotation of the tool with TOFRAME/TOROT

Explanation of the functions

TOFRAME:

With **TOFRAME**, you can define a frame whose Z axis points in the tool direction. Whereby, an existing programmed frame is overwritten by **TOFRAME**, which describes a pure rotation. Any offsets, mirroring or scaling in the previously active frame are deleted.

This behavior can sometimes have a negative effect. It is often useful to retain a work offset (WCS) that defines the reference point in the workpiece.

TOROT:

For this reason, the **TOROT** language command has been introduced which only overwrites the rotation component in the programmed frame and the other components remain unchanged.

The rotation defined with **TOROT** is the same as with **TOFRAME** and is dependent on the existence of an orientable tool carrier.

The **TOROT** command is particularly useful for 5-axis transformations because the programming with active orientable tool carrier remains the same for each kinematics type.

For these reasons, frames should only be programmed with **TOROT** in which the Z axis points in the tool direction. This definition is suitable for milling operations in which **G17** is active.

Particularly for milling operation or generally with active G18/G19, it is desirable that frames can be defined where the alignment is along the X or Y axis. The following G codes of group 53 have therefore been introduced:

TOROTX,
TOROTY,
TOROTZ.

Note:

The programming of TOROT is particularly helpful for retraction from an inclined
hole after a power failure with kinematics types T and M.

More detailed information can be found in Module M103 "5-axis transformation", "Information for machine operators"

Notes

Example: Tool frame rotation for kinematics types T and M

If the tool can be rotated by one or two rotary axes (kinematics types T and M), a tool frame can be specified by means of TOFRAME or the tool rotation programmed by means of **TOROT** (see Fig. 4.8 and 4.9).

The resulting tool frame created by the programmable tool rotation with TOROT, is only used for retraction from an inclined plane, e.g. drilling, with the kinematics types T and M.

Fig. 4.8: Kinematics type T

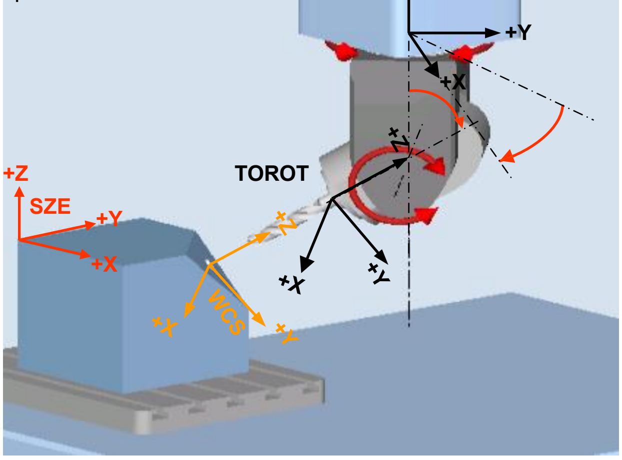
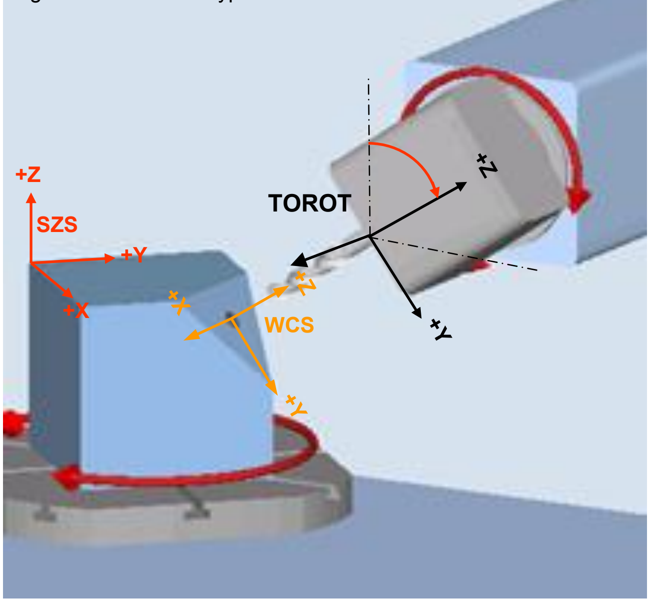


Fig. 4.9: Kinematics type M



Notes

Writing the rotation to the system frame

It is possible to write frames created with TOROT or TOFRAME to a separate system frame ($P_TOOLFR). For this purpose, bit 3 must be set in machine data ***MD28082 $MC_MM_SYSTEM_FRAME_MASK***.
The programmable frame is then retained without changes.

Differences occur when the programmable frame is edited further. It is recommended for the frames created from the G codes of group 53 that only the intended system frame ($P_TOOLFR) be used for new systems.

Example

TRANS is programmed after TOROT. TRANS without specification of parameters deletes the programmable frame. In the variant without system frame, the frame component of the programmable frame caused by TOROT is therefore also deleted. If the TOROT component is in the system frame, it is retained.

Deselection

TOROT and TOFRAME, etc. are switched off with the command **TOROTOF**. **TOROTOF** deletes the entire system frame $P_TOOLFR. If the commands TOFRAME and TOROT do not describe the system frame, but rather the programmable frame, then **TOROTOF** only deletes the rotation component and leaves the other frame components unchanged.

Description of the NC command

NC command	Description
TOFRAME	Definition of a frame rotation with tool alignment along the Z axis in the G17 plane. Previously active frames are deleted. Identical to **TOFRAMEZ**.
TOFRAMEX	Definition of a frame rotation with tool alignment along the X axis. Previously active frames are deleted.
TOFRMAEY	Definition of a frame rotation with tool alignment along the Y axis. Previously active frames are deleted.
TOFRAMEZ	Definition of a frame rotation with tool alignment along the Z axis. Previously active frames are deleted.
TOROT	Definition of a frame rotation with tool alignment along the Z axis in the G17 plane. Previously active frames are deleted. Identical to **TOROTZ.**
TOROTX	Definition of a frame rotation with tool alignment along the X axis. Previously active frames are deleted.
TOROTY	Definition of a frame rotation with tool alignment along the X axis. Previously active frames are deleted.
TOROTZ	Definition of a frame rotation with tool alignment along the Z axis. Previously active frames are deleted. Identical to **TOROT.**
TOROTOF	Deactivates TOROT and TOFRAME, etc. and can also delete the entire system frame $P_TOOLFR (see deactivation).

M101: END

Notes

M102
Sinumerik
Operate

Swivel CYCLE800
3+2 axis positioning

Module Description:

After completing this module you will be able to operate the swivel function and to program the swivel cycle CYCLE800. You will also be taught the fundamentals for the settings of the swivel cycle.

Module Objective:

After completing this module you will understand all aspects of CYCLE800.

Based on practical examples, you will be shown the operation, programming, application options and settings for the swivel function and CYCLE800.

Content:

Theory and fundamentals

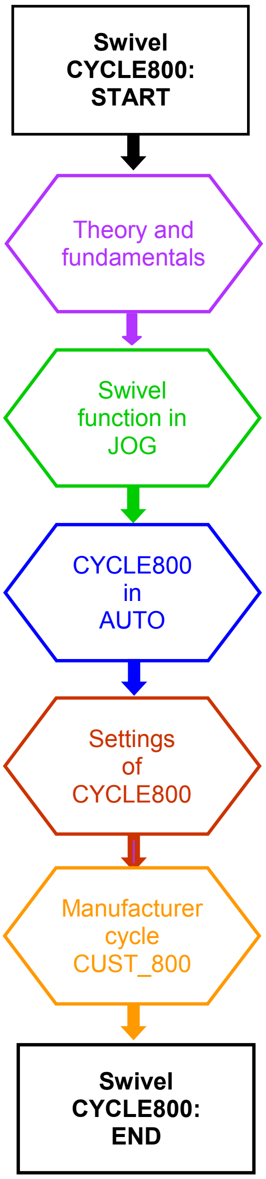
Swivel function in JOG

Settings CYCLE800

CYCLE800 in AUTO mode

Manufacturer cycle CUST_800

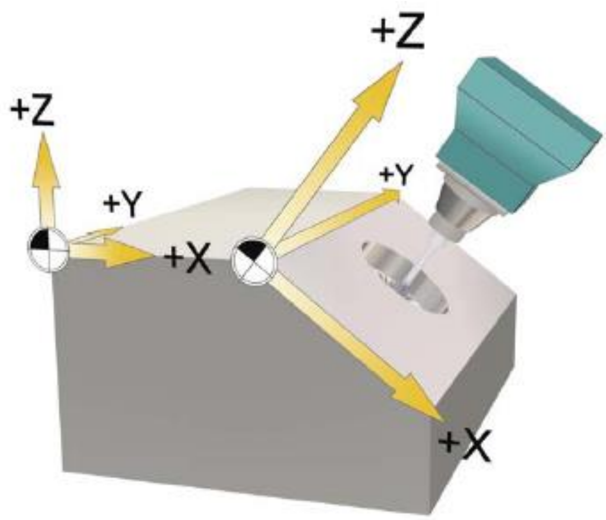
840D sl SINUMERIK Operate


Swivel CYCLE800: START
Theory and fundamentals
Swivel function in JOG
CYCLE800 in AUTO
Settings of CYCLE800
Manufacturer cycle CUST_800
Swivel CYCLE800: END

Notes

Theory and fundamentals

Introduction of swivel cycle CYCLE800



The Swivel **CYCLE800** is a static plane transformation, which allows you on a 3+2 axis machine (e.g. swivel head or swivel table) to define a rotated working plane in space.

In this work plane you can than program a 2D or 3D machining operation.
This is also referred to as **3+2 axis positioning**, by means of the 2 rotary axes involved in the transformation being only positioned while the linear axes move during machining.

By calling the appropriate NC functions, the cycle converts the active workpiece zero (e.g. G54) and tool offsets to refer to the rotated surface, taking into account the kinematics chain offsets on the machine, and positions the physical rotary axes (optional) normal to the programmed working plane.

You can program the rotary axes of the machine (**A,B,C**) or can simply specify the rotation around the geometry axes (**X,Y,Z**) of the workpiece coordinate system.

The rotation of the workpiece coordinate system in the program is then automatically converted into a rotation of the relevant rotary axes (physical axes) of the machine during machining. The previously set zero offset is here by automatically transformed into the new machining plane, and the traversing movements of the linear axes now relate to this machining plane.

The swivel axes are always rotated to place the machining plane perpendicular (normal) to the tool axis for machining. The machining plane remains fixed during machining.

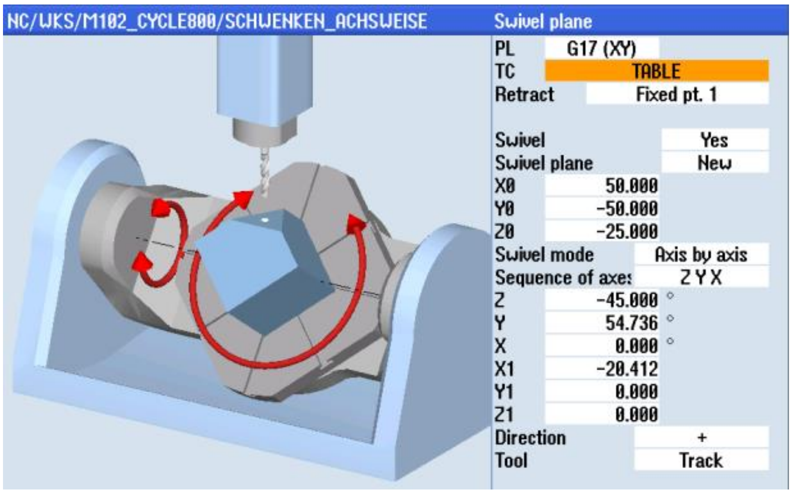
Notes

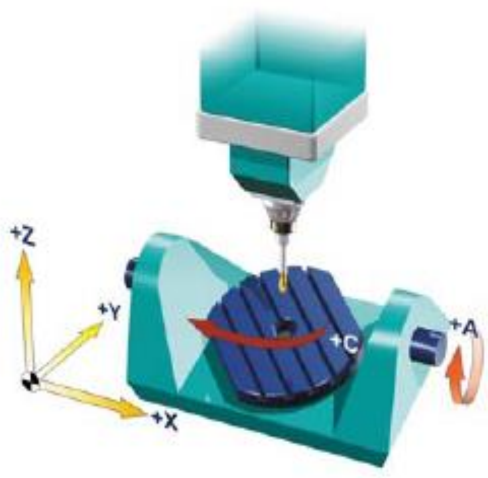
Implementation of CYCLE800 into SINUMERIK Operate

The following figures show the integration of the swivel cycle "**CYCLE800**" in the SINUMERIK Operate system for the currently supported 5-axis machine kinematics.

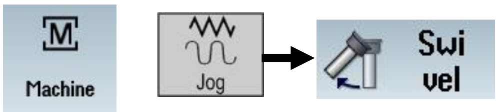
Example of a Swivel table kinematic type P (part)

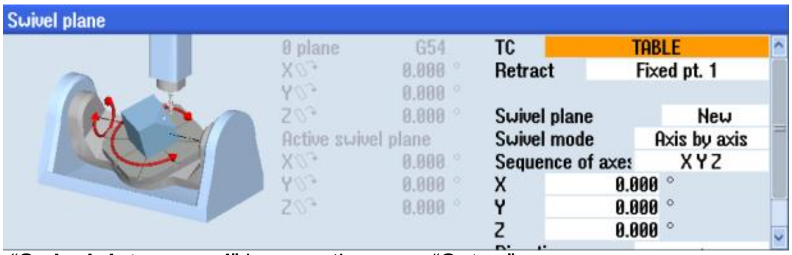
CYCLE800 “**Swivel plane**” in operating area “**Program**” programGUIDE G code



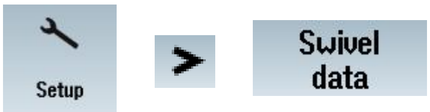


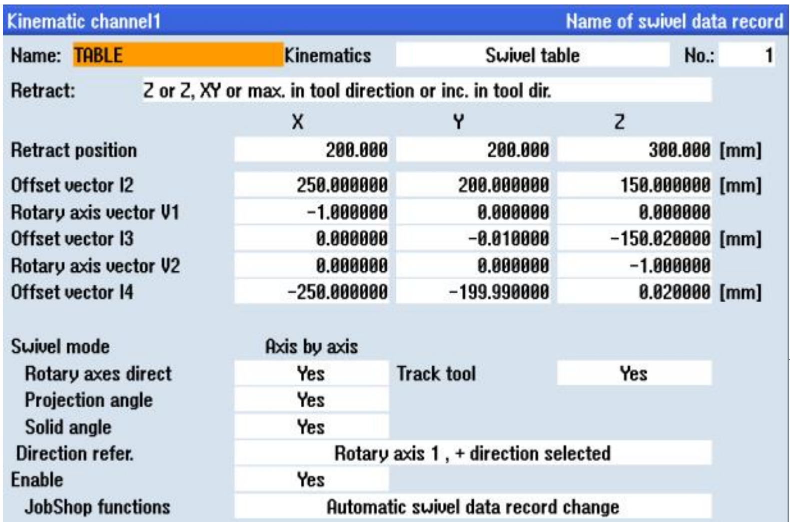
Function “**Swivel plane**” in “**JOG**“ mode





“**Swivel data record**” in operating area “**Setup**”





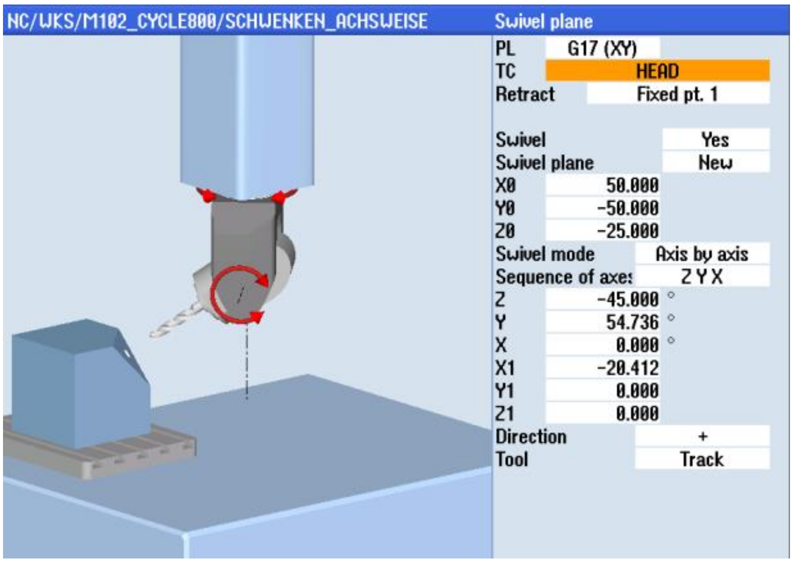
The kinematic of the machine is defined through offset vectors and rotary axis vectors in the **swivel data record**.

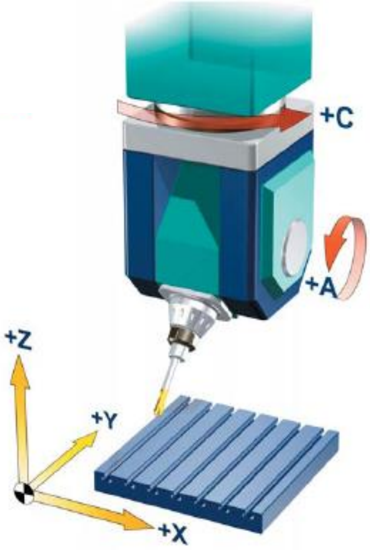
For this swivel rotary table kinematic the rotary axis vectors are defined as follows:
1. Rotary axis A (rotation about X)
2. Rotary axis C (rotation about Z)

Notes

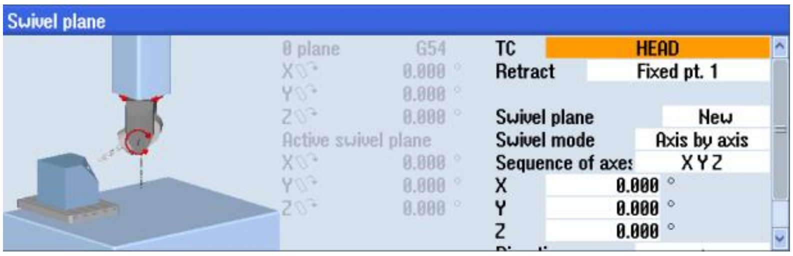
Example of a rotary fork head kinematik type T (tool)

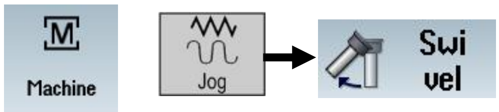
CYCLE800 “**Swivel plane**” in operating area “**Program**” programGUIDE G code



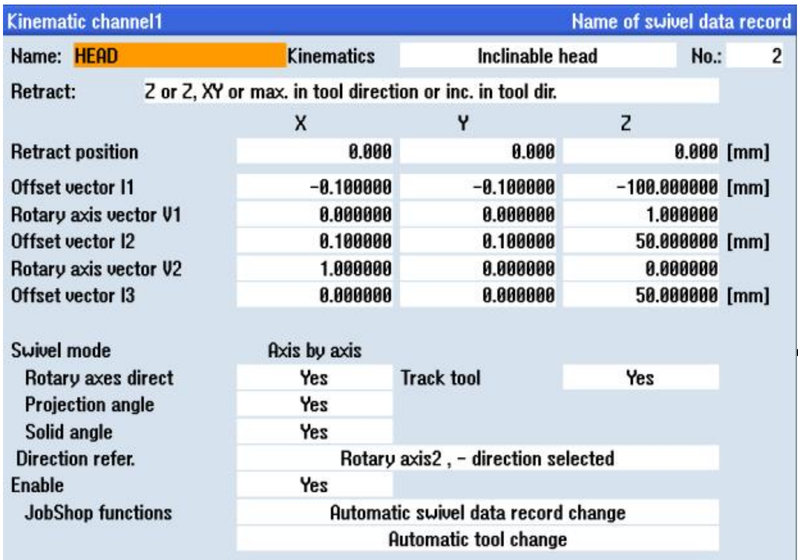


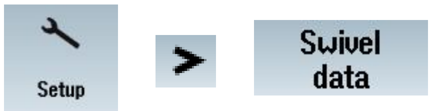
Function “**Swivel plane**” in „**JOG**“ mode





“**Swivel data record**” in operating area “**Setup**”





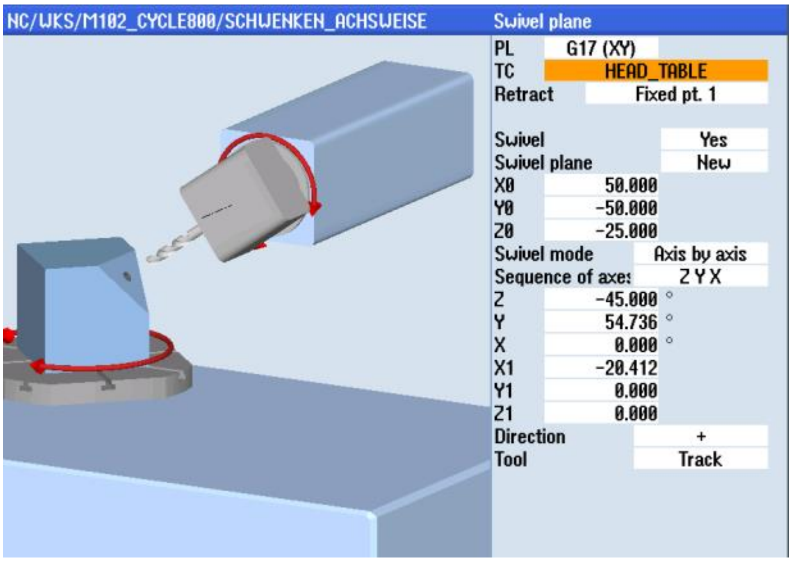
The kinematic of the machine is defined through offset vectors and rotary axis vectors in the **swivel data record**.

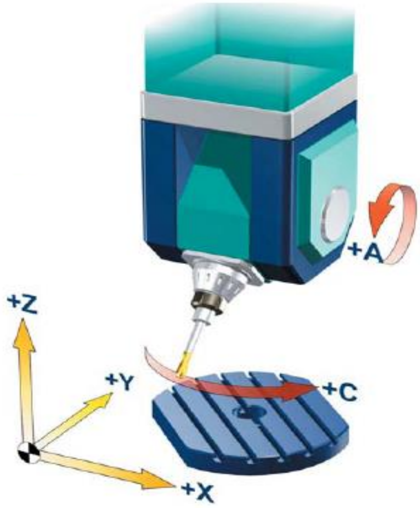
For this rotary fork head kinematic the rotary axis vectors are defined as follows:
1. Rotary axis C (rotation about Z)
2. Rotary axis A (rotation about X)

Notes

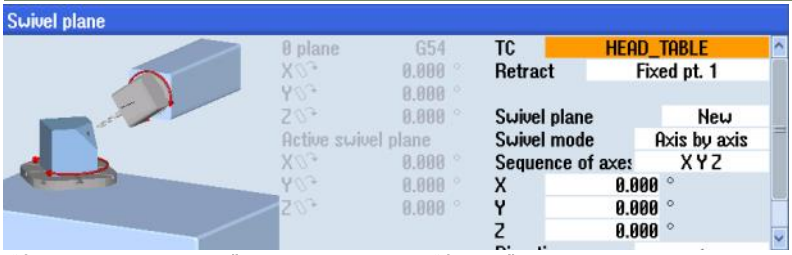
Example of a mixed rotary fork head kinematik type M (mixed)

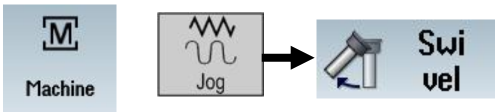
CYCLE800 “**Swivel plane**” in operating area “**Program**” programGUIDE G code



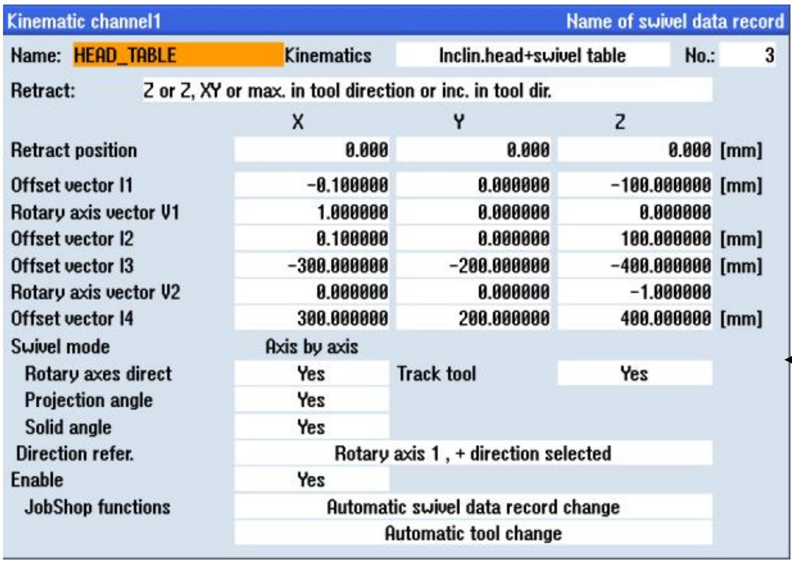


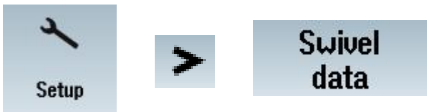
Function “**Swivel plane**” in „**JOG**“ mode





“**Swivel data record**” in operating area “**Setup**”





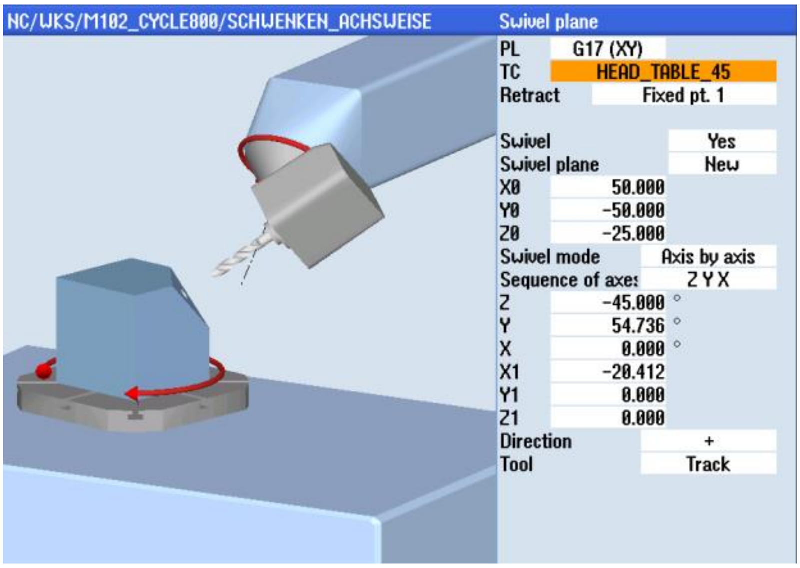
The kinematic of the machine is defined through offset vectors and rotary axis vectors in the **swivel data record**.

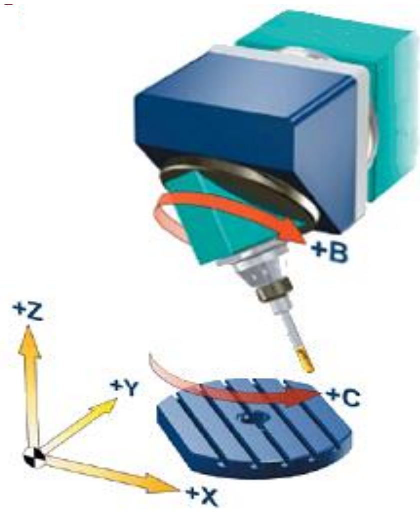
The rotary axes vectors for a mixed rotary head kinematic are defined as follows:
1. Rotary axis A (rotation about X)
2. Rotary axis C (rotation about Z)

Notes

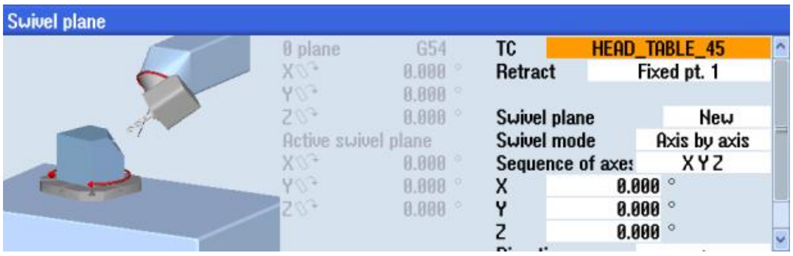
Example of a mixed nutaded head 45° kinematik type M (mixed)

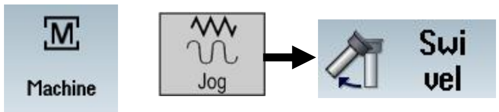
CYCLE800 “**Schwenken Ebene**” im Menu “**Programm**” programGUIDE G-Code



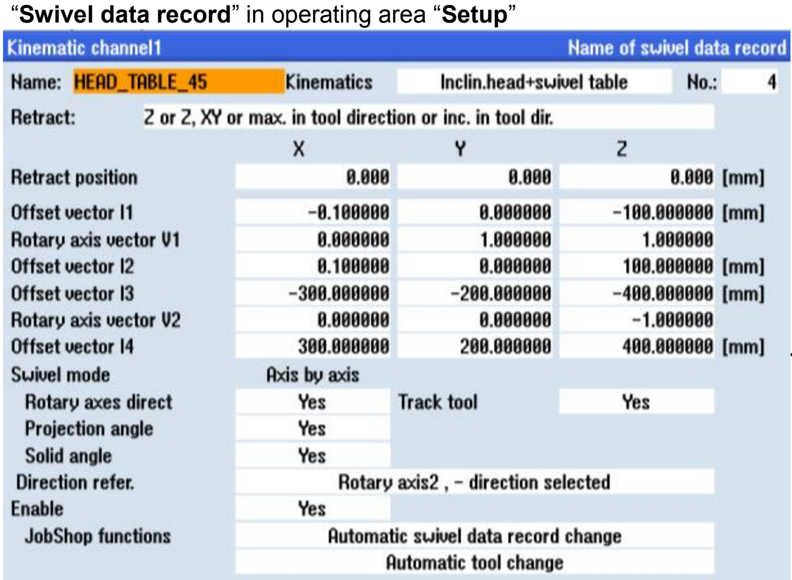


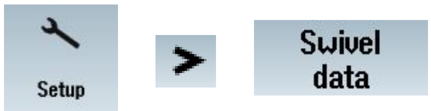
Function “**Swivel plane**” in „**JOG**“ mode





“**Swivel data record**” in operating area “**Setup**”





The kinematic of the machine is defined through offset vectors and rotary axis vectors in the **swivel data record**.

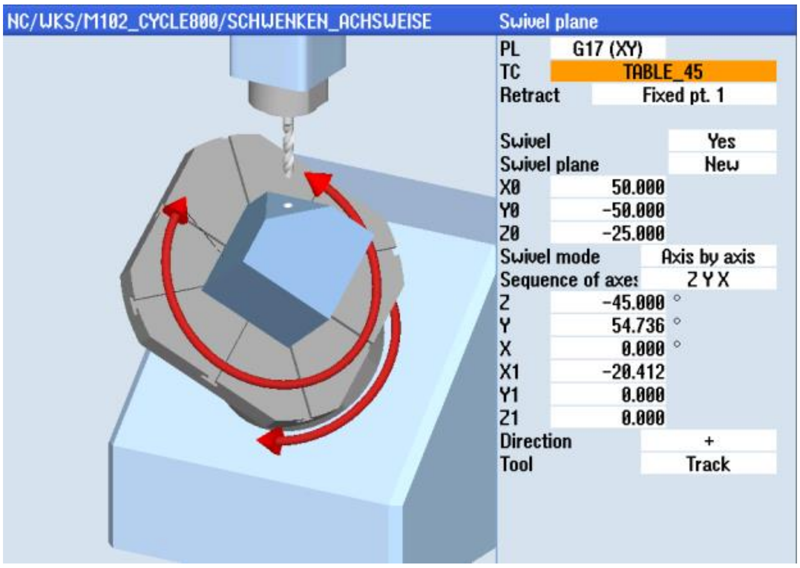
The rotary axes vectors for this mixed nutaded rotary head kinematic are defined as follows:

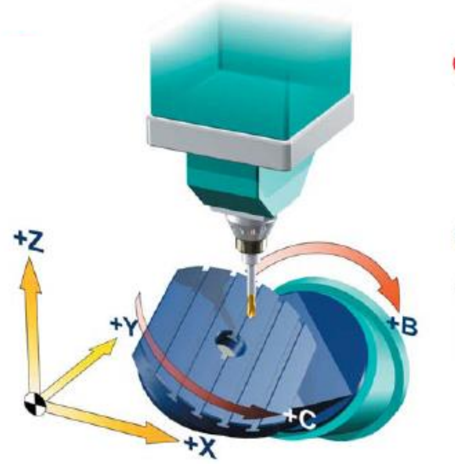
1. Rotary axis B (rotation between Y/Z at 45°)
2. Rotary axis C (rotation about Z)

Notes

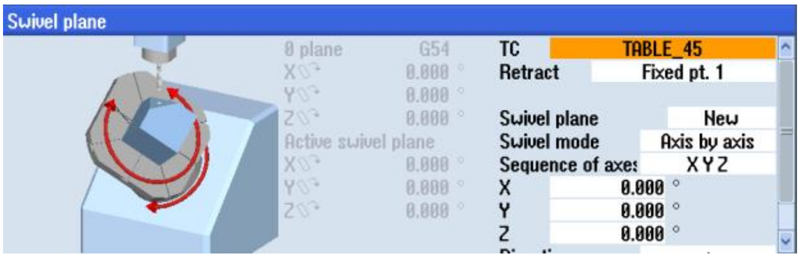
Example of a nutated swivel rotary table 45° kinematic type P (part)

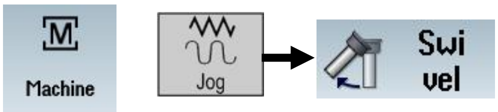
CYCLE800 “**Swivel plane**” in operating area “**Program**” programGUIDE G code



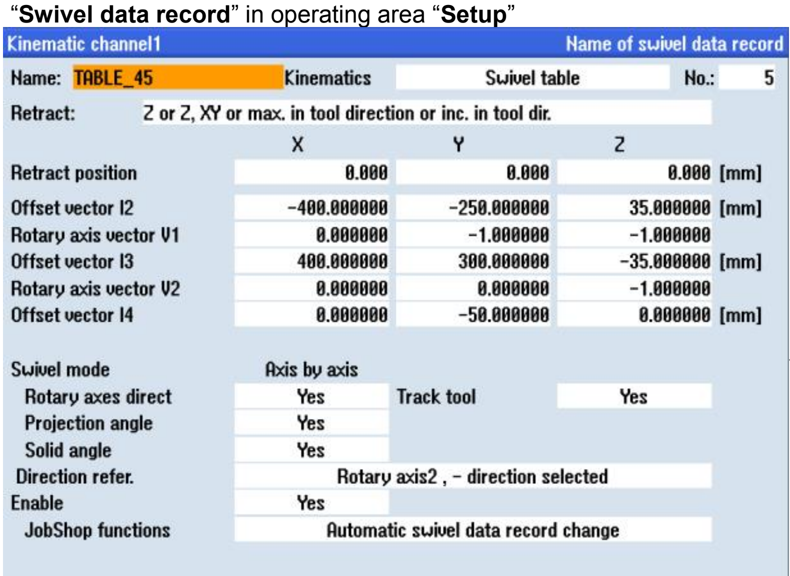


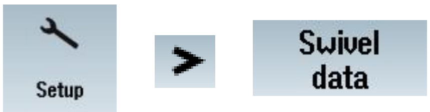
Function “**Swivel plane**” in „**JOG**“ mode





“**Swivel data record**” in operating area “**Setup**”





The kinematic of the machine is defined through offset vectors and rotary axis vectors in the **swivel data record**.

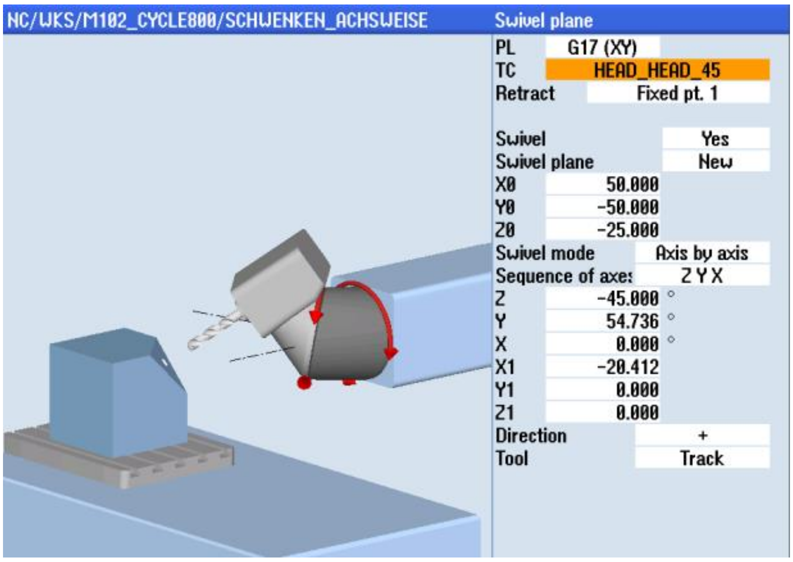
The rotary axes vectors for this nutaded swivel rotary table kinematic are defined as follows:

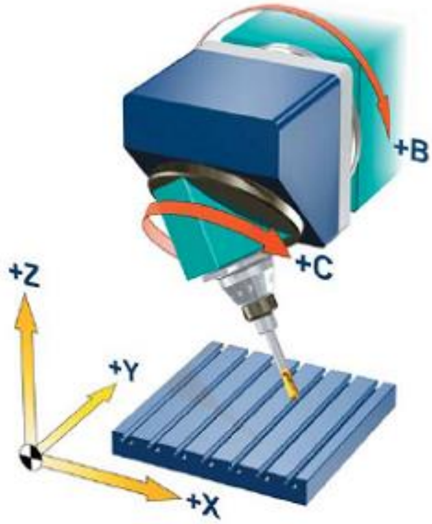
1. Rotary axis B (rotation between Y/Z at 45°)
2. Rotary axis C (rotation about Z)

Notes

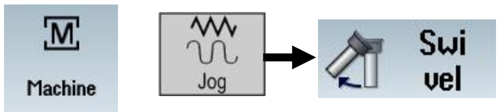
Example of a nutated swivel rotary head 45° kinematic type T (tool)

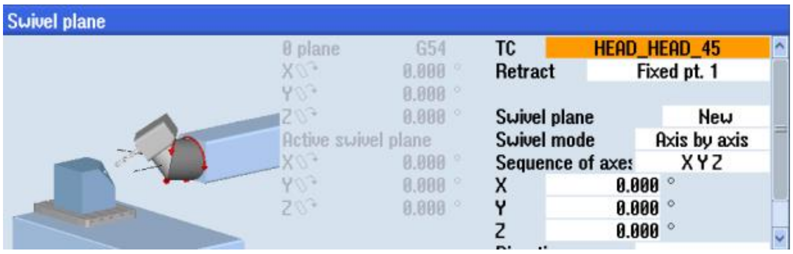
CYCLE800 “**Swivel plane**” in operating area “**Program**” programGUIDE G code



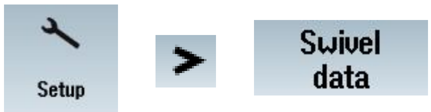


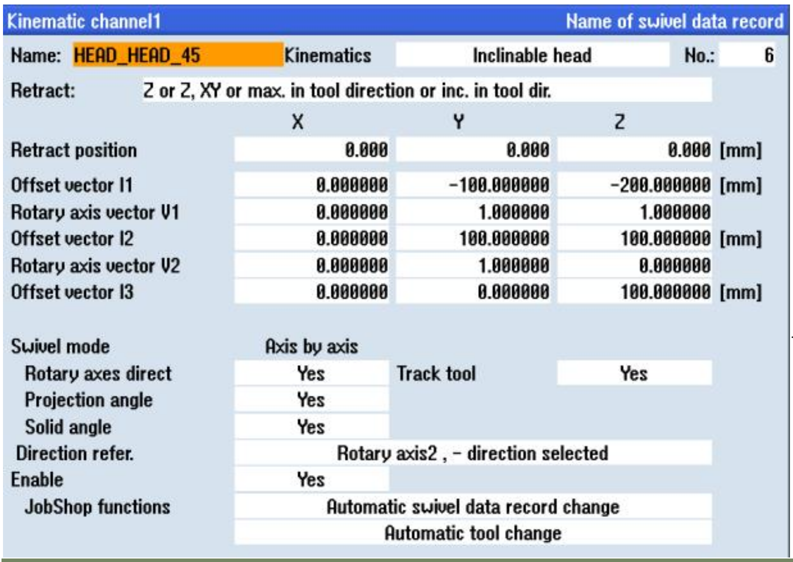
Function “**Swivel plane**” in „**JOG**“ mode





“**Swivel data record**” in operating area “**Setup**”





The kinematic of the machine is defined through offset vectors and rotary axis vectors in the **swivel data record**.

The rotary axes vectors for this nutaded rotary head kinematic are defined as follows:

1. Rotary axis B (rotation about Y)
2. Rotary axis C (rotation between Y/Z at 45°)

Notes

Application field of CYCLE800

CYCLE800 is availabel in two variations for the technology milling:

- CYCLE800 “Swivel plane” and
- CYCLE800 “Approach tool”

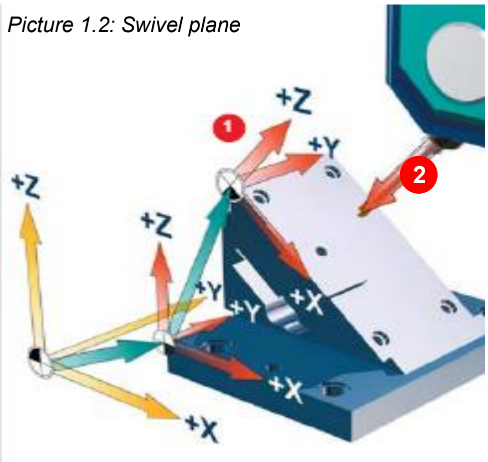
CYCLE800 “Swivel plane”

The feature “**Swivel plane**” is mainly used for milling and drilling in inclined planes in the 2D application range, in which the linear axes XYZ are able to move and the rotary axes only position to define a swiveled machining plane (3+2 axis positioning).

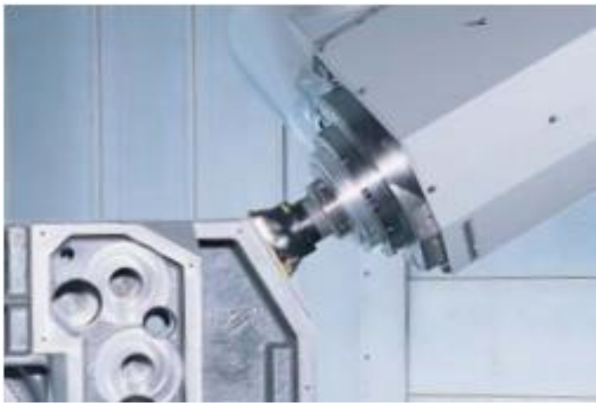
Once the coordinate system (1) is shifted and rotated to align with the inclined surface you can simply program for example a hole position with the call of a drilling cycle (see picture 1.2).

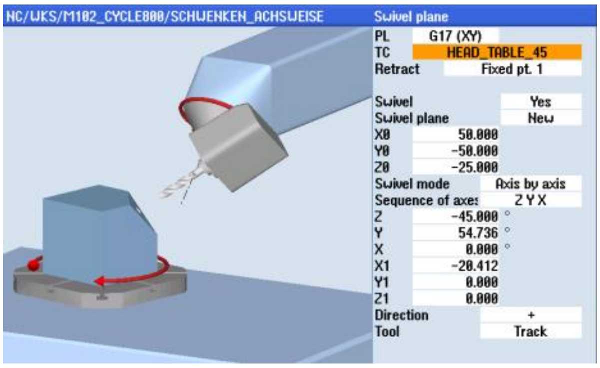
The use of drilling and milling cycles with a change of tool orientation is only possible with plane transformation in CYCLE800. Here by the tool axis (2) is always oriented perpendicular (normal)
to the inclined machining plane (1) (see picture 1.3).

Picture 1.2: Swivel plane



Picture 1.3: Machining in a inclined plane





Notes

CYCLE800 "Approach tool"

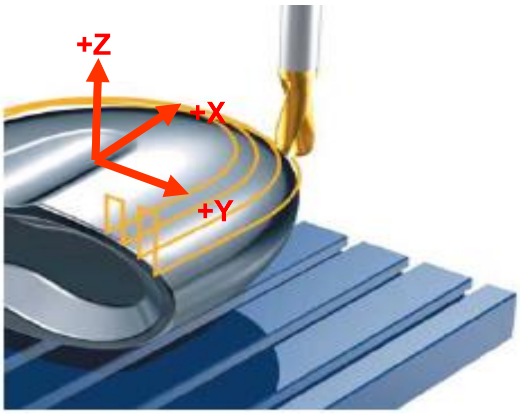
The cycle "**Approach tool**" is primarily used for 3D mould & surface machining and can be applied on a 3+2 axis machine. Machining takes place with change in tool orientation, whereby the linear axes XYZ are able to move and the rotary axes are only positioned to cause a change in tool orientation and remain static during machining.

In picture 1.4 the cutting conditions deteriorate as the cutter moves up towards the top (surface feed towards zero) or to the side of the work-piece (cutter shank pushes against part).

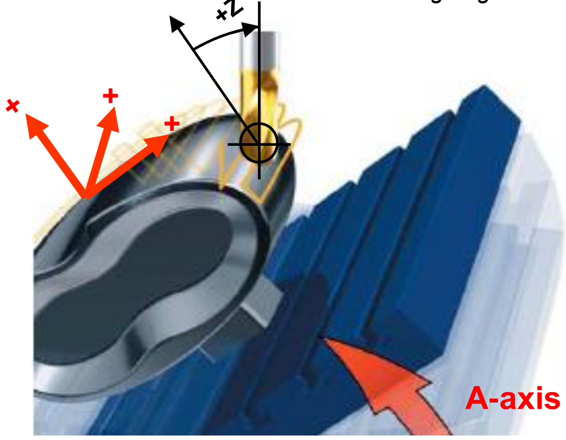
In picture 1.5 the cutter is working with optimum cutting conditions through a change in tool orientation. In order to machine a free-form surface completely, it is sometimes necessary to change tool orientation several times.

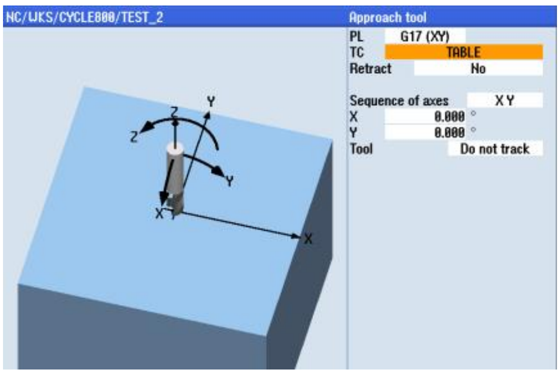
With CYCLE800 "Approach tool" the workpiece coordinate system is not rotated, but only the tool orientation is changed. The machining plane remains normal to the WCS (see picture 1.5).

Picture1.4: Without tool setting angle



Picture 1.5: With tool setting angle

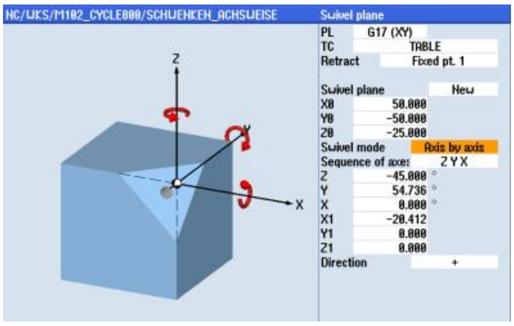




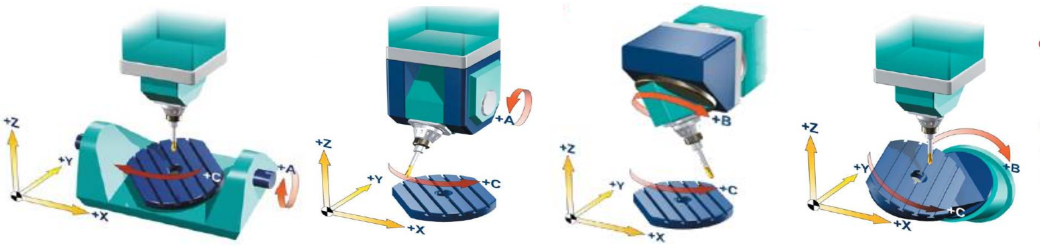
Notes

Programming advantages with CYCLE800

- Quick programming of machining operations in inclined planes, in a workpiece oriented coordinate system. Calculation of rotary axes positions are not required and the workpiece reference is maintained in the swiveled environment.



- Kinematic independent programming if swivel mode is “axis by axis”. This means the program runs on any kinematic regardless of the type.



- No kinematic specific postprocessor required for programming on a CAM-system.

- Tool- and work offsets can be set and modified at any time on the machine in the parameters without modifying the NC-program. (Basic rotations and rotary axes offsets for workpiece alignment when setting up the work piece are taken into consideration in the NC-program during main run).

- Milling and drilling cycles as well as measuring cycles can be used, due to alignment of the tool with the machining surface (tool axis is parallel to Z-axis G17 plane).

- Automatic retract before swiveling with consideration of software limits. Various retraction strategies are hereby available.

- The swivel frame is being retained after NC reset or power failure. This allows tool retraction in inclined planes (swivel frame is stored to the static NC memory for rotary axis reference, work piece reference and tool reference)

Notes

<u>Detailed explanation of swivel cycle “CYCLE800”</u>

How to program the swivel cycle “CYCLE800” ?

With the swivel cycle “**CYCLE800**” it is possible to transform a coordinate system (WCS) with the help of frames, which allows machining of inclined planes under activation of an orientable tool carrier “**TCARR**”. The transformation of the active workpiece zero offset (WO) is stored in a so called “**Swivel frame**” and is composed by the following frame chain:

—Rotary table reference ($P_PARTFRAME) for type P and M

—Tool reference ($P_TOOLFRAME) for type T and M

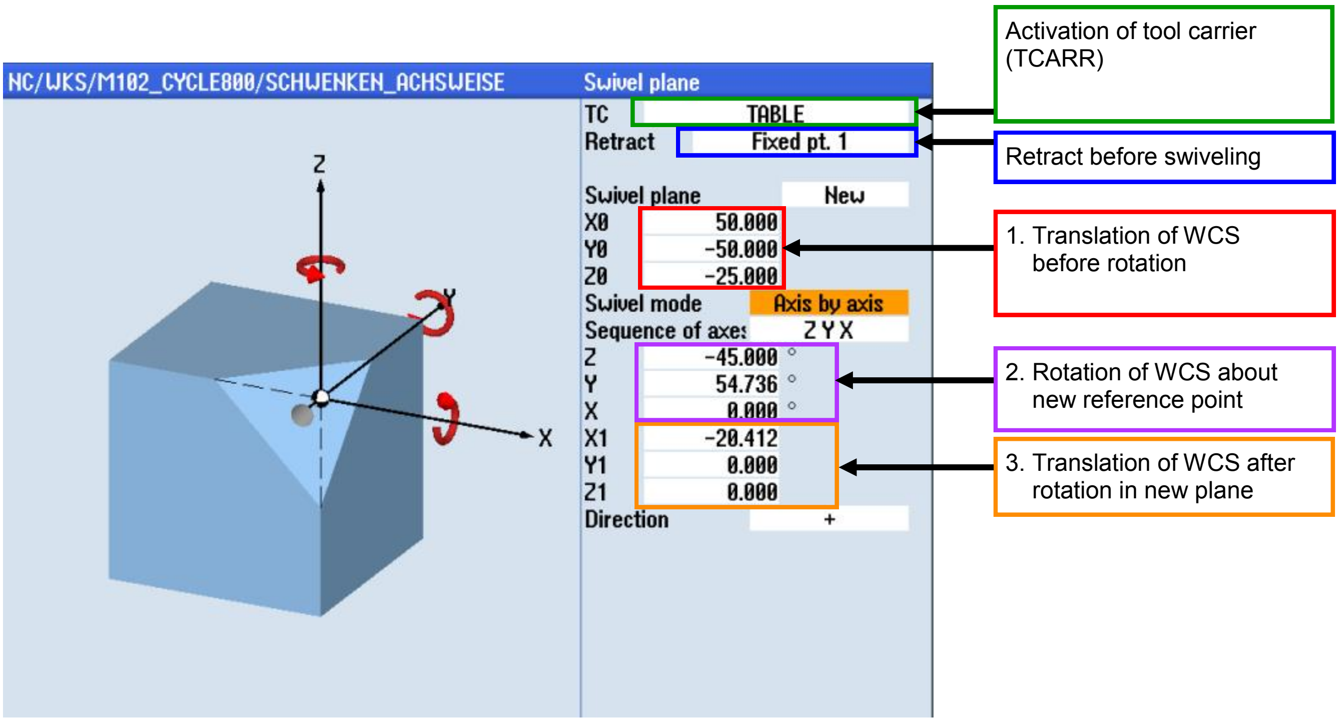
—Workpiece reference ($P_WPFRAME) for type P,M and T

With the aid of frames it is possible to translate and rotate coordinate systems in such a way, that the workpiece coordinate system is aligned with the inclined plane of the workpiece. This allows simplification and reduction of programming time.

Swiveling into an inclined machining plane basically requires the following steps:

1. Translation of the WCS before rotation
2. Rotation of the WCS about the new reference point
3. Translation of the WCS after rotation in the new swivel plane

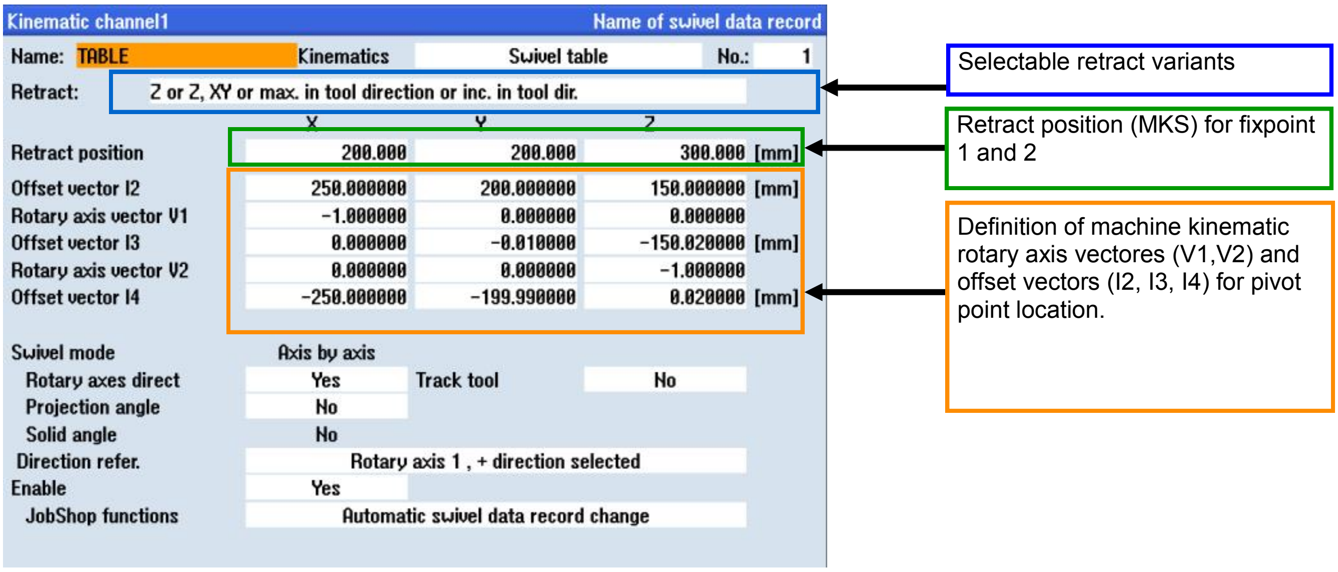
Steps 1 to 3 determine the "**Swivel frame**" or the new workpiece coordinate system and therefore the reference for the machining program.

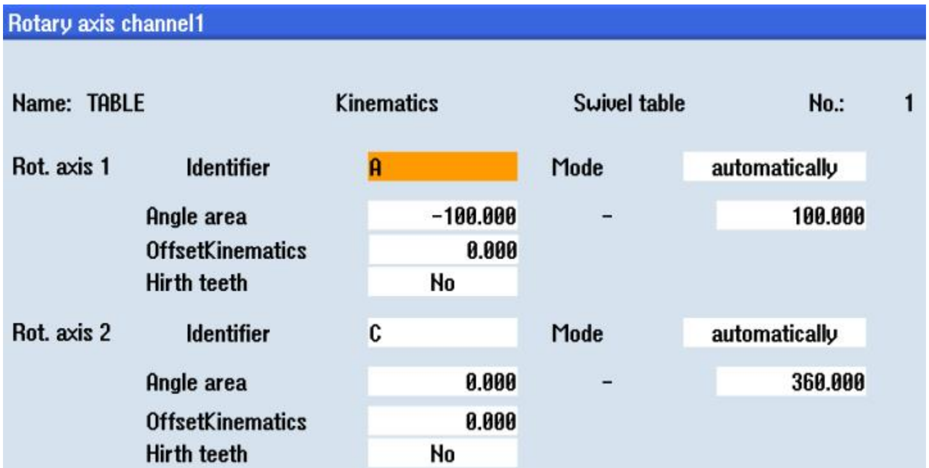


Notes

Swivle data record (orientable tool carrier TCARR)

The kinematic of the machine with the array of the rotary axes and their pivot point is defined in the swivel data record. The set values for offset vectors and rotary axes vectors are used for calculation of the workpiece location after swiveling in reference to the programmed swivel frame. Additional functions for the swivel cycle can here be set and enabled upon requirement.

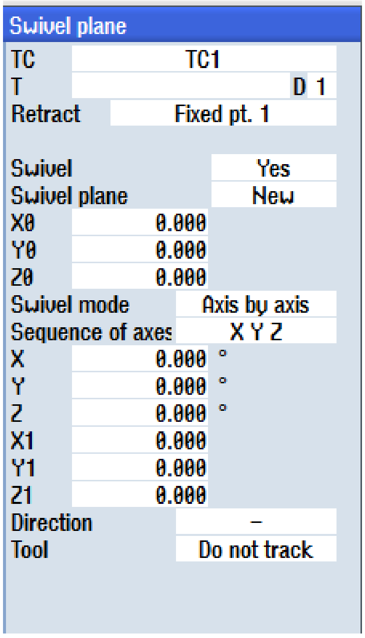




Note:

A detailed explanation of all tool carrier variables and their settings can be found on page 83
in this module.

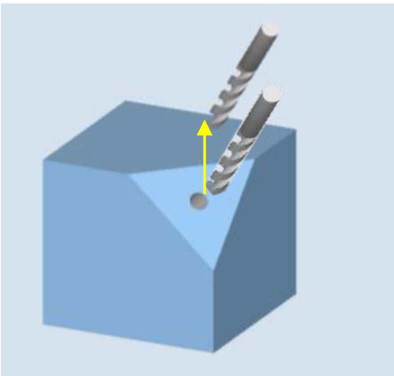
Notes



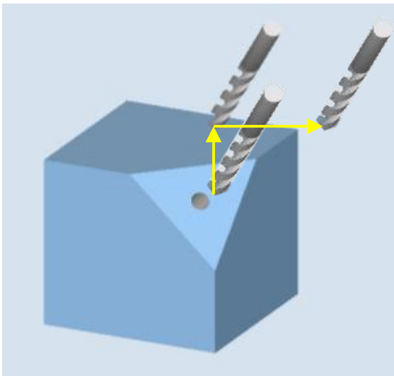
Swivel plane	
TC	TC1
T	D 1
Retract	Fixed pt. 1
Swivel	Yes
Swivel plane	New
X0	0.000
Y0	0.000
Z0	0.000
Swivel mode	Axis by axis
Sequence of axes	X Y Z
X	0.000 °
Y	0.000 °
Z	0.000 °
X1	0.000
Y1	0.000
Z1	0.000
Direction	–
Tool	Do not track

Retract

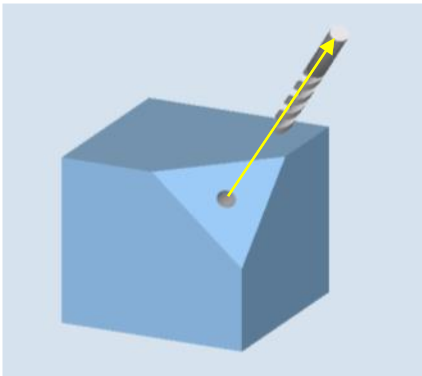
Here it is possible to retract the tool from the workpiece before swiveling the rotary axes into the new machining plane in order to avoid collision with the workpiece. The retraction variants which are displayed in the input mask of the swivel cycle can be enabled in the swivel data record.



Retraction to "**Fixed pt. 1**"
The retract position in Z is specified in the swivel data record in relation to the MCS. Retraction is only along the Z axis.



Retraction to "**Fixed pt. 2**"
The retract position in X,Y,Z is specified in the swivel data record in relation to the MCS. Retraction is first in Z then in X and Y.



Retraction "**Tool direction max.**"
Retraction is in the tool direction in relation to the MCS maximum to the software limit switch. For kinematics types T and M, several axes move simultaneously.

Note:

A detailed description of the retraction variants can be found in Section 4 in this module.

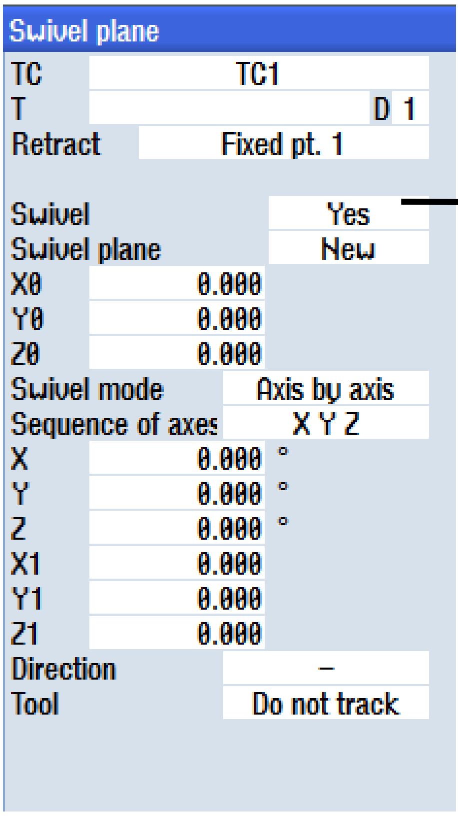
Notes

Retract:	**Retract before swivelling** ■ **No:** No retraction before swivelling ■ **Z:** Retraction of machine axis Z before swivelling. (Refers to the retract position Z only in the MCS defined in the swivel data record ■ **Z,X,Y:** Move machining axis to retraction position before swivelling (Refers to the retract position in the MCS defined in the swivel data record ■ **Max. tool direction:** Maximum retraction in tool direction (up to software end position) ■ **Inc. tool direction:** Retraction incremental in tool direction. ■ **ZR (for G17):** Retraction path (this field is only displayed for incremental retraction in tool direction). The display text for the retract positions in the cycle input mask can be modified in the channel setting data ***SD55221 $SCS_FUNKTION_MASK_SWIVEL_SET*** with setting of ***Bit 1***. =1: Display text "Z" = "**Fixed pt. 1**", Display text "Z, XY" = "**Fixed pt 2**". In case of modification of the retract variants for "Z" or "Z, XY" in the manufacturer cycle CUST_800.SPF the neutral text "**Fixed pt. 1**" and "**Fixed pt 2**" are displayed instead. =0: Display text "Z" = "**Z**", Display text "Z, XY" = "**Z,XY**" The Default for the parameter **"Retract"** with use of the softkey **"Initial setting"** can be set in channel ***SD 55420 $SCS_MILL_SWIVEL_RESET_RETRACT***. The following settings are possible:
	0 = no change 1 = no 2 = Z or Fixed pt 1 3 = Z, XY or Fixed pt 2 4 = Tool direction max. 5 = Tool direction inc.

Notes for Retraction:

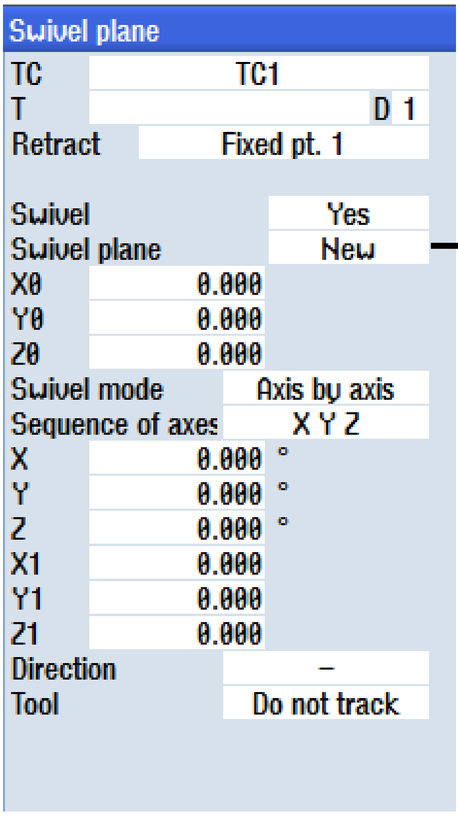
- When retracting in the tool direction, in the swivelled machine state, several axes can move (only for type T and M).
- The Retraction variant and the retract position in can be set in the Setup area of the swivel data record.
- The retraction positions are traversed absolutely. If another sequence or incremental positioning is desired, this can be modified during installation and set up in the CUST_800 user cycle.
- When programming with standard cycles and high settings for the retraction plane and swivel angles (swivelling through 90 degrees for multi sided machining), it is possible that the traversing area of the machine may be too small (software end position error), since the order of approach is always the machining plane (for G17 X, Y) first, followed by the infeed axis (Z). The retraction plane can be reduced to optimize this behavior.

Notes



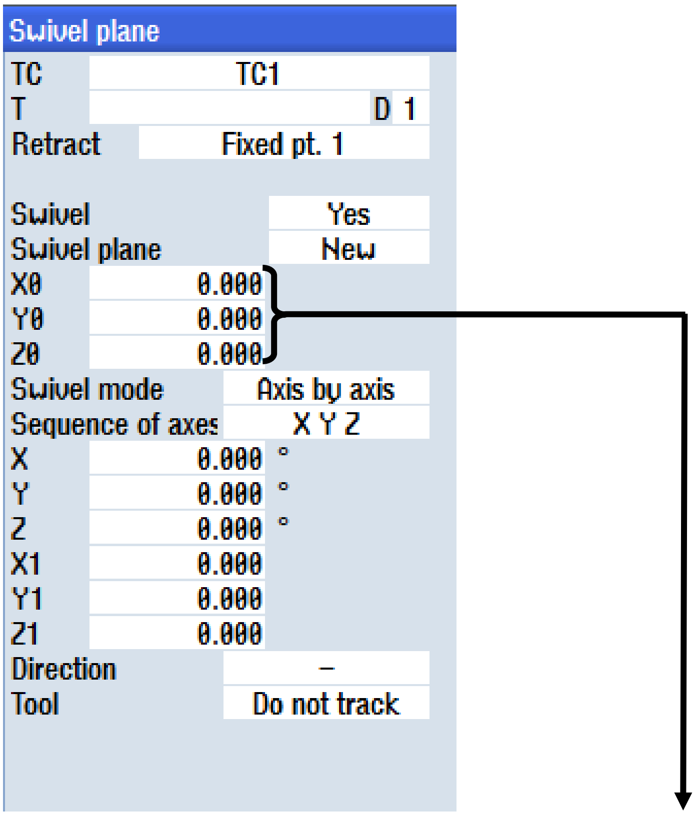
Swivel:	• **Swivel yes:** Rotary axes are positioned and manual rotary axes can be rotated in by the operator (see CUST_800.SPF). • **Swivel no** (only calculation): If the rotary axes are not to be traversed after activating the swivel cycle, then the "Swivel no" selection applies. Application: Creation of an auxiliary swivel plane according to the workpiece drawing. The "**Swivel no**" selection can be enabled in the channel-specific setting data ***SD 55221 $SCS_FUNKTION_MASK_SWIVEL_SET*** with ***bit 0 = 1*** in the cycle input screen.

Notes

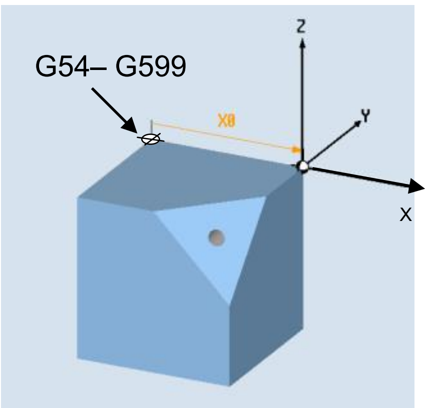


Swivel plane:	• **New:** Previous swivel frames and programmed frames are deleted and the values defined in the input screen form the new swivel frame. Each main program must start again with the swivel plane with a swivel cycle in order to ensure that no swivel frame from another program is active. • **Additive:** The swivel frame is added (incremental) to the swivel frame of the last swivel cycle. If several swivel cycles are programmed in a program and between these programmable frames are also active (e.g. AROT, ATRANS), they are taken into account in the swivel frame. If the currently active WO contains rotations, e.g. from a previously programmed machining of the workpiece, they are also taken into account in the swivel cycle.

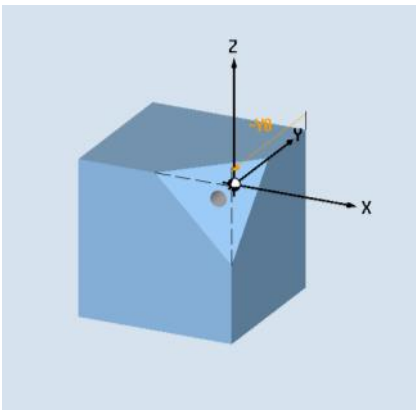
Notes



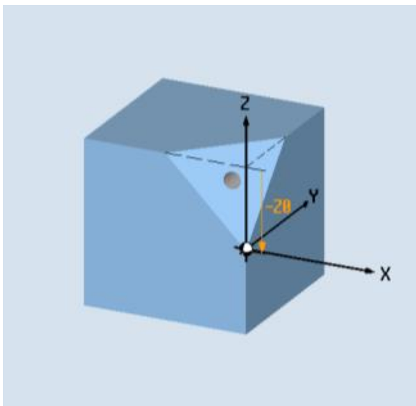
1. Translation of the WCS before rotation



Workpiece zero shift along the X axis.

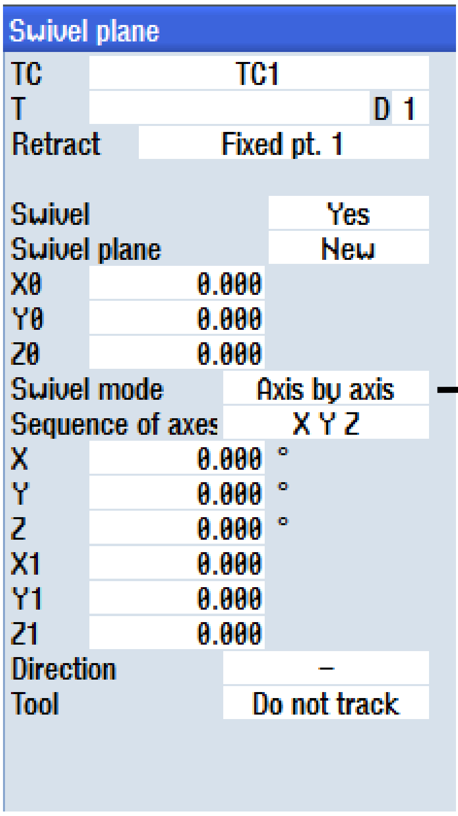


Workpiece zero shift along the Y axis.



Workpiece zero shift along the Z axis.

Notes



The swivel mode of the axes is specified with this parameter.

- Axis-by-axis (recommended "default")
- Projection angle
- Solid angle
- Rotary axes directly

The swivel mode always refers to the workpiece coordinate system and is therefore independent of the machine (exception "directly").

***Swivel mode* "*axis by axis*"**

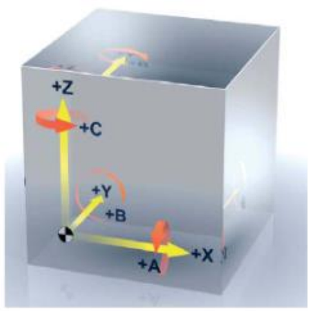
Here a selection between various swivel modes can be made and enabled in the swivel data record. Swivelling the coordinate system about the geometry axes "**axis by axis**" is recommended since the programming is independent of the machine kinematic. The swivel frame is hereby translated onto the participating rotary axes of the 5-axis machine.

Sequence of axes:

The sequence of the individual coordinate rotations is freely selectable with the "**SELECT**" key.
If more than one rotation is required to define an oblique plane, then the order of rotation as of the **RPY** principle is recommended.

The following rule applies for the G17 work plane:
1st rotation about the Z-axis
2nd rotation about the new Y-axis
3rd rotation about the new X-axis

The 3rd rotation is omitted in this example



Notes

Swivel mode:

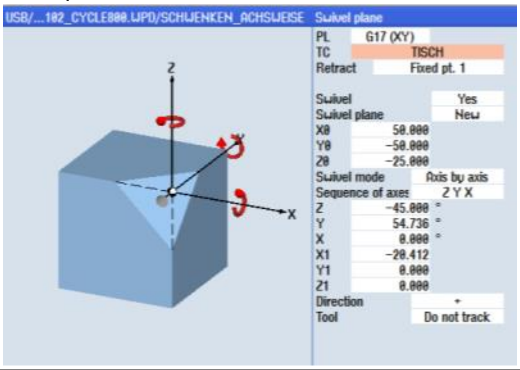
This parameter defines the swivel mode for the axes.

- Axis by axis (default)
- Projection angle
- Solid angle
- Directly

Swivel mode always refers to the coordinate system of the workpiece and does not, therefore, depend on the machine kinematic (exception “directly”).

Axis by axis

- With swivel mode “Axis by axis”, the tool is rotated about the individual axes of the coordinate system in succession, with each rotation starting from the previous rotation. The axis sequence is selectable.



Sequence of axes:

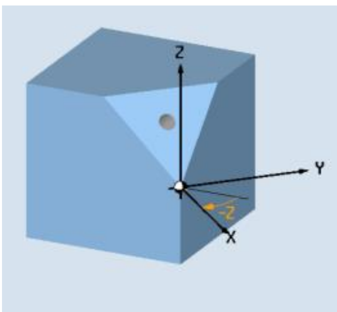
Sequence of axes around which rotation is executed
XYZ , XZY, YXZ, YZX, ZXY, **ZYX**,

The Input order is freely selectable with the “**Select**” key

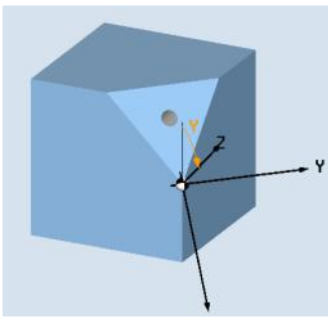
Z, Y, X:

Help screens (animation)

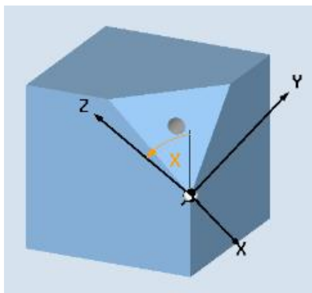
1. Rotation about Z-axis



2. Rotation about new Y-axis



3. Rotation about new X-axis

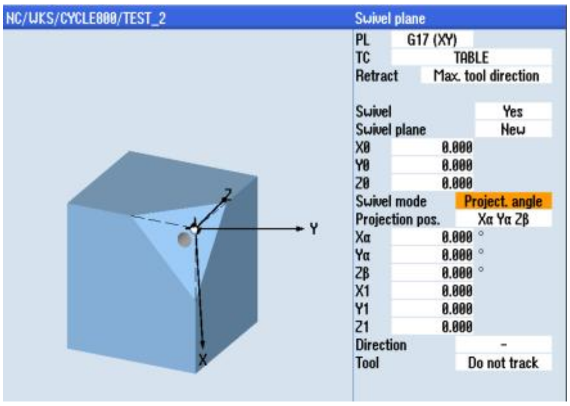


Notes

Swivel mode:

Projection angle

- With swivel mode “projection angle”, the angle value of the swivelled surface is projected onto the first two axes of the coordinate system. The user can select the axis rotation sequence. The third rotation starts from the previous rotation. The active plane and the tool orientation must be taken into consideration when the projection angle is applied:



– G17 (XY) projection angle Zβ, 3rd rotation around Z
– G18 (ZX) projection angle Yα, 3rd rotation around Y
– G19 (YZ) projection angle Xα, 3rd rotation around X

- When projection angles around XY and YX are programmed, the new X-axis of the swiveled coordinate system lies in the old ZX plane.

Swivel mode:

- When projection angles around XZ and ZX are programmed, the new Z-axis of the swiveled coordinate system lies in the old YZ plane.

- When projection angles around YZ and ZY are programmed, the new Y-axis of the swiveled coordinate system lies in the old XY plane.

Projection position:

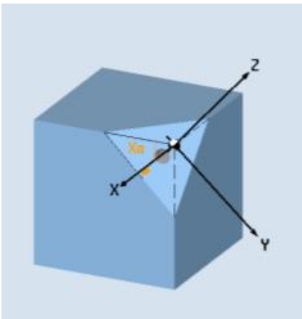
Position of projection in space
Xα Yα Zβ, Yα Zα Zβ, Zα Xα Zβ,

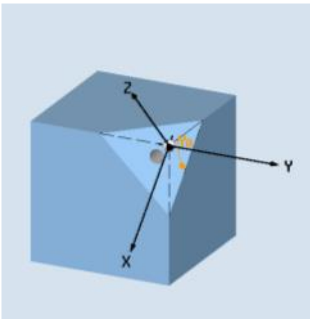
The Input order is freely selectable with the “**Select**” key

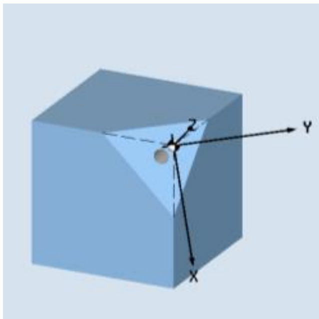
Xα Yα Zβ:

Help screens (animation)

Xα (projection angle) **Yα** (projection angle) **Zβ** (rotation about Z)





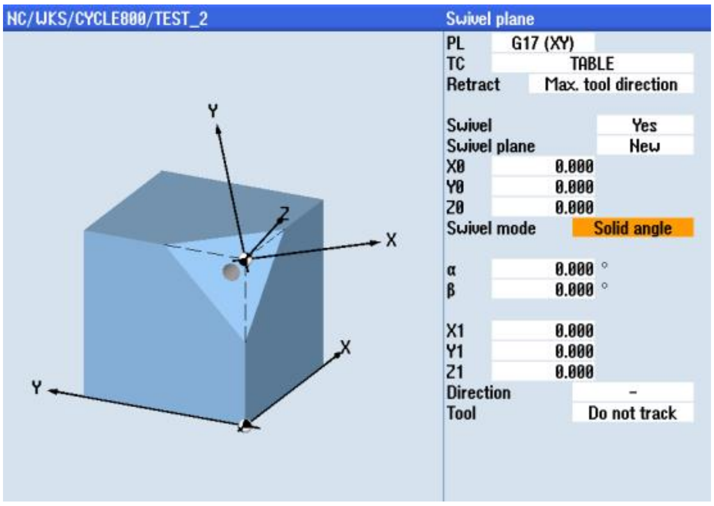


Notes

Swivel mode:

Solid angle

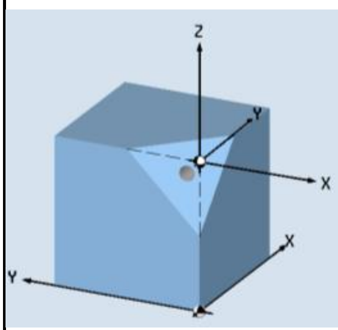
- With swivel mode “Solid angle”, the tool is first rotated about the Z axis (α angle) and then about the Y axis (β angle). The second rotation starts from the first. The positive direction of each rotation is shown in the following picture.



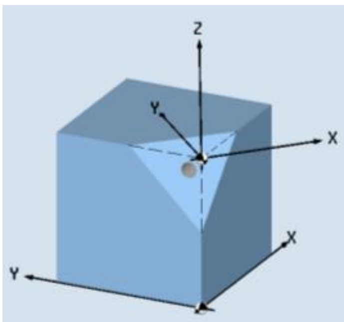
α, β:

Help screens solid angles (animation)

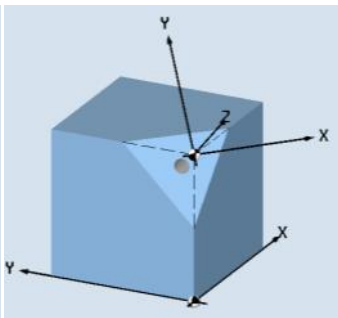
Start orientation



α (solid angle Z)



β (solid angle X)

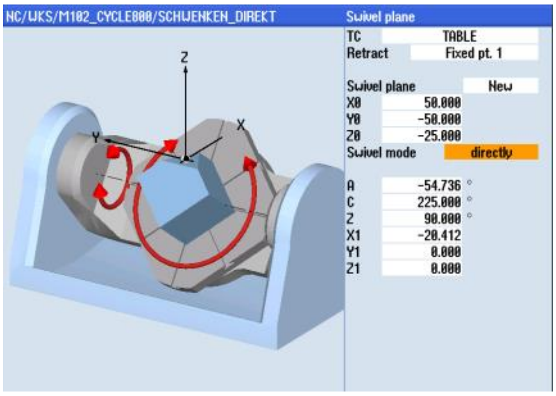


Notes

Swivel mode:

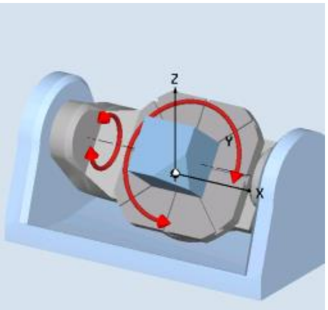
directly

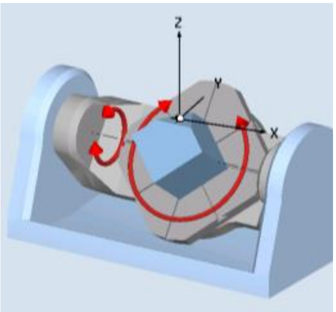
- With swivel mode "**directly",** the WCS positions of the rotary axes can be specified directly. The target swivel positions are entered in the input fields of the rotary axes (e.g. **A, C**) of the selected swivel data record (see parameter _TC).
- A swivel frame is computed for these positions in CYCLE800. This ensures that the tool orientation is aligned at a perpendicular angle to the swivel plane.
- It is possible to enter an additional rotation around the tool orientation in input field "**Z**" (Rotation tool). The reference axis is Z with G17.
  This means that the traversing directions in the plane (with G17 XY) are also precisely defined after swiveling the rotary axes "directly".

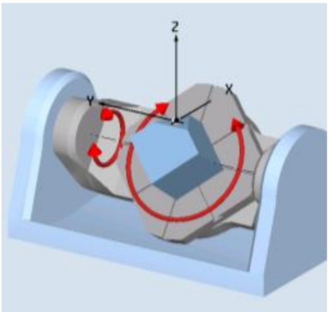


A, C, Z:

A (A-axis rotation) | **C** (C-axis rotation) | **Z** (coordinate rotation)







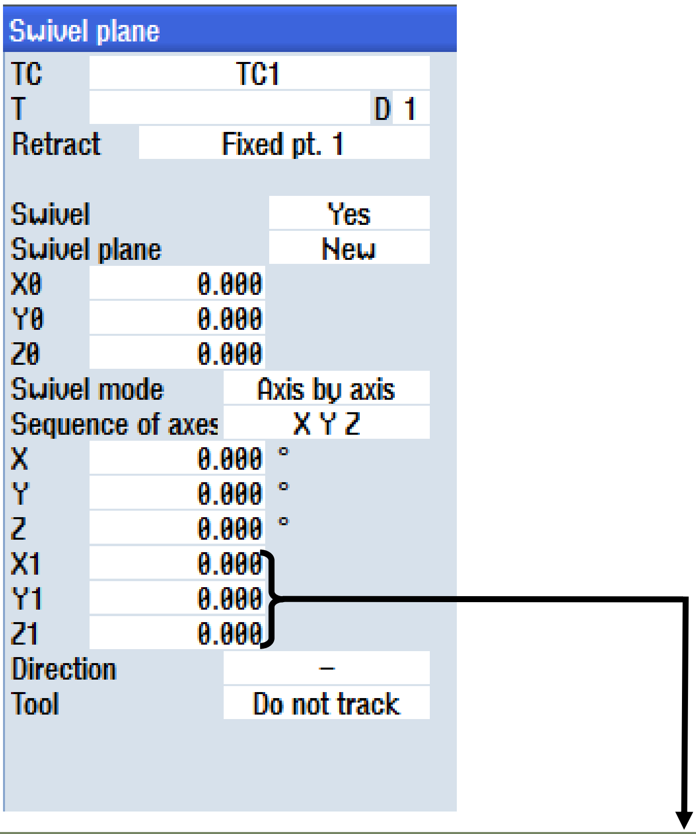
Note:

If swivel mode "**directly**" is applied, the corresponding NC program is machine dependent, i.e. the NC program can run only on machines with the same swivel kinematics.

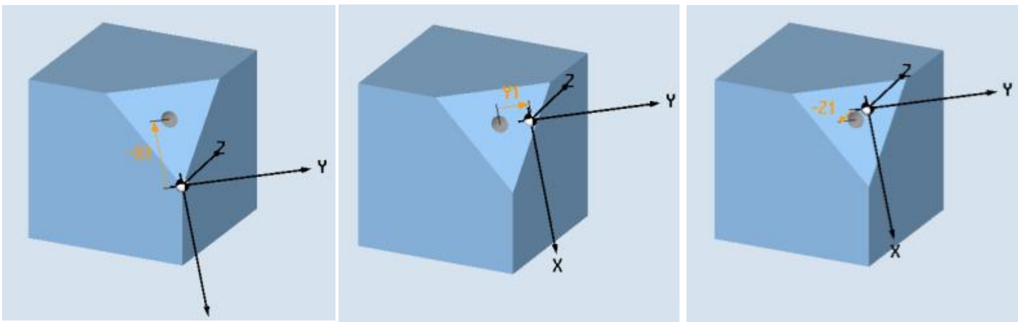
Manual and semi-automatic rotary axes are also compatible with swivel mode "**directly**".

Swivel mode "**directly**" can be selected in the setup area of the swivel data screen if this makes sense technologically.

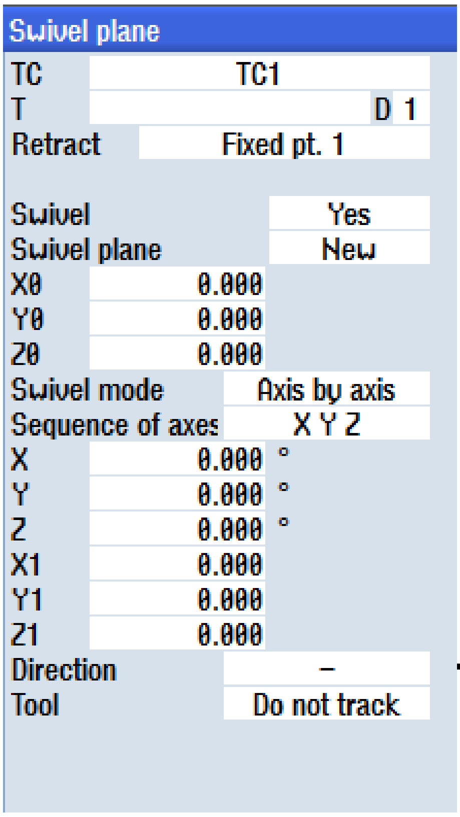
Notes



X1, Y1, Z1:	**Shifting the reference point after rotation of the WCS** Here it is possible to the shift the workpiece zero in the tilted working plane after rotation. The established workpiece zero represents the new workpiece reference for the part program. Additive translation along X Additive translation along Y Additive translation along Z



Notes



Direction:	**Direction reference +/-** Direction reference for traversing direction of 1st and 2nd rotary axis of the active swivel data record (machine kinematics). In the example of a swivel table kinematic the direction reference is set to control the 1st rotary axes (A) and is defined in the swivel data record as such. The NC calculates two possible solutions of the axes rotations programmed in CYCLE800 using the angle traversing range of the rotary axes of the machine kinematics. Usually, only one of these solutions is technologically suitable. The solutions differ by 180° in each case. Selecting the "**minus**" or "**plus**" direction determines which of the two possible solutions is to be applied. The following applies: Plus = smaller traverse path of the 2nd rotary axis (type P and M) Minus = larger traverse path of the 2nd rotary axis (type P and M)

Note:

The selected direction reference in the swivel data record e.g. **Rotary axis 1, + direction selected** or **Rotary axis 1, - direction selected** refers also to the default traversing direction for the initial kinematic setting in the swivel cycle when pressing the softkey **"initial setting**".

Notes

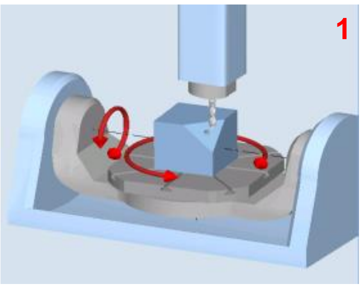
Direction

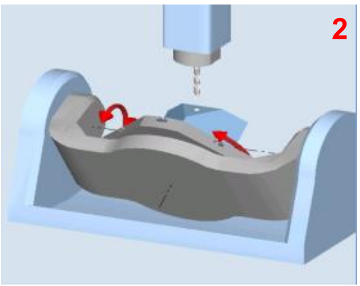
A new swivel frame based on the kinematics of the machine is now being computed. The control has to position now its physical machine rotary axes **A** and **C** (1) two the computed rotary axes positions, in order to align the tool with the inclined plane (2).

The NC calculates two possible solutions of the rotation programmed in CYCLE800 using the angle traversing range of the rotary axes of the machine kinematics. Usually, only one of these solutions is technologically suitable. The solutions differ by 180° in each case. Since both solutions are within the possible angle range of the rotary axes it is possible to swivel in both directions. Selecting the "**minus**" or "**plus**" direction determines which of the two possible solutions is to be applied.

Solution 1

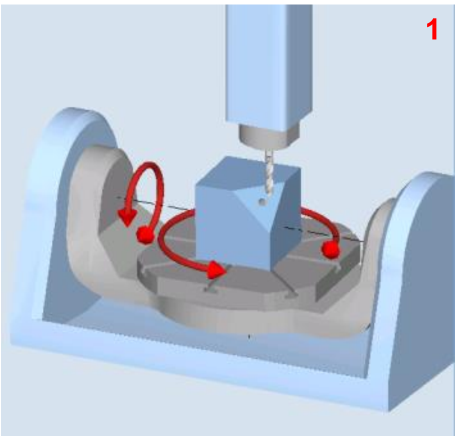
Traversing direction of the 1. rotary axis (A) is “**plus**”
Calculated rotary axes positions:
A-axis (table) = +54.736°
C-axis (table) = +45°

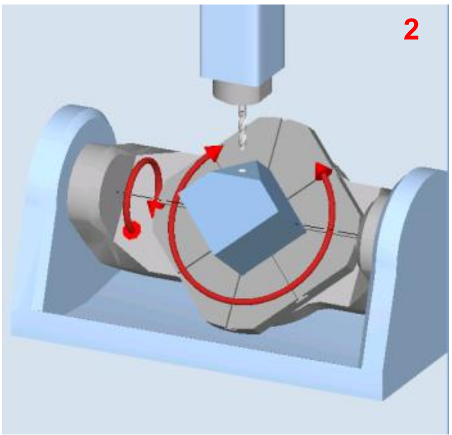




Solution 2

Traversing direction of the 1. rotary axis (A) is “**minus**”
Calculated rotary axes positions:
A-axis (table) = - 54.736°
C-axis (table) = +225°





Notes

Direction:

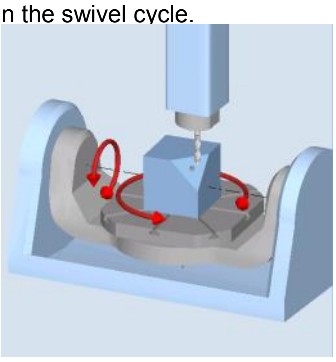
Example 1:

Rule for type P:
Rotary axis 2 (C) is based on rotary axis 1 (A)

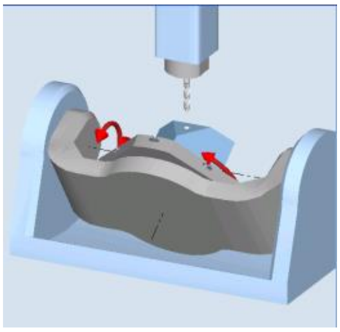
- Machine kinematics with rotary swivel table.
  1. Rotary axis = A rotation around X (angle range -90° to +90°)
  2. Rotary axis = C rotation around Z (angle range 0° to 360° modulo)

- Machine manufacturer has set the preferred direction of rotary axis to "**Rotary axis 1, + direction selected** ", since the smaller traverse path of the rotary axis is being used.

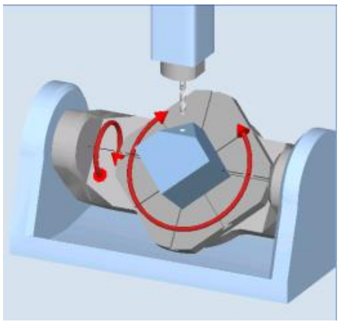
- Swivel mode is "**axis by axis**" a rotation around Z of -45° and Y of 54.736° are programmed in the swivel cycle.



Base orientation
Initial kinematic setting:
A-axis (table) = 0°
C-axis (table) = 0°



Solution 1
Traversing direction of 1st rotary axis (A) is "**plus**"
Movement:
A-axis (table) = +54.736°
C-axis (table) = +45°
(Modulo 360)



Solution 2
Traversing direction of 1st rotary axis (A) is "**minus**"
Movement:
A-axis (table) = -54.735°
C-axis (table) = +225°
(Modulo 360)

Notes

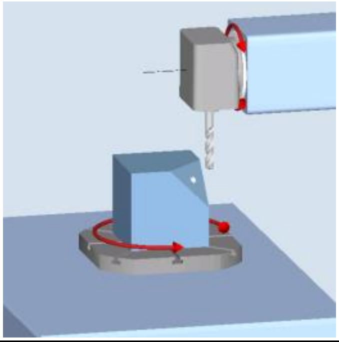
Direction:

Example 2:

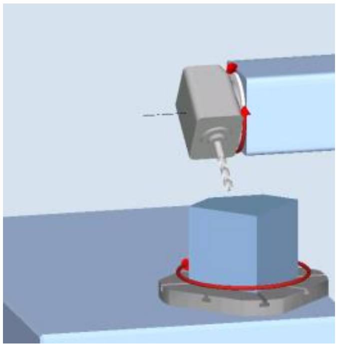
Rule for type M:
With mixed kinematics, rotary axis 1 is always the axis that changes the tool orientation.

- Machine with mixed kinematics.
  1. Rotary axis = B rotation around Y (angle range –110° to +60°)
  2. Rotary axis = C rotation around Z (angle range 0° to +360° modulo).

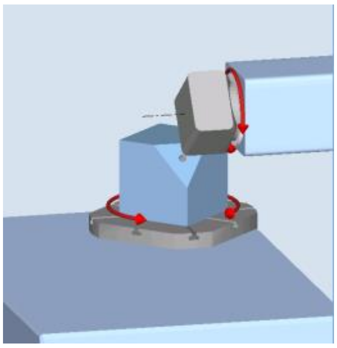
- Machine manufacturer has set the preferred direction of rotary axis to “**Rotary axis 1, - direction selected** “, since the B-axis has the bigger range in minus direction.
- Swivel mode is “axis by axis” a rotation around Z of -45° and Y of 54.736° is programmed in the swivel cycle.



Base orientation
Initial kinematic setting:
B-axis (Head) = 0°
C-axis (table) = 0°



Solution 1
Traversing direction of 1st rotary axis (B) is “**plus**”
Movement:
B-axis (Head) = +54.736°
C-axis (table) = +45°
(Modulo 360)



Solution 2
Traversing direction of 1st rotary axis (A) is “**minus**”
Movement:
B-axis (Head) = - 54.735°
C-axis (table) = +225°
(Modulo 360)

Notes

Direction:	**Example 3:**

Direction:

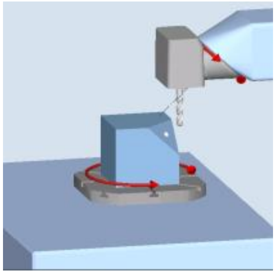
Example 3:

Rule for type M:
With mixed kinematics, rotary axis 1 is always the axis that changes the tool orientation.

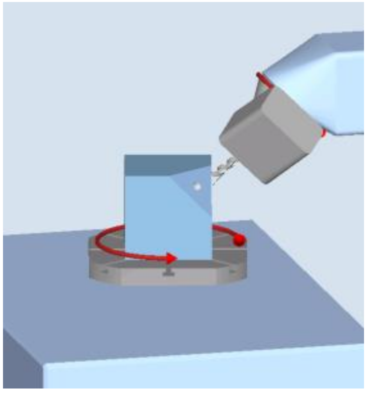
- Machine with mixed kinematics.
  1. Rotary axis = B rotation around Y (angle range 0° to +180°)
  2. Rotary axis = C rotation around Z (angle range 0° to +360° modulo)

- Machine manufacturer has set the preferred direction of rotary axis to **"No: display, +direction"**, since the B-axis can only move in plus direction

- Swivel mode is "axis by axis" a rotation around Z of -45° and Y of 54.736° is programmed in the swivel cycle.



Base orientation
Initial kinematic setting:
B-axis (Head) = 0°
C-axis (table) = 0°



Only one possible solution, because of negative B-Axis limitation
Movement:
B-axis (Head) = +81.101°
C-axis (table) = +283.825°

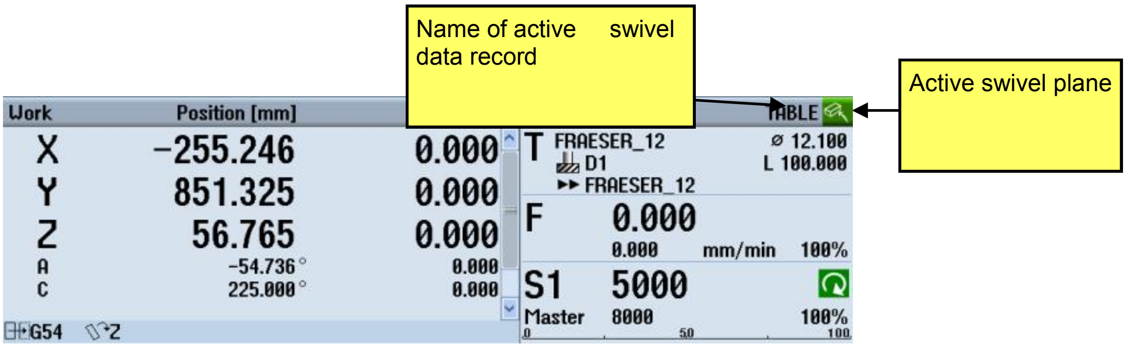
Note:

Because of the B-Axis limitation in negative direction it makes sense to deactivate the direction selection in the cycle input mask. This can be done in the Setup area of the swivel data record.

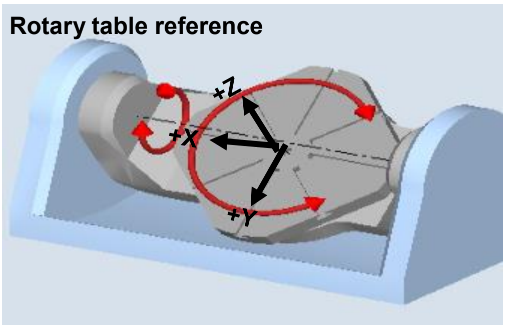
Notes

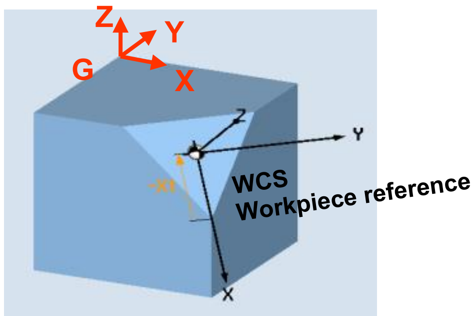
In our example the following rotary axes positions are approached with the direction selection “**minus**” of the resulting swivel frame:

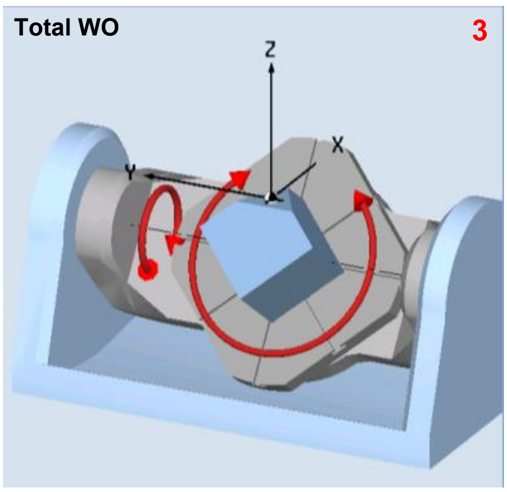
A -54.736° and **C 225.000°**

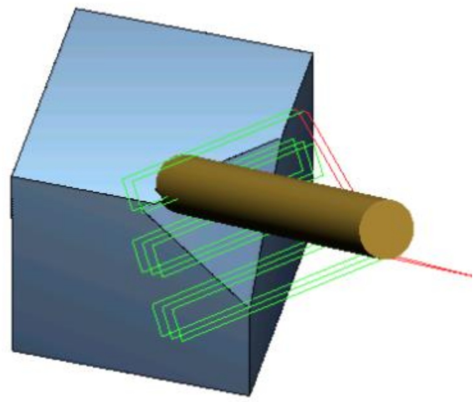


The tool is now being aligned with the new machining plane. All subsequent traversing commands then refer to the new, with frames shifted and rotated workpiece coordinate system (3).









Notes

Workpiece zero offset - Rotary table reference - Workpiece reference

The workpiece zero offset, the programmed translations and coordinate rotations are combined into a swivel frame and stored as a chaining of individual system frames. This allowes retracting the tool in a swiveled stage away from the workpiece (e.g. inclined hole) after reset or even after a power failure.

Work offset - active [mm]

	X	Y	Z	A	C
Machine act value	186.933	113.867	300.000	-54.736	225.000
Rotary table ref.	568.191	302.053	34.517	0.000	0.000
G54	-198.213	-202.199	109.319	0.000	0.000
Workpiece ref.	41.667	-41.667	-8.334	0.000	0.000
Total WO	851.325	255.246	143.235	0.000	0.000
Tool: FRAESER_12	0.000	0.000	-100.000	0.000	0.000
Work actual value	-141.380	664.391	56.765	-54.736	225.000

Frame chain of all active frames

Work offset - details: Rotary table ref. [mm]

	Coarse	Fine			
X	568.191	0.000	-45.000	1.000	
Y	302.053	0.000	-35.265	1.000	
Z	34.517	0.000	150.000	1.000	
A	0.000	0.000		1.000	
C	0.000	0.000		1.000	
SP1	0.000	0.000		1.000	

Rotary table ref.
$P_PARTFRAME

Work offset - details: G54 [mm]

	Coarse	Fine			
X	-198.213	0.000	0.000	1.000	
Y	-202.199	0.000	0.000	1.000	
Z	109.319	0.000	0.000	1.000	
A	0.000	0.000		1.000	
C	0.000	0.000		1.000	
SP1	0.000	0.000		1.000	

Storable WO G54
$P_UIFR[1]

Work offset - details: Workpiece ref. [mm]

	Coarse	Fine			
X	41.667	0.000	0.000	1.000	
Y	-41.667	0.000	54.736	1.000	
Z	-8.334	0.000	-45.000	1.000	
A	0.000	0.000		1.000	
C	0.000	0.000		1.000	
SP1	0.000	0.000		1.000	

Workpiece reference
$P_WPFRAME

Work offset - details: Total WO [°]

	Coarse	Fine			
X	851.325	0.000	0.000	1.000	
Y	255.246	0.000	0.000	1.000	
Z	143.235	0.000	90.000	1.000	
A	0.000	0.000		1.000	
C	0.000	0.000		1.000	
SP1	0.000	0.000		1.000	

Total WO
(Result of frame chain)

Notes

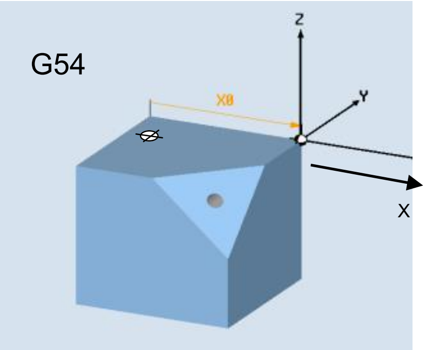
Swivel mode „directly“

The same example can also be programmed with direct programming of rotary axis positions. In this case swivel mode “**directly**” must be selected.

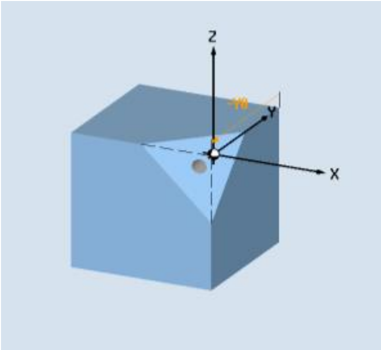
Direct swivelling into a inclined machining plane basically requires the following steps:

1. Translation of the WCS before rotation
2. Direct positions of rotary axis 1 and 2
3. Rotation of the WCS about the surface normal
4. Translation of the WCS after rotation in the new swivel plane

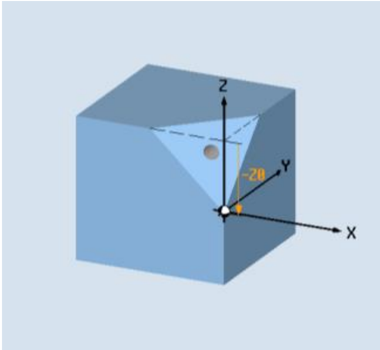
1. Translation f the WCS before rotation



Workpiece zero shift along the

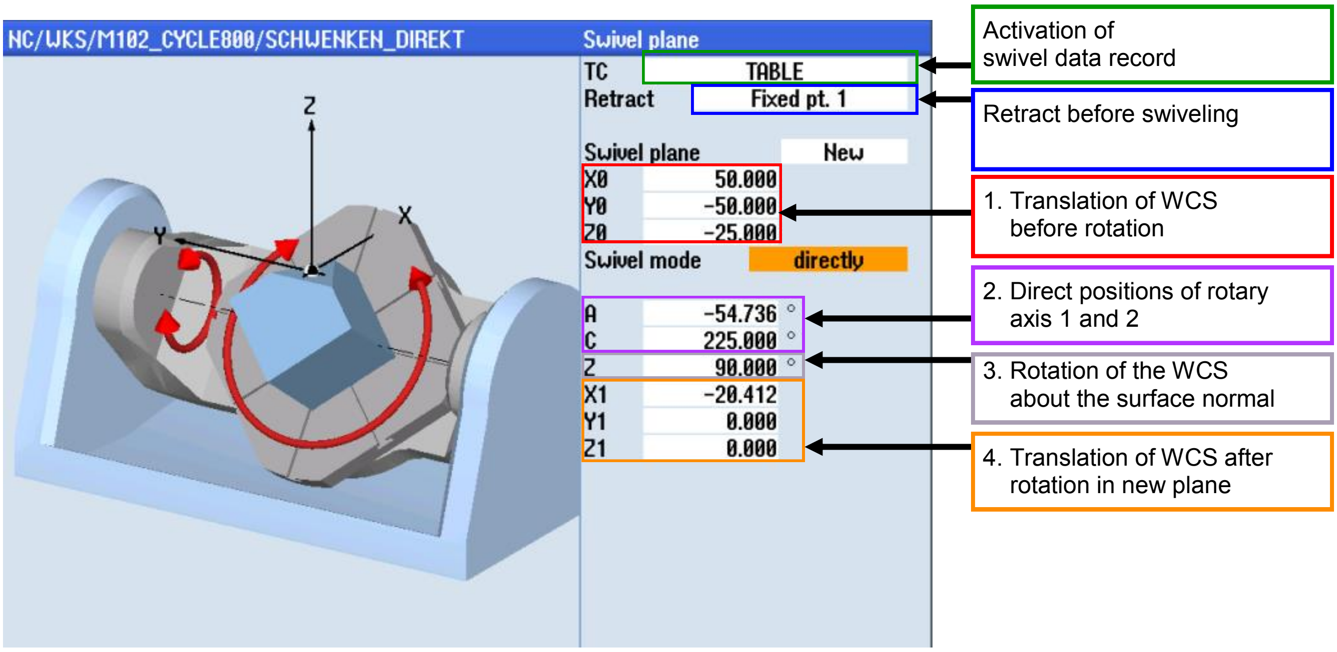


Workpiece zero shift along the Y axis.



Workpiece zero shift along the

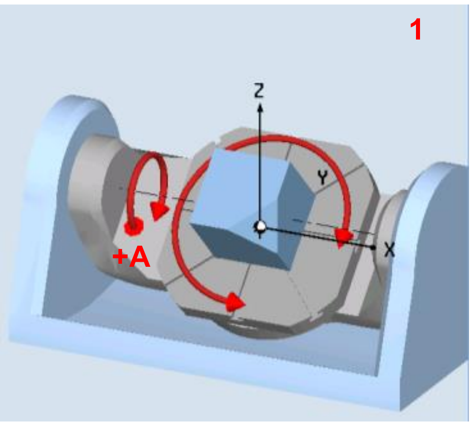
Steps 1 to 4 define the “**swivel frame**” in other words the new coordinate system which defines the reference for the machining program.

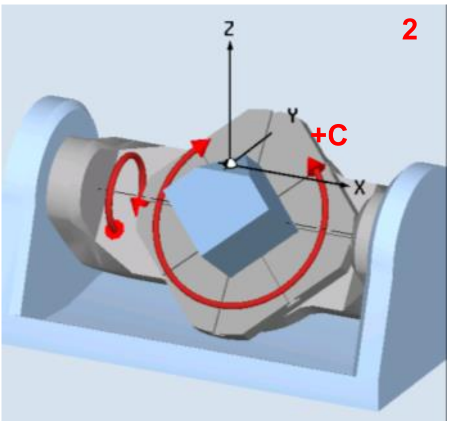


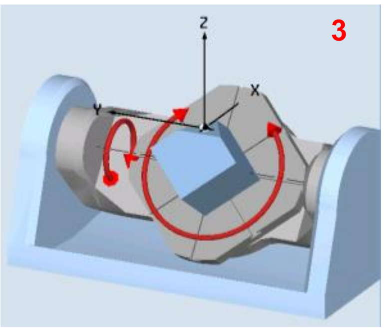
Notes

2. Direct positions of rotary axis 1 and 2

With selection of Swivel mode „**directly**“ the first rotary axis **A** is positioned to -54.736° (1) and the second rotary axis **C** to 225°, so that the inclined plane of the cube is in the XY plane (G17) (2) .

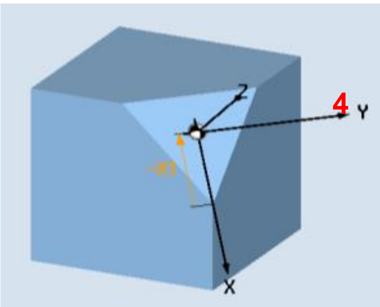






3. Rotation of the WCS about the surface normal

The coordinate system must now be rotated in such a way
that the Y axis is aligned with the upper edge of the cube (e.g. **Z**=90°).



4. Translation of the WCS after rotation in the new swivel plane

The last step is a additive shift of the coordinate system along the X axis to the center of the hole (e.g. **X1**=-20.412).

Notes

Swivel function in JOG mode

Explanation of the swivel function in JOG

The function “swivel” in JOG is used to set up work pieces (alignment of workpiece) with swivelled (inclined) machining planes and represents the prerequisite for further Set-up operations (scratching/measuring workpiece) and machining.

Swivelling in JOG mode is supported by its own input screen form.

Once the input values are applied and the NC is started, the rotary axes are positioned automatically (in case of automatic rotary axes) and a swivel frame is calculated for this plane.
The swivel frame ensures that the tool orientation is aligned perpendicular to the swivel plane.

The following functions can be carried out with "swivelling in JOG":

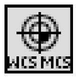
- Activation of a swivel data record (orientable toolcarrier TCARR=...)
- New swivelling or additive swivelling to any plane if permitted by the machine kinematics.
- Workpiece alignment through compensation of basic rotations in reference to the machine coordinate system, by means of setting a new swivel plane as new zero plan. The new zero plan is then reference for all subsequent machining operations or workpiece measurement.
- Moving the machine rotary axes into the initial kinematic setting in reference to the new “zero plane” with automatic retraction of the linear axes (e.g. Fixed pt. 1 or 2, max tool direction or inc. tool direction).
- Deactivation of swivel data record (TCARR=0)

Example:

In Manual Operation Mode “**JOG**” it is possible for the machine operator to define a swivel plane with the softkey **„Swivel“** in an easy and fast way. In secession it is then possible to measure the workpiece with help of the workpiece measuring functions or face mill the workpiece with the face milling function.

The machine operator can set a work offset manually or use a probe to set a work offset in any swivelled plane. In this case, the determined values of the zero offset refer to the initial setting or the basic setting of the machine kinematics.

The machine operator can also specify the plane to be defined as the new zero plane. In this case, the determined values of the zero offset refer to the new zero plane.



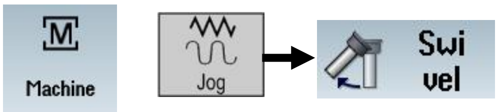
Traversing axes in JOG with active swivel plane:

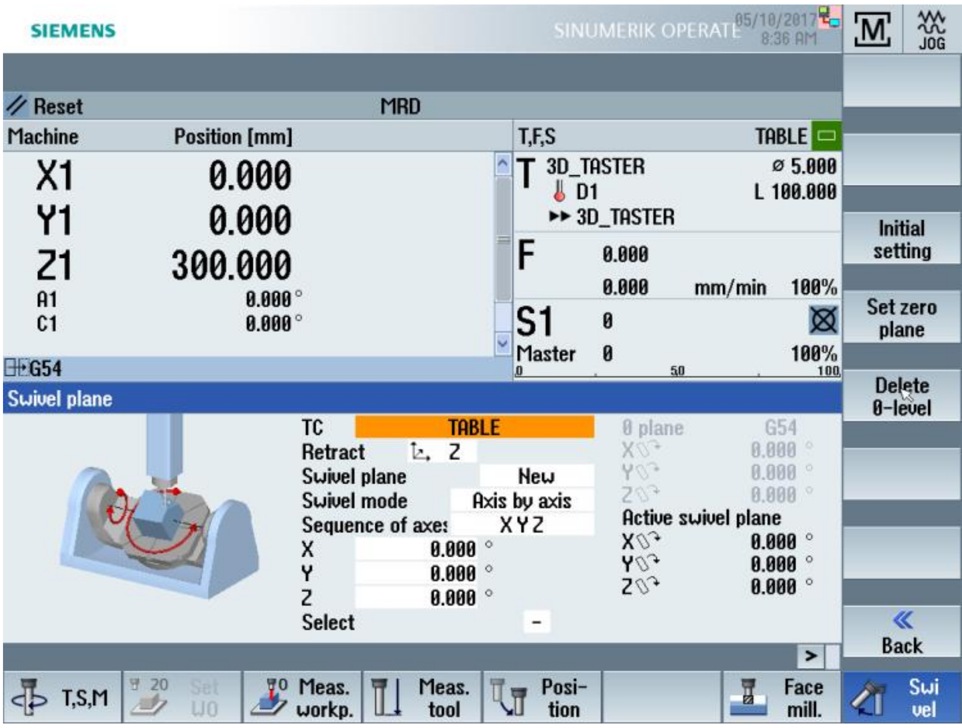
The Softkey “**WCS MCS**” allows switching between axes traversal relative to the workpiece coordinate system (WCS) or the machine coordinate system (MCS). Pressing the Softkey ensures that the geometry axes are traversed (green LED lit = WCS) and not the machine axes, if retraction from a oblique surface or hole is required (only for kinematic type T and M).

Notes

Procedure for swivelling in JOG

Press the following softkeys to open the “**swivel function**” in JOG mode





Travers rotary axes to Initial orientation relative to zero plane

Set active swivel plane as new zero plane in the active storable work offset

Delete active zero plane in the storable work offset (basic initial setting).

The input mask for the function "Swivel plane” in “JOG" mode corresponds to a shortened version of the CYCLE800 input mask. Translation before and after rotation is here not available.

- **TC:**
  Here you can select the desired Swivel data record (TCARR) in our example “**TABLE**” (corresponds to a swivel table kinematic type P in the swivel data record).

- **Retract:**
  Select the desired retract mode e.g. „**Fixed pt.1**” (The available retract possibilities in the cycle input mask and the retract position can be set in the area “Setup” of the Swivel data record.

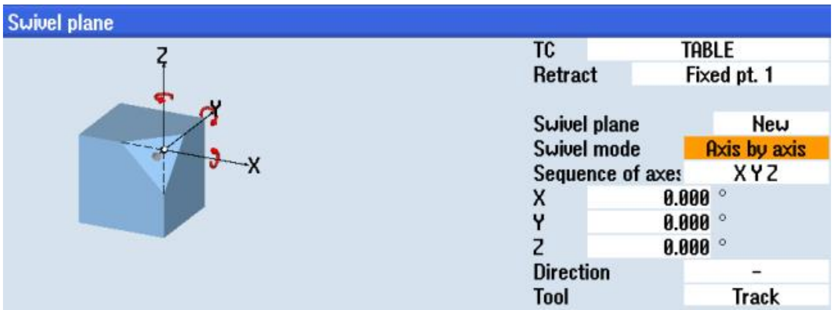
- **Swivel plane:**
  “new“ (absolute) or **„additive“** (incremental).

- **Swivel mode:**
  Selection of swivel mode „**axis by axis**“ or „**direct**“.

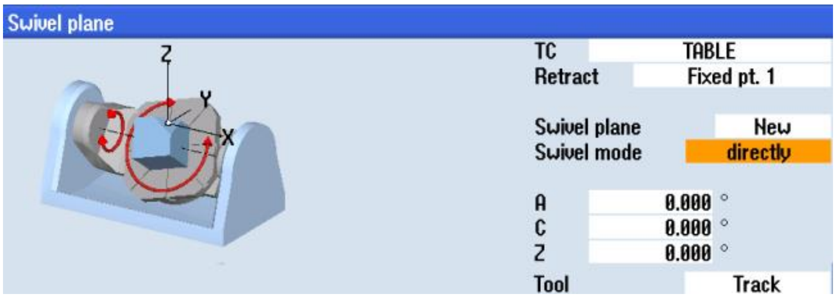
- **Sequence of axes:**
  Here you can freely select the input order for the rotation about the geometry axes XYZ.

Notes

- If swivel mode „**axis by axis**“ is selected, input the desired swivel angles as rotation about the geometry axes **X Y Z** .



- If swivel mode “**directly**” is selected, input for **A** and **C** the desired angles for each rotary axis. For **Z** you can program a subsequent coordinate rotation about the Z axis, in order to align the workpiece edge with the transformed coordinate system.



Note:

With swivel mode **“direct”** no coordinate transformation takes place on a programmed C axis rotation. The machine rotary axis only moves to the programmed axis position whereby the active work offset is maintained. If the coordinate system is to be rotated in parallel to the rotation of the C axis, then a additional rotation about the Z axis is required and must be programmed in the subsequent parameter “Z” (rotation about Z-axis).

- **Direction:** determines in which of the two available possibilities a work plane is reached. Usually, only one of these solutions is technologically suitable. The solutions differ by 180 degrees in each case. Selecting the **"minus"** or **"plus"** direction determines which of the two possible solutions is to be applied. The axis to be affected by this setting can be selected in the are Set-up of the swivel data mask.This parameter is not available for swivel mode direct since the direction of the rotary axes is already determined by the rotary axis sign

- **Tool:** here you can select weather the tool tip is being tracked, by means of maintaining the relative position during positioning of the rotary axes. This function requires the optional 5-axis transformation function (TRAORI). The "**Tool**" input field can be enabled in the Setup area of the swivel data record.

Notes

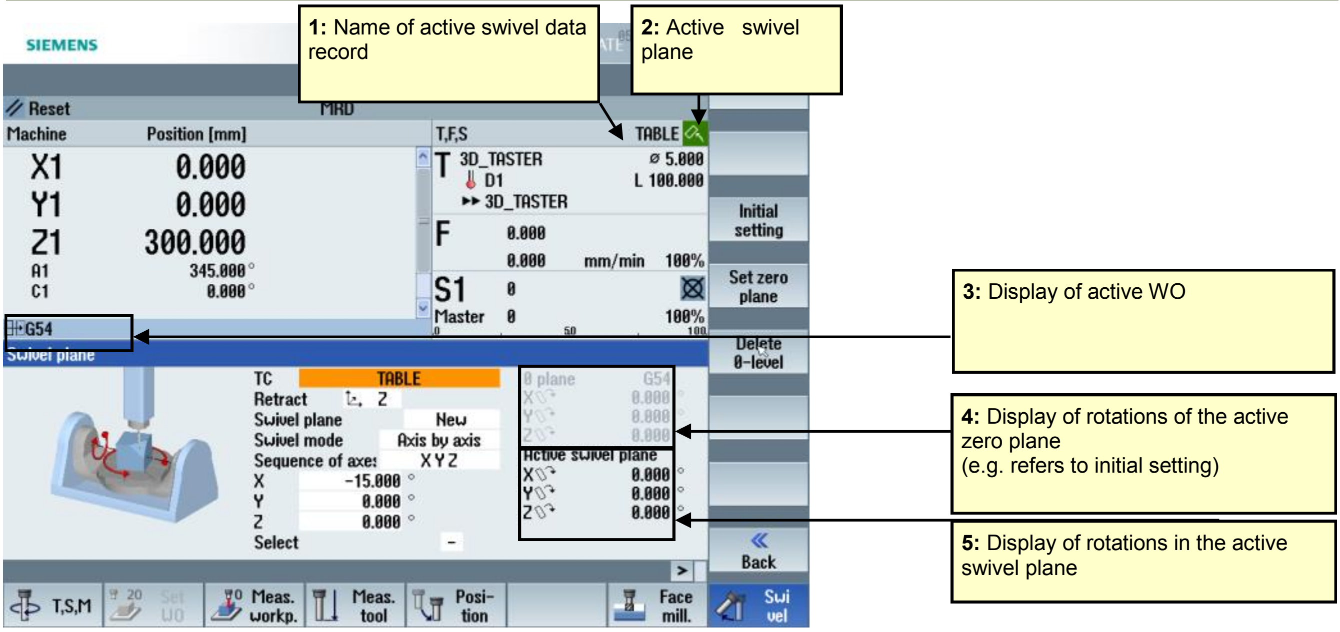
Application example swivelling in JOG

This example shows how the machine operator can create a swivelled work plane and set this plane as new workpiece reference in a storable work offset.

In this case, the determined values of the work offset refer to the initial setting (basic setting of the machine kinematics). The machine operator can then specify the swivel plane as the new zero plane.

Procedure:

- Select the swivel data record in the input field “**TC**” (e.g. “TABLE”)
- Input swivel angle as rotation about the X axis of -15°
- Press **"CYCLE-START"** to activate the swivel data record for the “**TABLE**” kinematic (see 1) and to activate the new swivel plane (see 2).
- The machine now swivels with the A and C axes in the new working plane and a coordinate rotation around geometry axis Z is activated.



Note:

The rotary axis minimum range for the A axis in this example is limited to –10 degrees.
As a result, there is only one solution to reach this plane, by rotating the C axis to 180 degrees and the A axis to +15 degrees.
The rotary axis range can be set in the “Setup” area of the swivel data record.

Notes

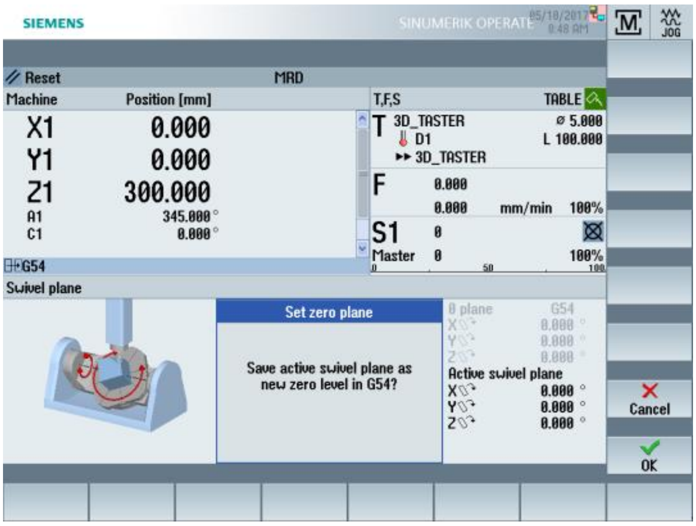
The programmed swivel frame causes a translation and a rotation upon activation of the toolcarrier (TCARR). The translation and rotation are being stored into the system frame for "rotary table reference". The values are being displayed under work offset details rotary axes reference.

Work offset - details: Rotary table ref. [°]

	Coarse	Fine			
X	0.000	0.000	15.000	1.000	☐
Y	6.815	0.000	0.000	1.000	☐
Z	51.764	0.000	180.000	1.000	☐
A	0.000	0.000		1.000	☐
C	0.000	0.000		1.000	☐
SP1	0.000	0.000		1.000	☐

- Press the „**Set zero plane**" softkey in order to set the active swivel plane as new zero plane (initial setting) in the work offset G54

- A window with the query „**Set zero plane**" appears.

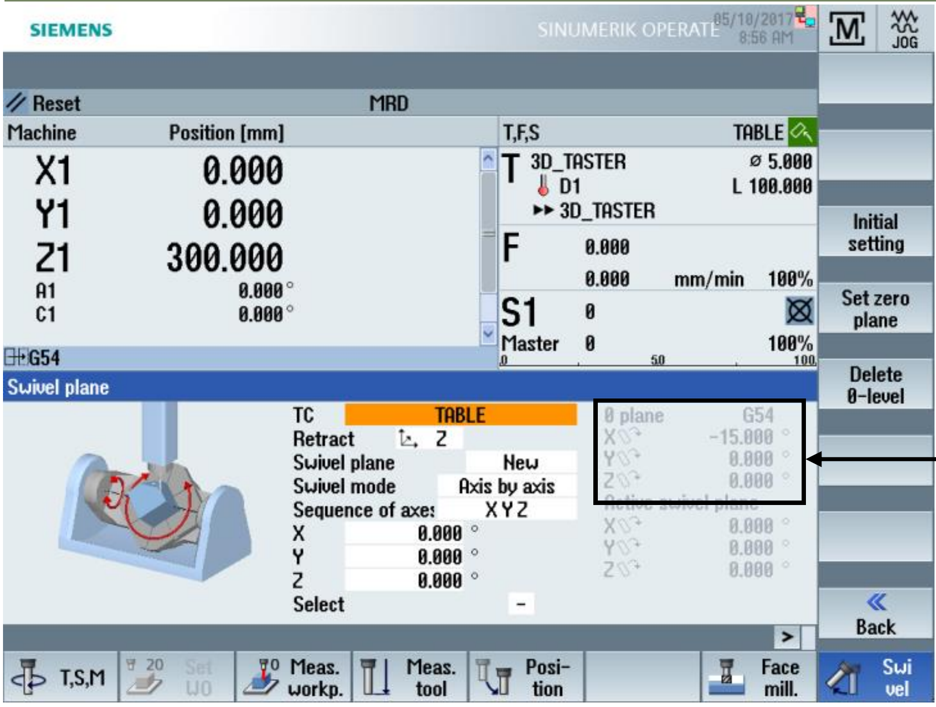


- Press the softkey "**OK**" to confirm saving the active swivel plane as new

Result:

The inclined work piece plane is now aligned for further machining or re-machining. When measuring in this case, the determined values of the work offset e.g. G54 refer now to the new zero plane (initial setting). When traversing to initial setting the rotary axis positions do refer now to the new zero plane stored under G54 and do no longer correspond to the original initial setting relative to the MCS that are agreed upon in the swivel data record.

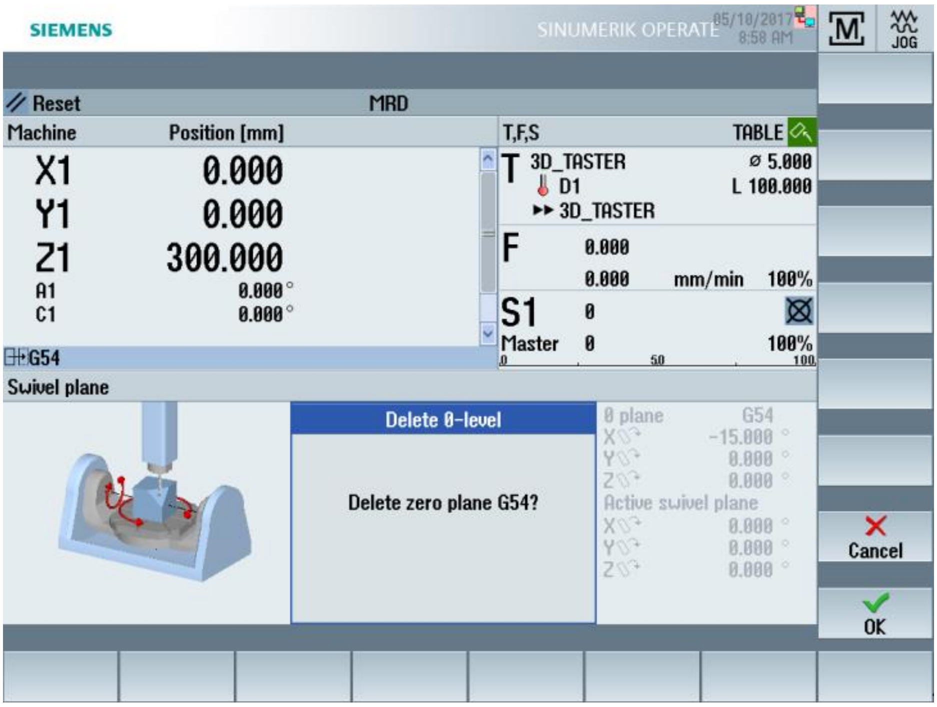
Notes



4: Display of the rotation in the active zero plane in active work offset G54 (corresponds to new reference for initial setting in G54).

- To delete the active zero plane in the active work offset, press the "**Delete 0-level**" softkey.

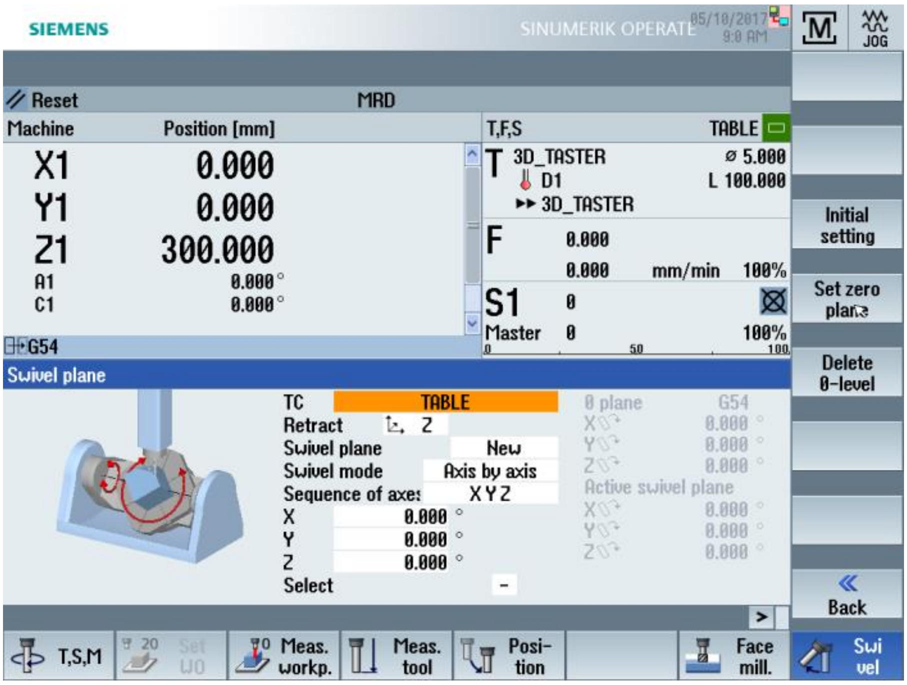
- A window with the query „Delete zero plane G54?“ appears on the screen.



- Press the "**OK**" softkey to confirm deleting the active zero plane in G54.

Notes

- With the softkey „**Initial setting**“ all default settings are being reset and all rotations set back to zero. With „**CYCLE START**“ the rotary axes return to the initial position. All values of the previously active zero plane are cleared and the swivel plane is no longer active.



Note:

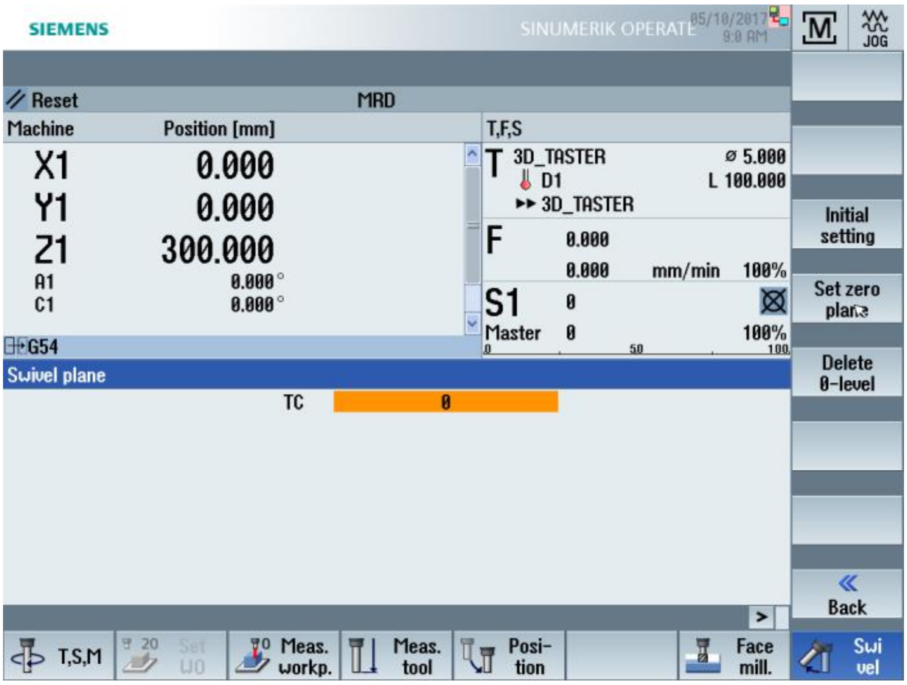
The default setting for parameter “**swivel mode**” is always “**axis by axis**”.

The default setting for parameter “**direction**” can be set in the swivel data record.

Default for **“Retract”** can be set in setting data ***SD 55420***

Default for **“Tool tracking”** can be set in channel setting data ***SD 55421***

- The tool carrier (TCARR) can be deselected with selection of **TC “0”** (equal to TCARR=0) in the swivel cycle. Press „**CYCLE START**“ to activate.



Note:

Deselection of the swivel data record can be enabled in the channel setting data ***SD 55221*** with setting of ***Bit 2***.

Notes

CYCLE800 in AUTOMATIC mode

Overview variations of CYCLE800

The swivel CYCLE800 in "AUTOMATIC" mode is used for automated machining or measuring in inclined planes in a program.

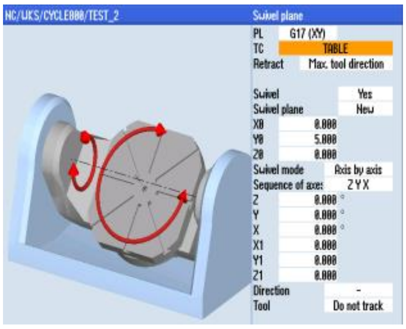
CYCLE800 is available in two programming formats:

- programGUIDE G code or
- ShopMill

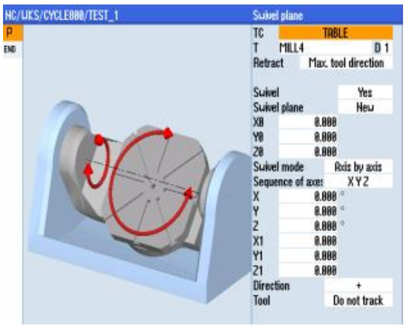
CYCLE800 comes in two basic variations for the technology milling:

- CYCLE800 "Swivel plane" (see also section 1.8)
- CYCLE800 "Approach tool" (see also section 1.9)

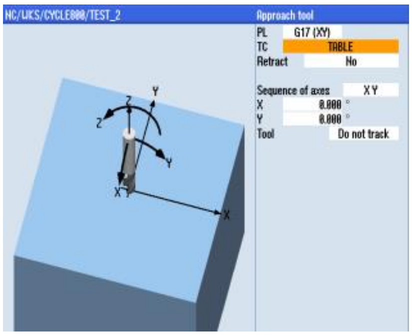
CYCLE800 "Swivel plane"
input mask **programGUIDE G code**



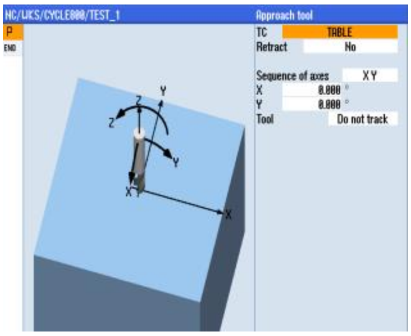
CYCLE800 "Swivel plane"
input mask **ShopMill**



CYCLE800 "Approach tool"
input mask **programGUIDE G code**



CYCLE800 "Approach tool"
input mask **ShopMill**



Notes

Important programming notes

Program header

- The main program with a swivelled plane should start in the basic setting (initial setting) of the machine.
- For a proper simulation of the workpiece it is required to initialize the CYCLE800 once in the kinematic base orientation (initial settings) and swivel to zero, after that the cycle should be cancelled.
- The definition of the blank (WORKPIECE) always refers to the currently active work offset.
- A tool (tool cutting edge D > 0) and the work offset (e.g. G54), with which the workpiece was scratched or measured, must be programmed before the swivel cycle is first called in the main program.

Example:

```
N100 ;*** Swivel into kinematic base orientation (initial setting) ***
N101 CYCLE800(4,"TABLE",200000,57,0,0,0,0,0,0,0,0,0,1,,1)
N102 CYCLE800()
N103 G54 G17
N104 WORKPIECE(,,,"BOX",112,0,51,-80,-2.5,-2.5,102.5,102.5)
N105 T10 D1
N106 M6
N107 S5000 M3
N108 G0 X0 Y0 M8
N109 ;*** Swivel plane  X=-15 deg ***
N110 CYCLE800(4,"TABLE",200000,57,0,0,50,-15,0,0,0,0,0,1,,1)
…..
```

FRAMES

Note:

For ShopMill programs, the blank in the program header is automatically referred to the base orientation (initial kinematic setting).

- There are no programmable frames used in the swivel cycle. Previously programmed frames by the user are only taken into account for additive swivelling. On the other hand, when swivelling to a new swivel plane the programmable frames are deleted (TRANS, ROT, SCALE, MIRROR).
- The last swivel plane remains active after a program reset or when the power fails. The behavior at reset and power on can be set in the machine data (see section 4.25 *Configuration of machine data*).

Notes

Deselection of swivel cycle

- Deselection of the swivel data record and deletion of the swivel frame (WPFRAME, PARTFRAME, TOOLFRAME) can also be programmed with CYCLE800().

Traversing the axes in JOG with active swivel plane

- Traversing the axes in the active swivel plane in JOG is possible when the WCS key is active on the machine control panel. The machine axes are then not traversed, but instead the geometry axes.
 (**Notice!** Observe the notes of the OEM).

Aligning the workpiece

- If the workpiece is aligned via coordinate rotation or an offset of the rotary axes (e.g. A,C), then the offset values entered in the work offset are taken into account when swiveling with CYCLE800.

Block search when swivelling the plane / swivelling the tool

- For block search with calculation, after NC start, initially, the automatic rotary axes of the active swivel data set are pre-positioned and then the remaining machine axes are positioned. (This does not apply if transformation type TRACYL ,TRANSMIT or TRAORI is active after block search. In this case, all axes simultaneously move to the positions.)

Approaching a machining operation

- When approaching the programmed machining operation in the swivelled plane for kinematic types T and M, under worst case conditions, the software limit switches could be violated. In this case, the system travels along the software limit switches above the retraction plane. In the event of violation below the retraction plane, for safety reasons, the program is interrupted with an alarm. To avoid this, before swivelling, e.g. move the tool in the X/Y plane and position it as close as possible to the starting point of the machining operation or define the retraction plane closer to the workpiece.

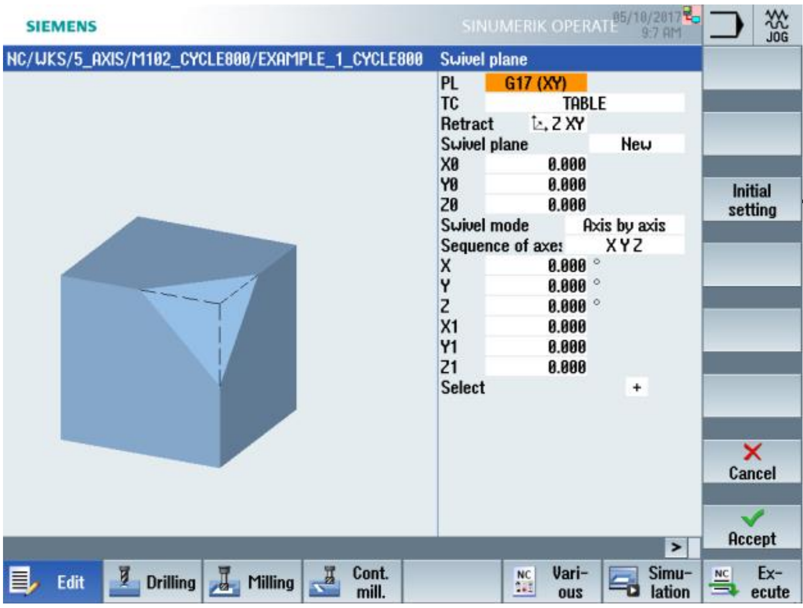
Retraction

- Before swivelling the axes you can move the tool to a safe retraction position. The retraction variants available are defined in the Setup area of the swivel data mask. The retraction mode is modal. When a tool is changed or after a block search, the retraction mode last set is used.

Notes

CYCLE800 "Swivel plane" in programGUIDE G code

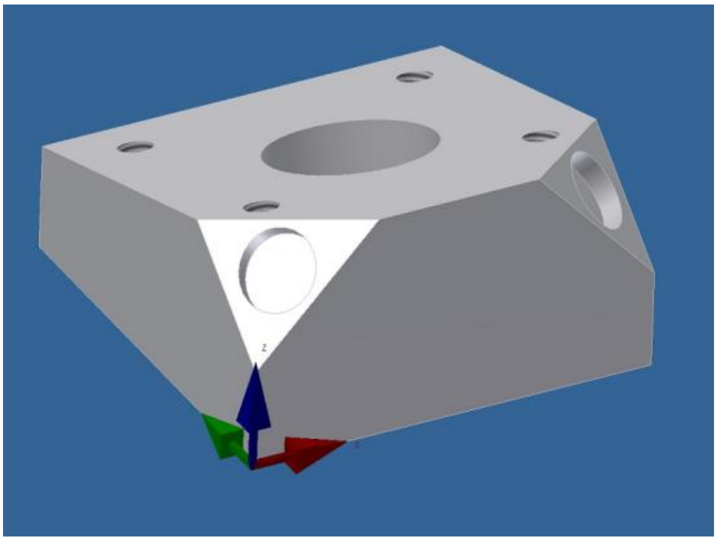
Press the following softkeys to open the CYCLE800 "**swivel plane**" in a part program:



With the softkey „**Initial setting**" all default settings are being reset and all rotations set back to zero. During program main run the initial kinematic setting is approached.

Notes

Programming example 1: CYCLE800 "Swivel plane" in programGUIDE G-Code



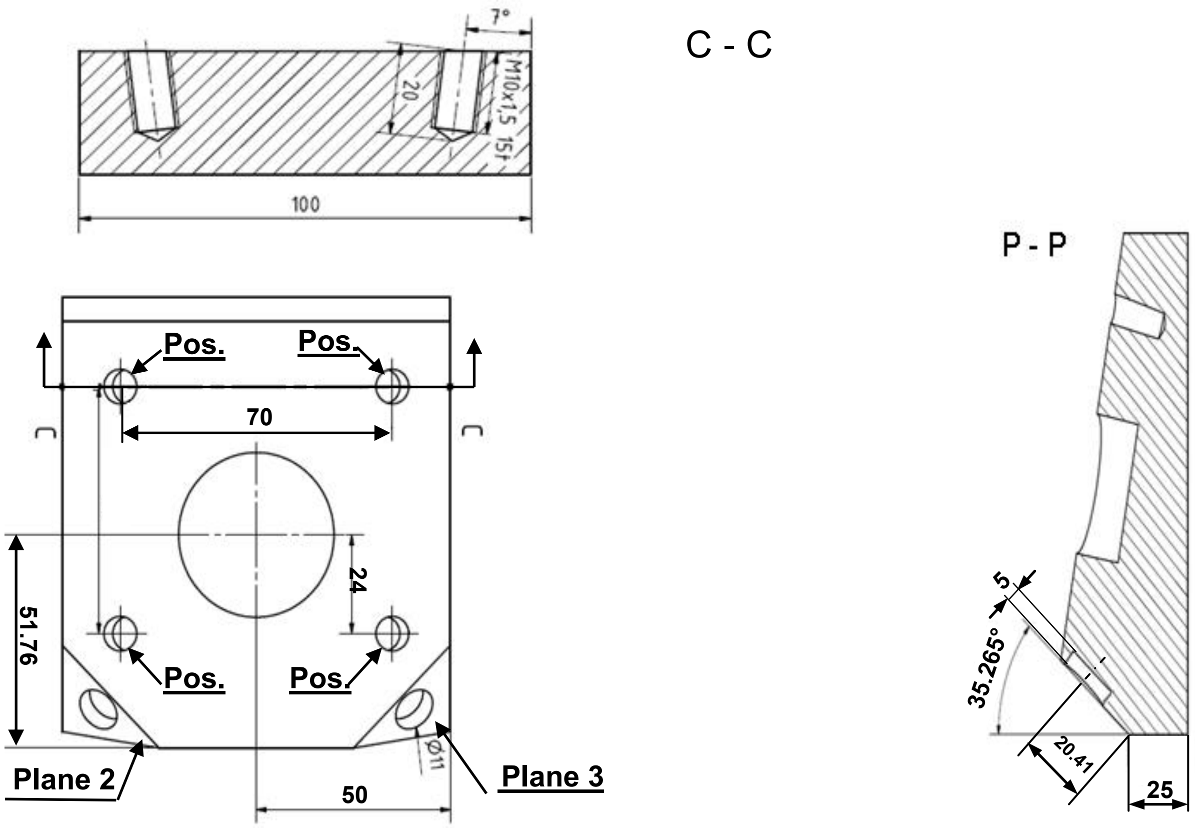
Work Steps

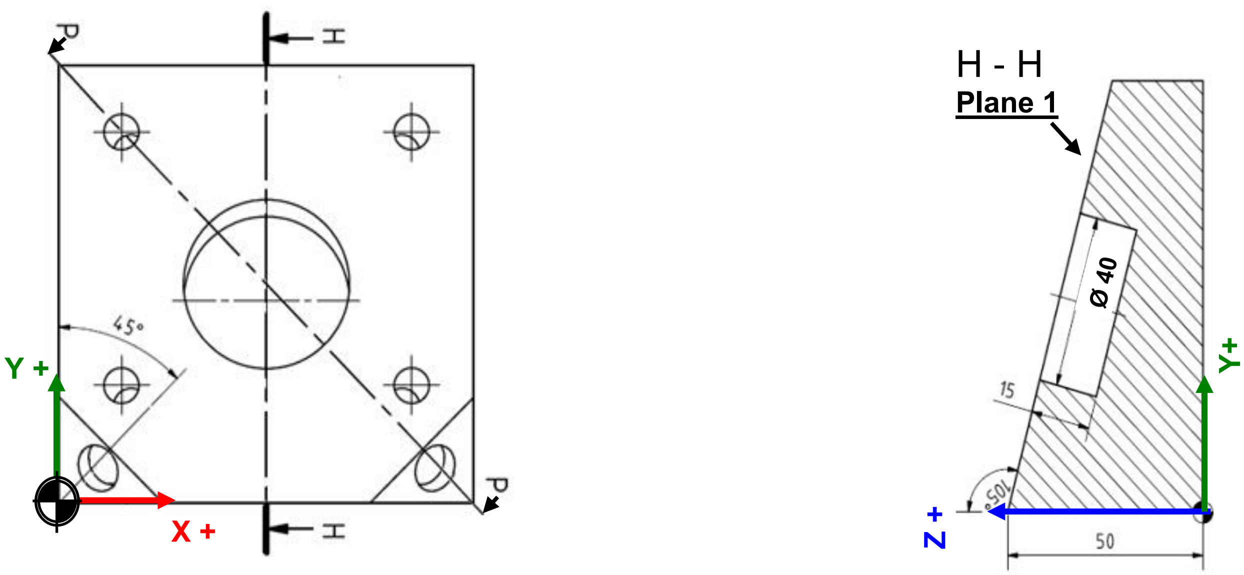
1. Square block 100 x 100 to size (initial position)
2. Face milling of workpiece (initial position)
3. Face milling inclined plane 15° (plane 1)
4. Circular pocket milling Ø40mm 15° (plane 1)
5. Spot face positions at -7° (additive from plane 1)
6. Spot face positions at +7° (additive from plane 1)
7. Drill positions Ø8.5mm at +7° (additive from plane 1)
8. Drill positions Ø8.5mm at -7° (additive from plane 1)
9. Tap positions M10x1.5 at -7° (additive from plane 1)
10. Tap positions M10x1.5 at +7° (additive from plane 1)
11. Face milling oblique plane (plane 2)
12. Face milling oblique plane (plane 3)
13. Circular pocket milling Ø11mm on oblique plane (plane3)
14. Circular pocket milling Ø11mm on oblique plane (plane 2)

Tool List

T10 (Indexable roughing endmill for aluminium D=32 R=2)
T11 (Carbide endmill 2 flutes D=16)
T12 (Carbide endmill 2 flutes D=8)
T13 (Carbide twist drill 8.5 mm)
T14 (Tap M10x1.5)

Notes


C - C
7°
20
M10x1,5 15t
100
Pos.
Pos.
Pos.
Pos.
70
24
51.76
Plane 2
50
Ø11
Plane 3
P - P
5
35.265°
20.41
25

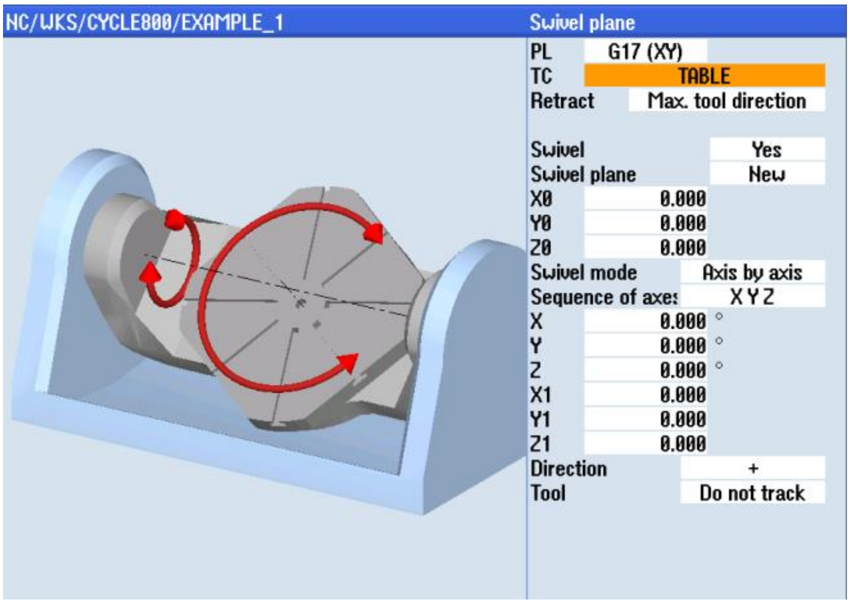

P
H
45°
Y +
X +
H
P
H - H
Plane 1
Ø 40
15
105°
Z +
Y+
50

Notes

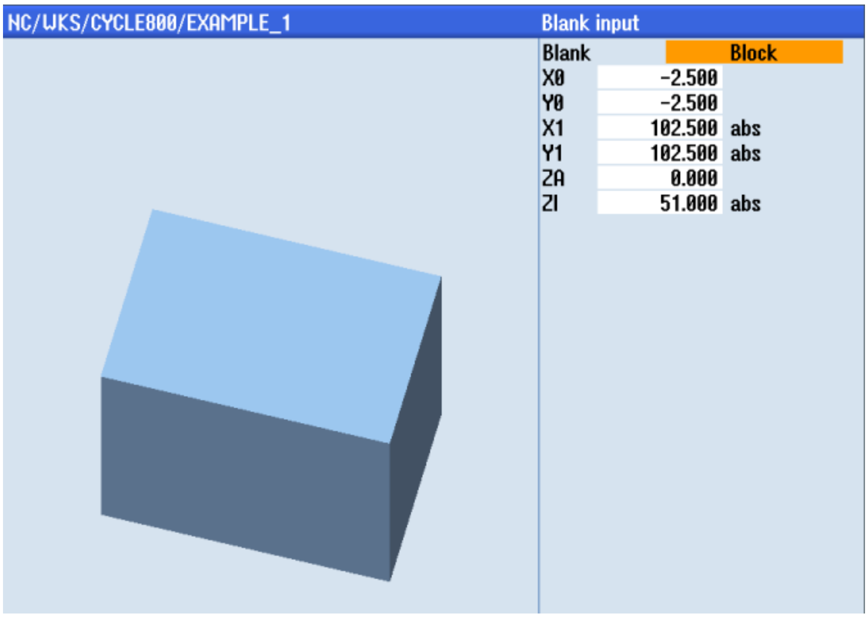
Program: EXAMPLE_1_CYCLE800.MPF

;**** 3+2 AXIS POSITIONING WITH CYCLE800 SWIVEL PLANE ****

N100 ;*** **SWIVEL INTO BASE ORIENTATION (INITIAL SETTING)** ***
N101 CYCLE800(4,"TABLE",200000,57,0,0,50,0,0,0,0,0,0,1,,1)



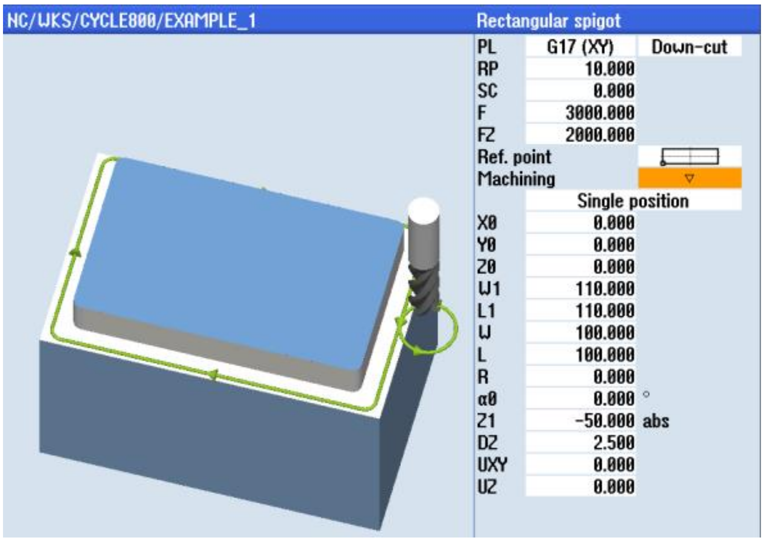
N102 ;*** **BLANK DEFINITION FOR SIMULATION** ***
N103 WORKPIECE(,,,"BOX",112,0,51,-80,-2.5,-2.5,102.5,102.5)



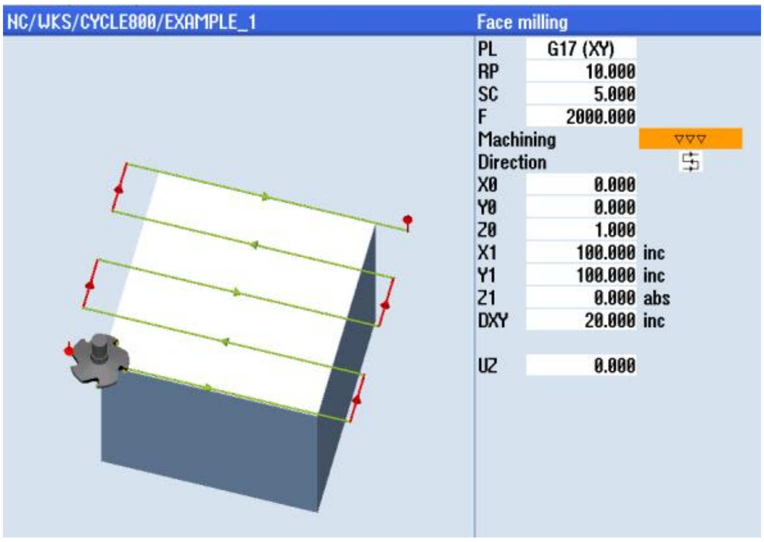
Notes

N104 **T10 D1**; T="ENDMILL_D32"
N105 M6
N106 S5000 M3
N107 G54 G0 X0 Y0 M8
N108 TRANS Z50; Shift Z0 to workpiece top surface

N109 **;***Rectangular spigot roughing 100x100*****
N110 CYCLE76(10,0,0,-50,,100,100,0,0,0,0,2.5,0,0,3000,2000,0,1,110,110,,,2100,1,2)

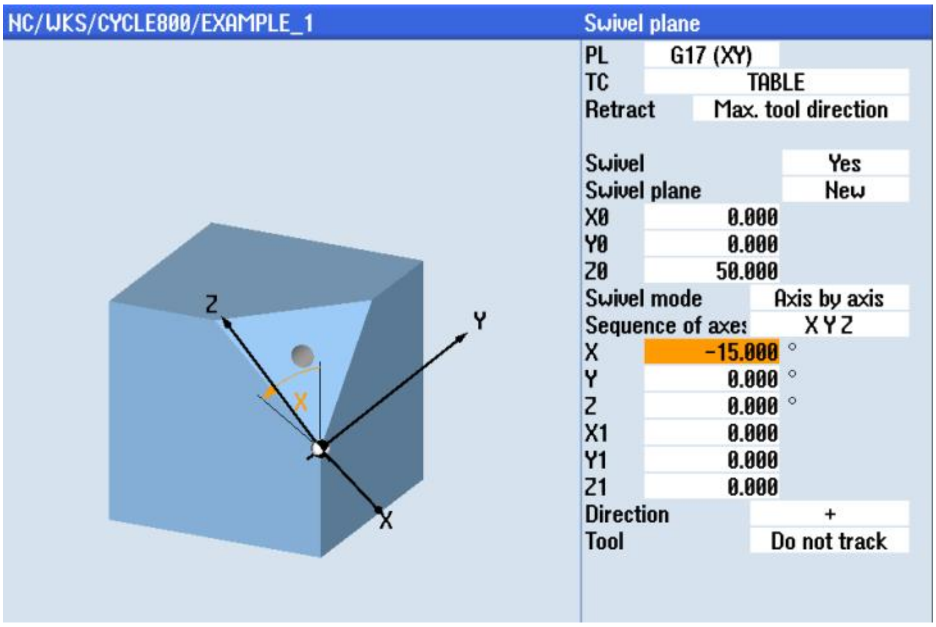


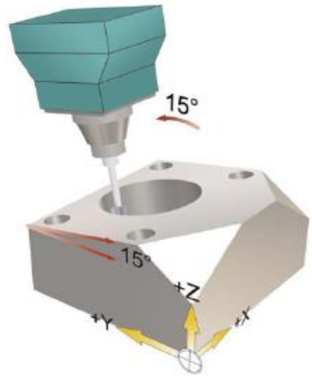
N111; ***Face milling in base orientation***
N112 CYCLE61(10,1,5,0,0,0,100,100,1,20,0,2000,32,0,1,0)



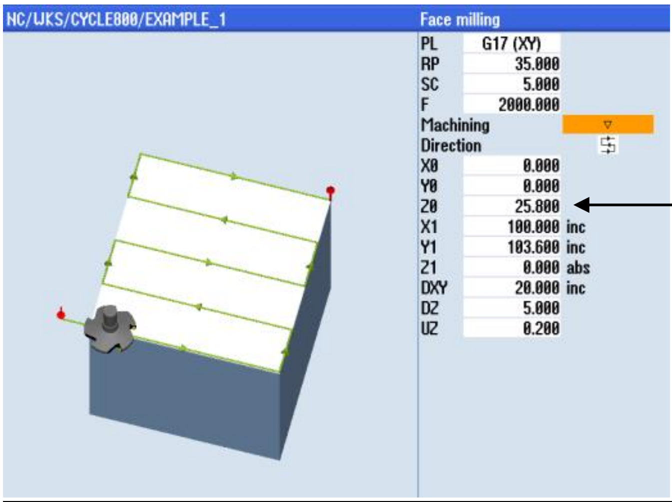
Notes

N113 ;***Swivel into Plane_1 rotation around X=-15 Grad***
N114 CYCLE800(4,"TABLE",200000,57,0,0,50,-15,0,0,0,0,0,1,,1)





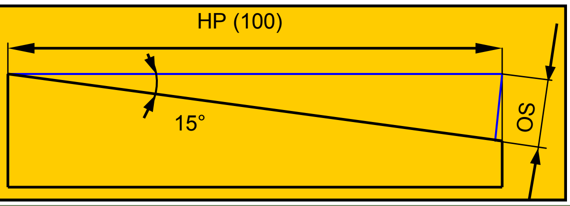
N115 ;***Face milling roughing Plane_1***
N116 CYCLE61(35,25.8,5,0,0,0,100,103.6,5,20,0.2,2000,31,0,1,0)



Must be calculated see note below

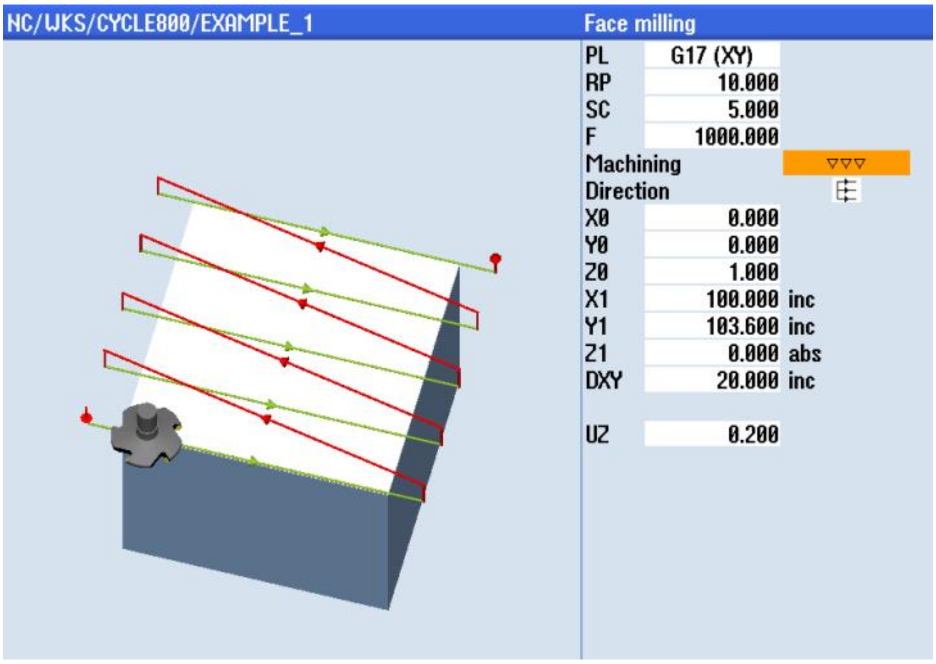
Note: (Calculation of parameter „**Z0**“)

Sin α = GK / HP   sin 15 = GK / 100
GK = sin 15 * 100
GK = **25.881**



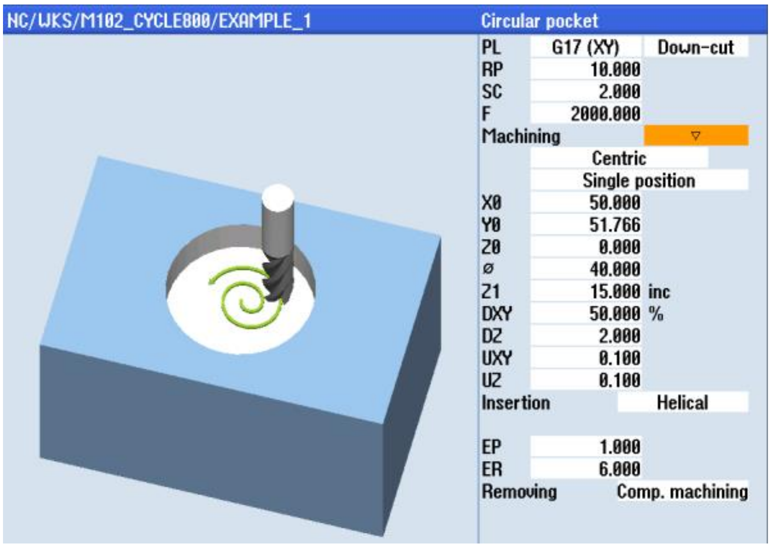
Notes

N117 ; ***Face milling finishing Plane_1***
N118 CYCLE61(10,1,5,0,0,0,100,103.6,5,20,0.2,1000,12,0,1,0)



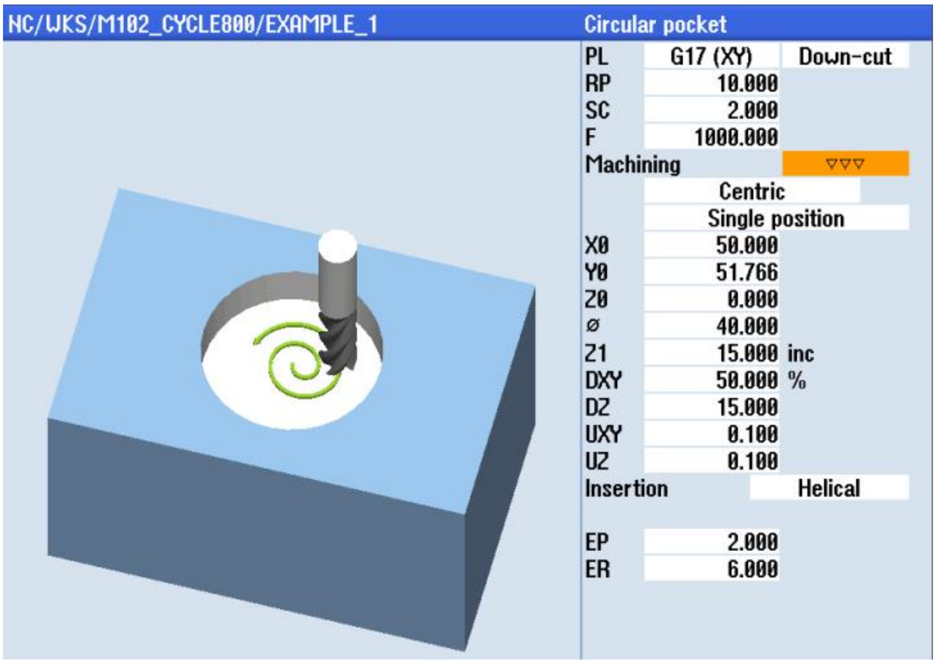
N119 **T11** D1 ; T="SC_ENDMILL_D16"
N120 M6
N121 S8000 M3
N122 G54 G0 X50 Y51.758 M8 ; Pre-position above pocket center

N123 ;*** **Circular pocket milling roughing in Plane_1*****
N124 POCKET4(10,0,2,-15,40,50,51.766,5,0.1,0.1,2000,2000,0,21,80,0,,10,2.5,0,,,10100,111,10)

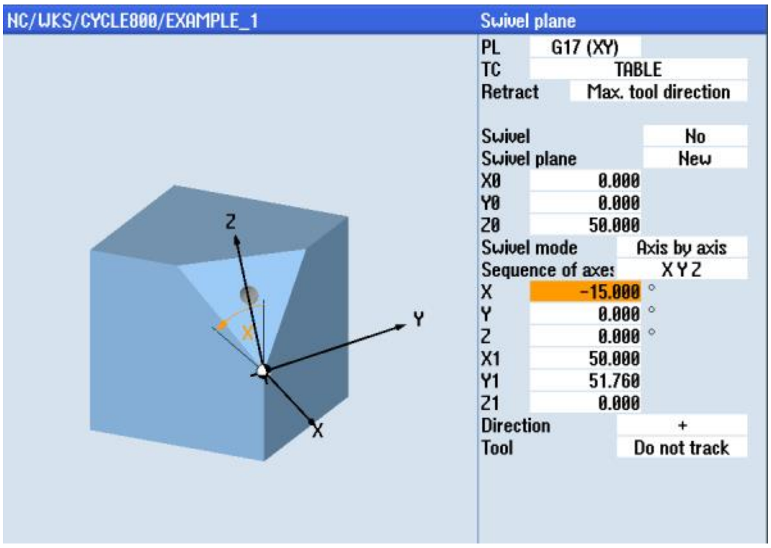


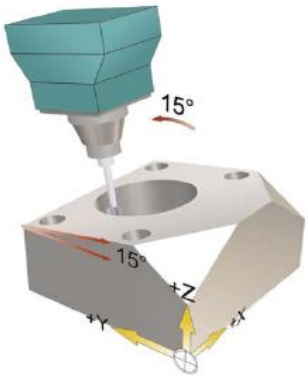
Notes

N125 ;*** **Circular pocket milling finishing in Plane_1*****
N126 POCKET4(10,0,2,-15,40,50,51.766,2.5,0.1,0.1,1000,1000,0,22,80,0,,5,2.5,0,,,10100,111,10)



N127 **T12 D1**; T=“SC_ENDMILL_D10“
N128 M6
N129 S8000 M3 F500
N130 G54 G0 X0 Y0 M8

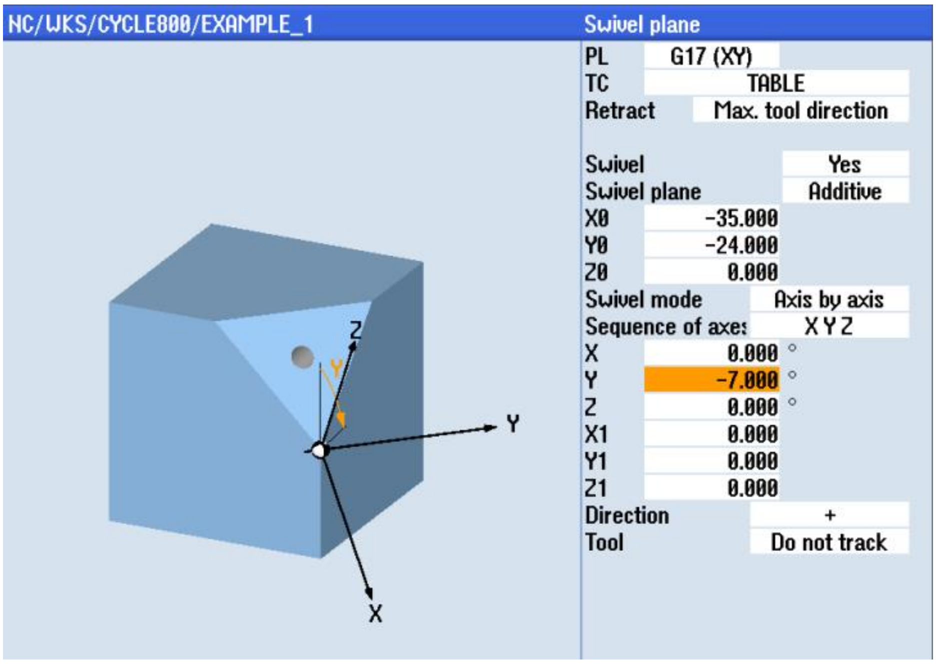


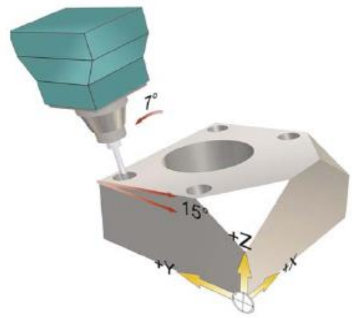


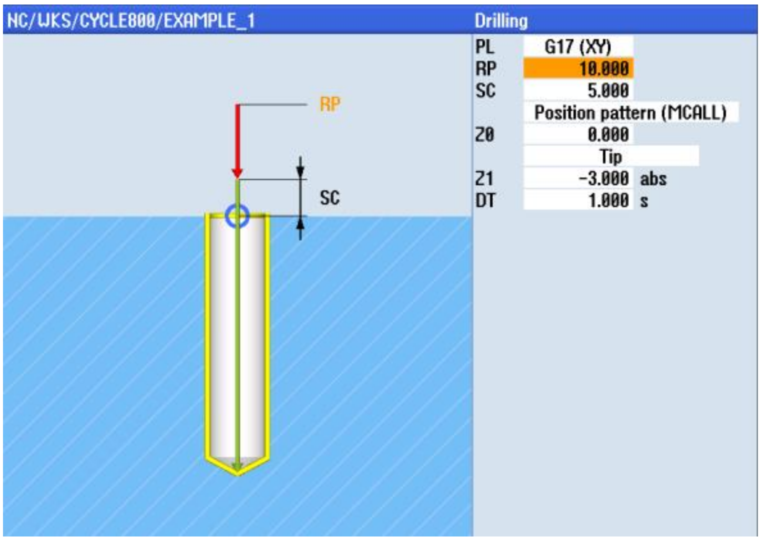
N131 ;***Calculate swivel Plane 1 rotation around X=-15 deg***
N132 CYCLE800(4,"TABLE",220000,57,0,0,50,-15,0,0,50,51.76,0,0,,1)

Notes

N133 ; ***Additive rotation around Y=-7 deg from Plane_1***
N134 CYCLE800(4,"TABLE",200001,57,-35,-24,0,0,-7,0,0,0,0,1,,1)





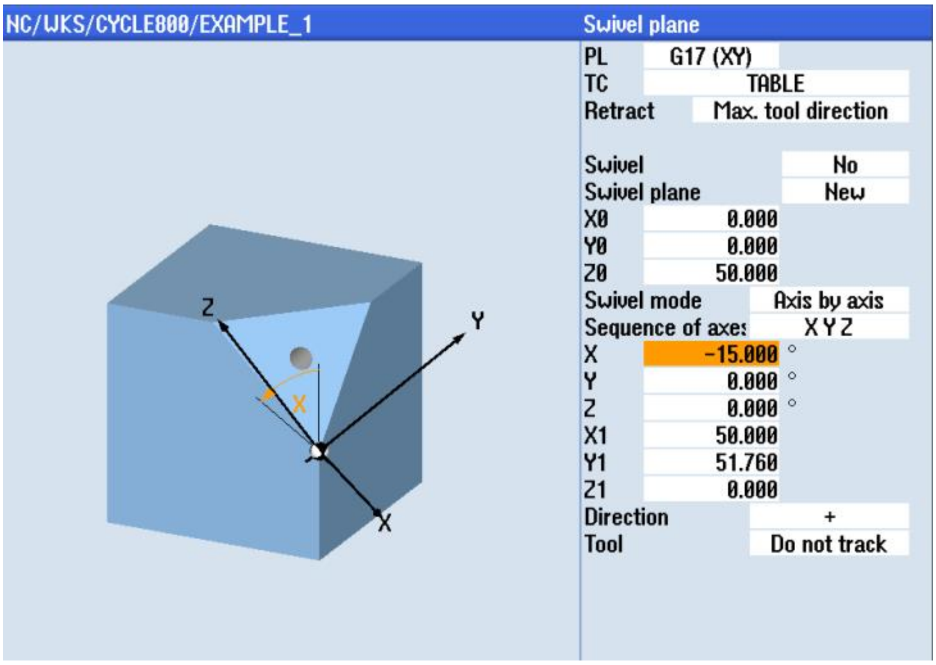


N135 ;***Plunge Pilot Hole***
N136 MCALL CYCLE82(10,0,5,-3,,1,0,1,12)

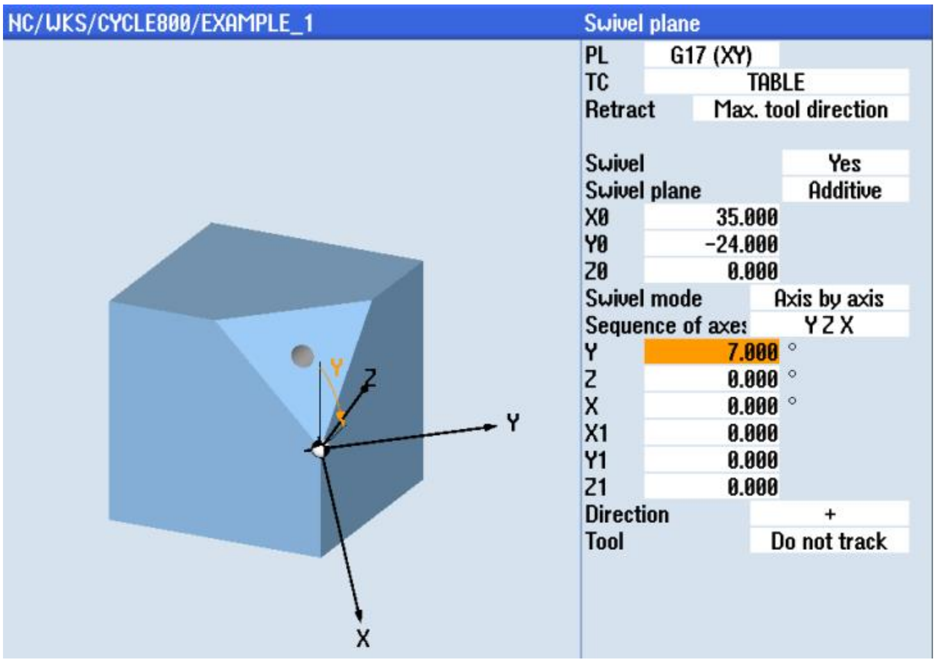
N137 G0 X0 Y0
N138 G0 X0 Y60

Notes

N140 ;***Calculate Swivel Plane_1 rotation around X=-15 deg***
N141 CYCLE800(4,"TABLE",220000,57,0,0,50,-15,0,0,50,51.76,0,0,,1)

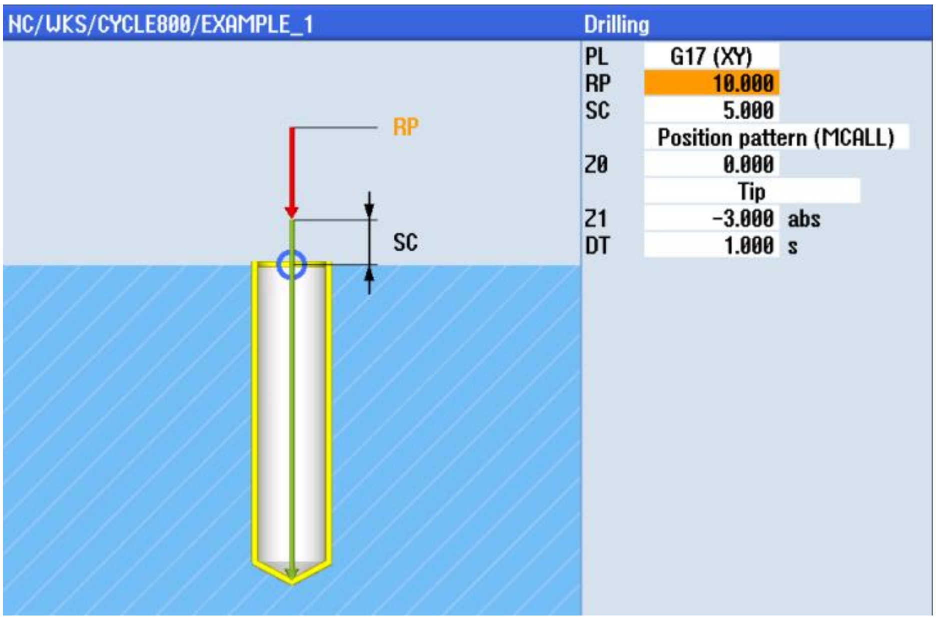


N142 ; ***Additive rotation around Y=+7 deg from Plane_1***
N143 CYCLE800(4,"TABLE",200001,30,35,-24,0,7,0,0,0,0,0,1,,1)



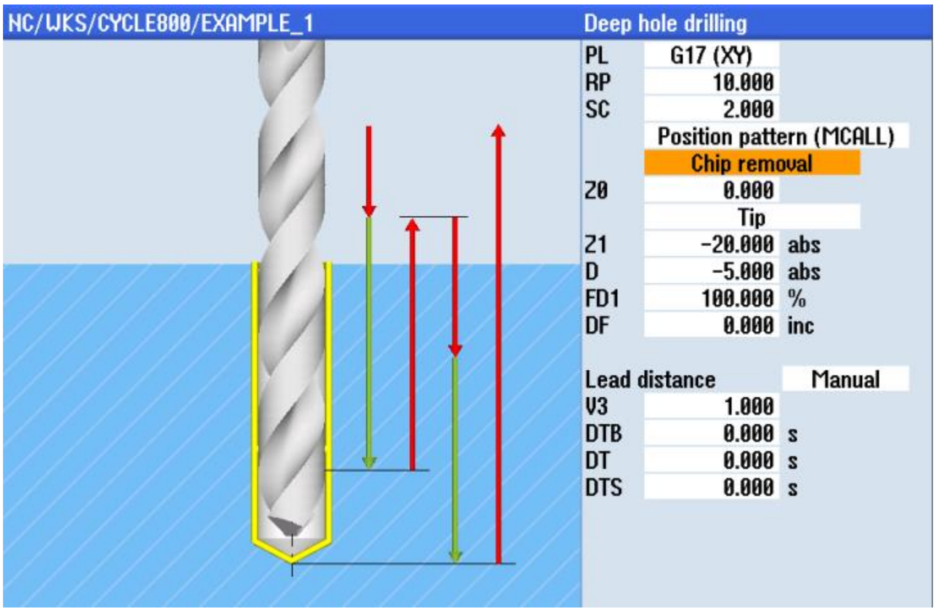
Notes

N144 ;***Plunge Pilot Hole***
N145 MCALL CYCLE82(10,0,5,-3,,1,0,1,12)



N146 G0 X0 Y0
N147 G0 X0 Y60
N148 MCALL

N149 **T13 D1**; T=“SC_DRILL_D8.5“
N150 M6
N151 S4000 M3 F500
N152 G54 G0 X0 Y0 M8
N153 ;***Deep Hole Drilling D=8.5 ***

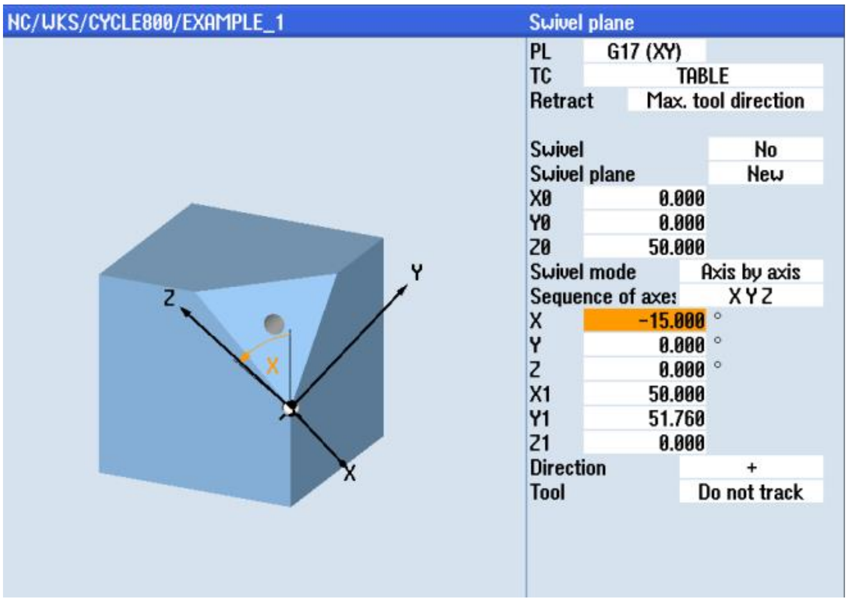


N154 MCALL CYCLE83(10,0,2,-20,,-

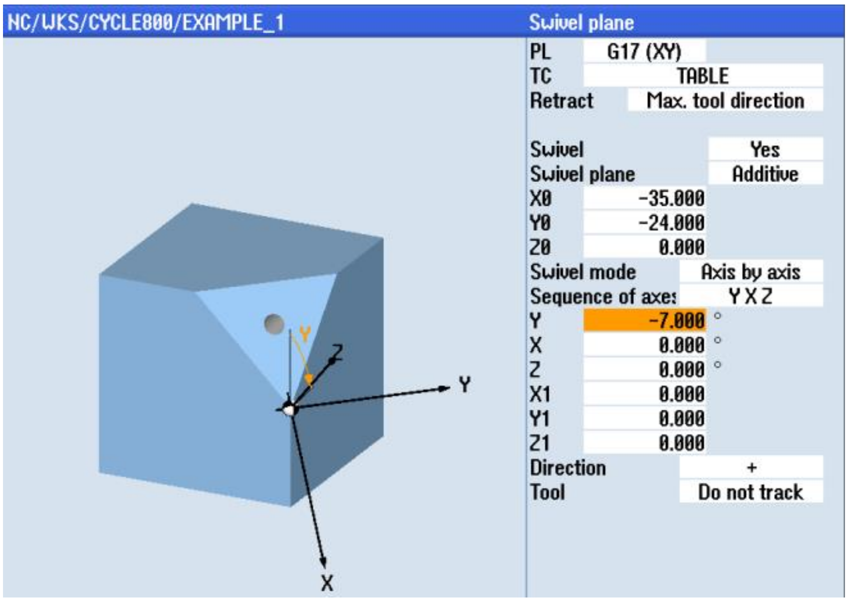
Notes

N155 G0 X0 Y0
N156 G0 X0 Y60
N157 MCALL

N158 ;***Calculate Swivel Plane_1 rotation around X=-15 deg***
N159 CYCLE800(4,"TABLE",220000,57,0,0,50,-15,0,0,50,51.76,0,0,,1)

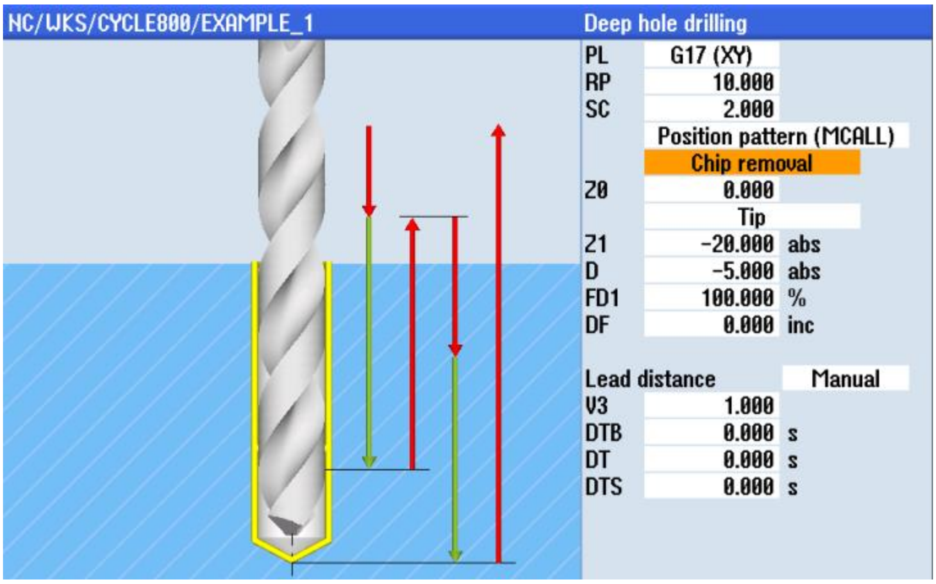


N160 ;***Additive rotation from Plane_1 around Y=-7 deg***
N161 CYCLE800(4,"TABLE",200001,30,-35,-24,0,-7,0,0,0,0,0,1,,1)



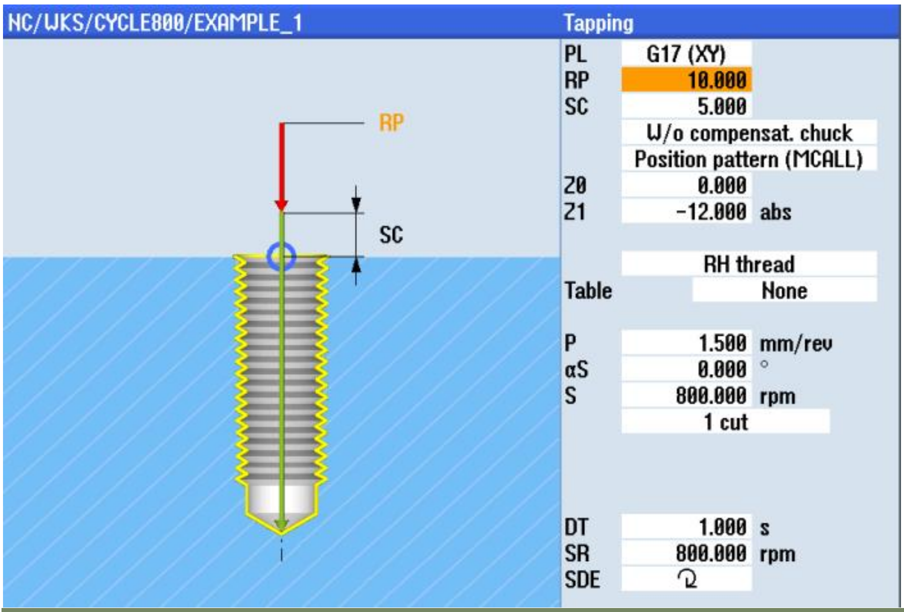
Notes

N162 ;***Deep hole drilling D=8.5***
N163 MCALL CYCLE83(10,0,2,-20,,-5,,0,0,0,100,1,0,5,,0,1,0,1,12121112)



N164 G0 X0 Y0
N165 G0 X0 Y60
N166 MCALL

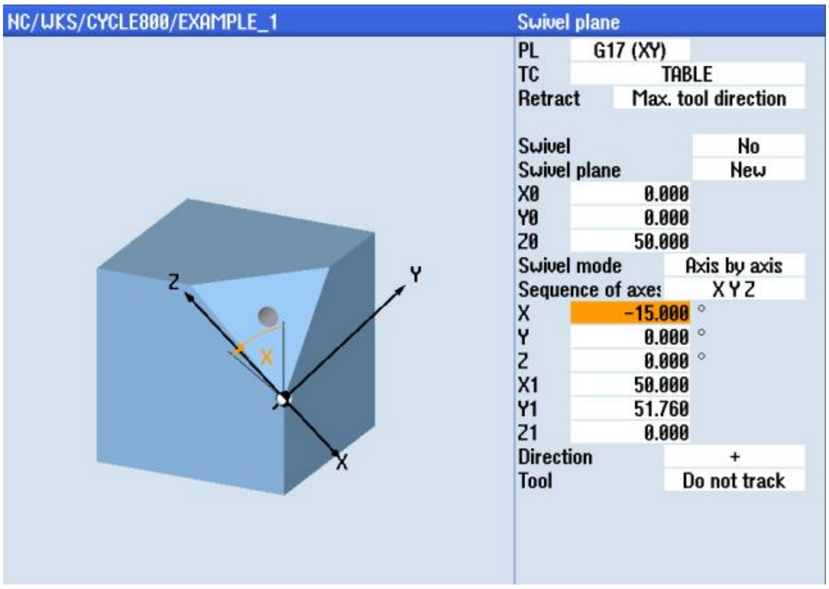
N167 **T14 D1**; T="TAP_M10"
N168 M6
N169 S800 M3
N170 G54 G0 X0 Y0 M8
N171 ;***Tapping M10***
N172 MCALL CYCLE84(10,0,5,-12,,1,3,,1.5,0,800,800,0,1,0,0,,1.4,,,,,1001,1001002)



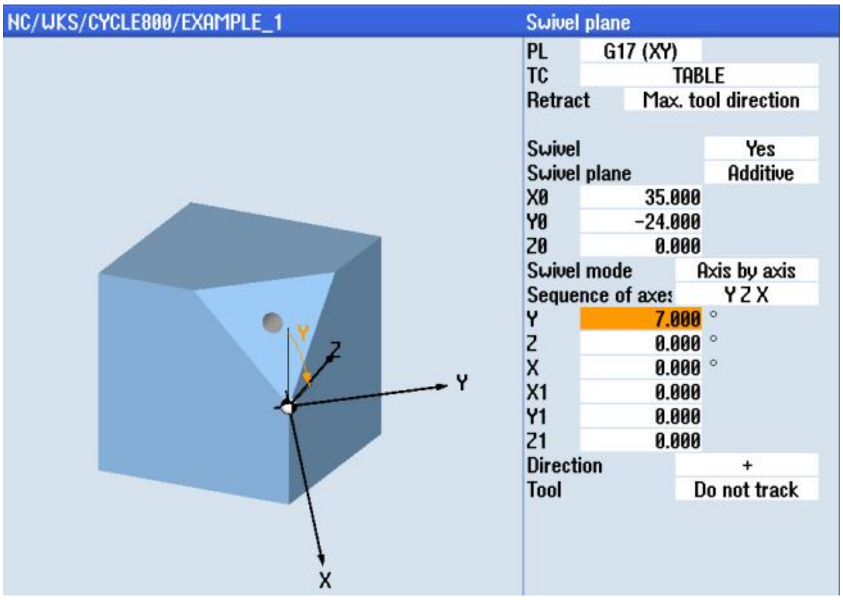
Notes

N173 G0 X0 Y0
N174 G0 X0 Y60
N175 MCALL

N176 ;***Calculate Swivel Plane_1 rotation around X=-15 deg***
N177 CYCLE800(4,"TABLE",220000,57,0,0,50,-15,0,0,50,51.76,0,0,,1)

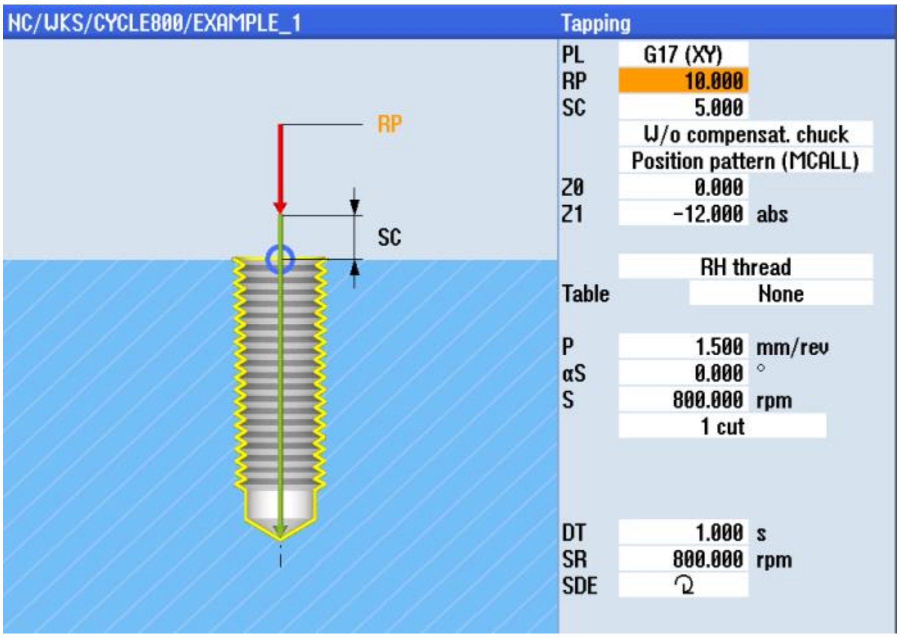


N178 ;*** Additive rotation from Plane1 around Y=+7 deg***
N179 CYCLE800(4,"TABLE",200001,30,35,-24,0,7,0,0,0,0,0,1,,1)



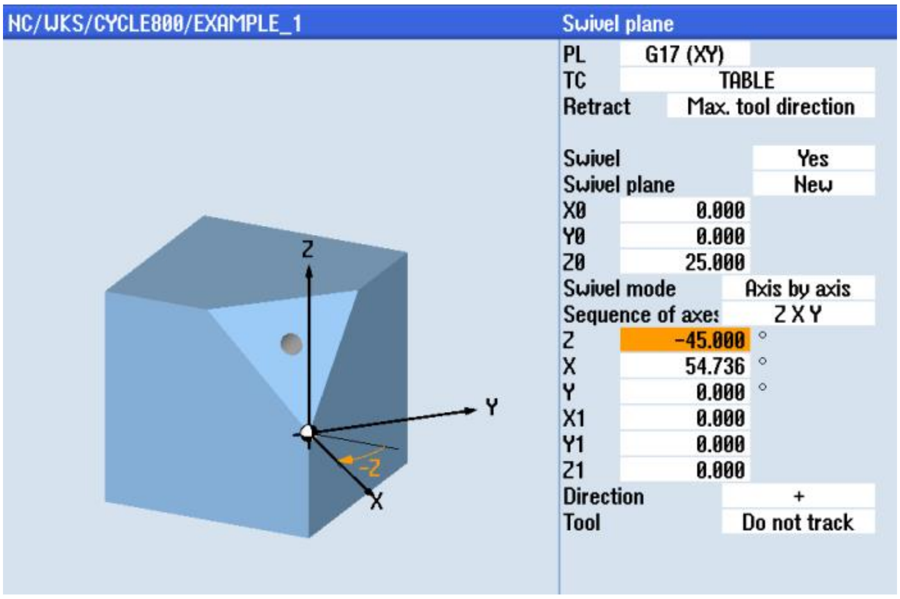
Notes

N180 ; ***Tapping M10***
N181 MCALL CYCLE84(10,0,5,-12,,1,3,,1.5,0,800,800,0,1,0,0,,1.4,,,,,1001,1001002)



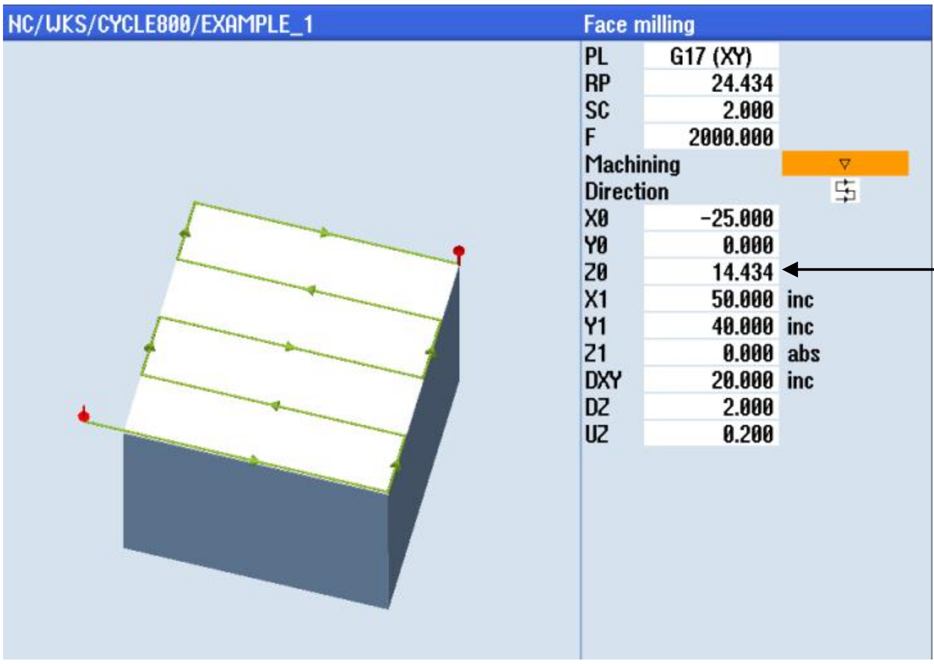
N182 G0 X0 Y0
N183 G0 X0 Y60
N184 MCALL

N185 **T10** D1; T="ENDMILL_D32"
N186 M6
N187 S8000 M3
N188 G54 G0 X0 Y0 M8
N189 ;***Swivel into Plane_2 rotation around Z=-45 X=54.736 deg***
N190 CYCLE800(1,"TABLE",200000,39,0,0,25,-45,54.736,0,0,0,0,1,)



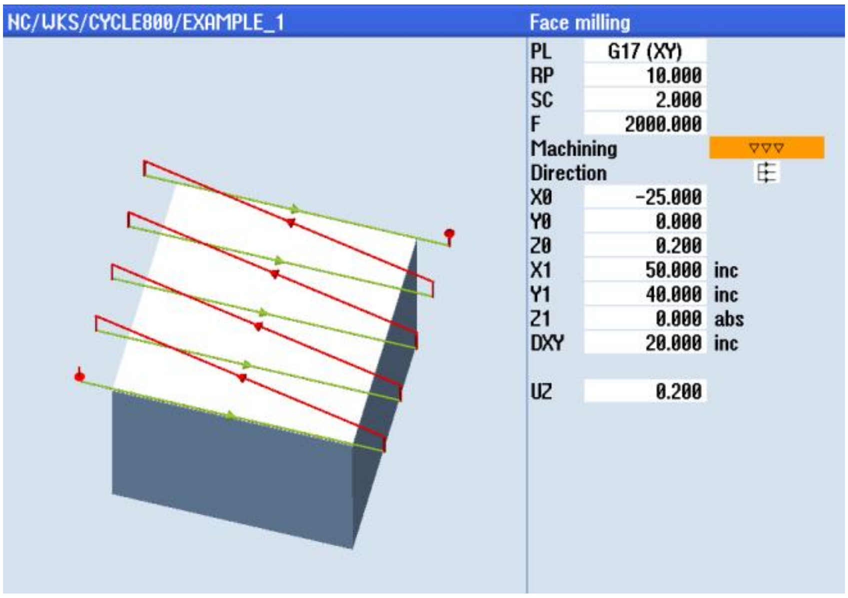
Notes

N191 ; ***Face milling roughing in Plane 2 ***
N192 CYCLE61(24.434,14.434,2,0,-25,0,50,40,2,20,0.2,2000,31,0,1,0)



Must be calculated see note below

N193 ; ***Face milling finishing in Plane 2 ***
N194 CYCLE61(10,0.2,2,0,-25,0,50,40,2,20,0.2,2000,12,0,1,0)

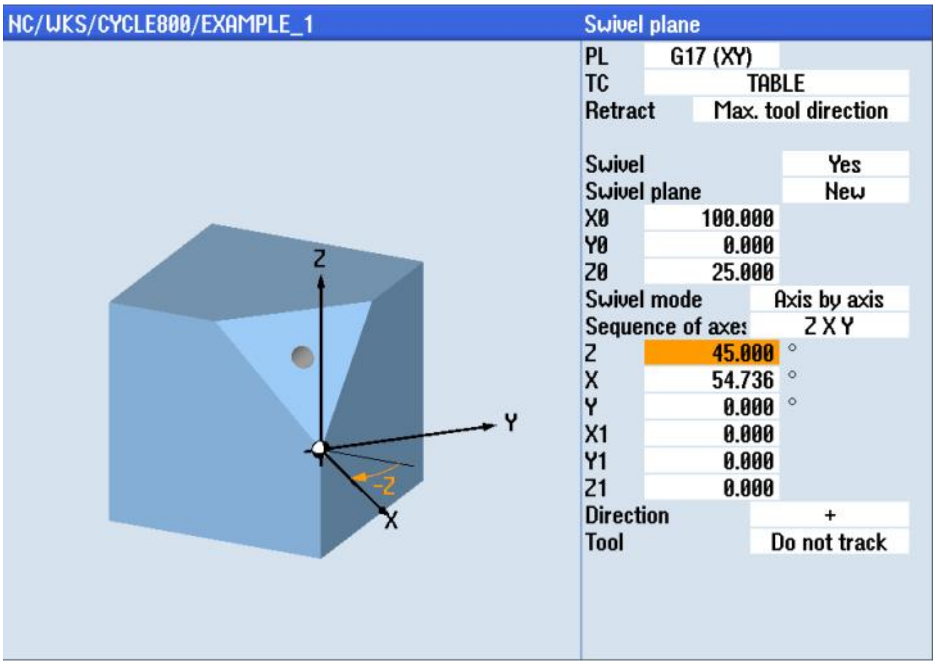


Note: (Calculation for parameter „**Z0**“)

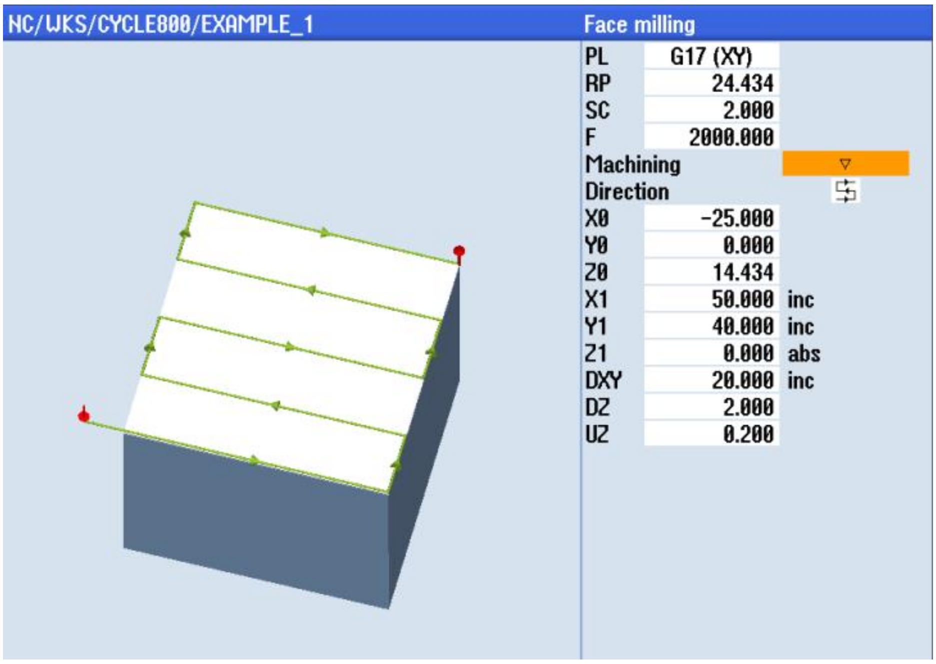
sin 35.264 = GK / 25 ⟶ OS = sin 35.264 * 25 ⟶ GK = **14.4336**

Notes

N195 ;*** **Swivel into Plane 3 rotation around Z=45 X=54.736 deg*****
N196 CYCLE800(4,"TABLE",200000,39,100,0,25,45,54.736,0,0,0,0,1,,1)



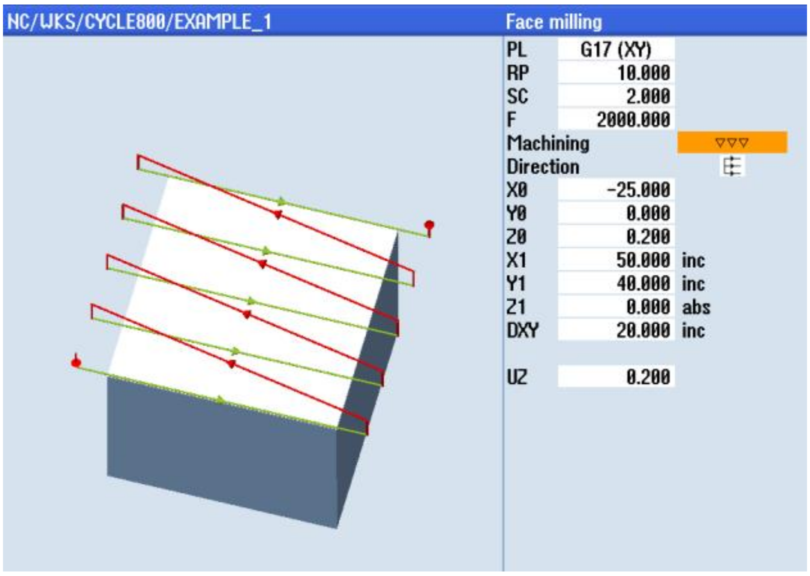
N197 ; ***Face milling roughing Plane 3 *****
N198 CYCLE61(24.434,14.434,2,0,-25,0,50,40,2,20,0.2,2000,31,0,1,0)
N197 M9



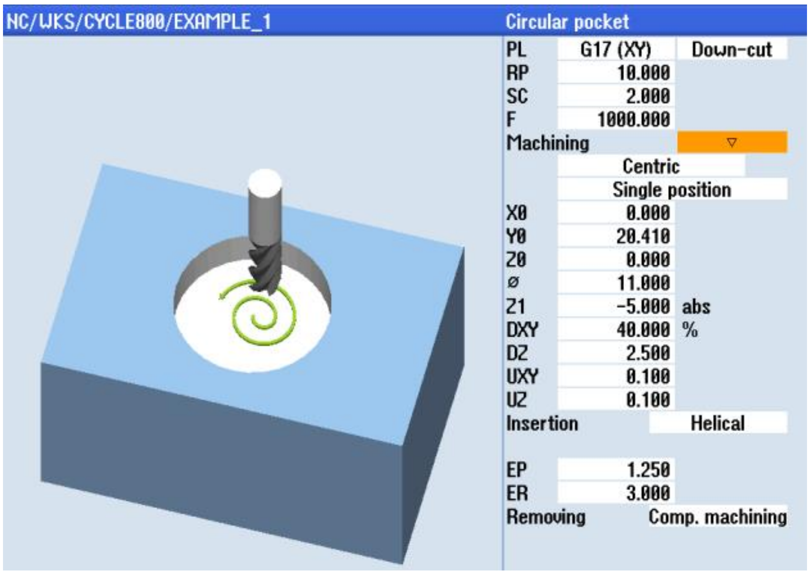
Notes

N199 ; ***Face milling finishing Plane 3***
N200 CYCLE61(10,0.2,2,0,-25,0,50,40,2,20,0.2,2000,12,0,1,0)

N201 **T12** D1; T=“SC_ENDMILL_D10“
N202 M6
N203 S8000 M3
N204 G54 G0 X0 Y0 M8

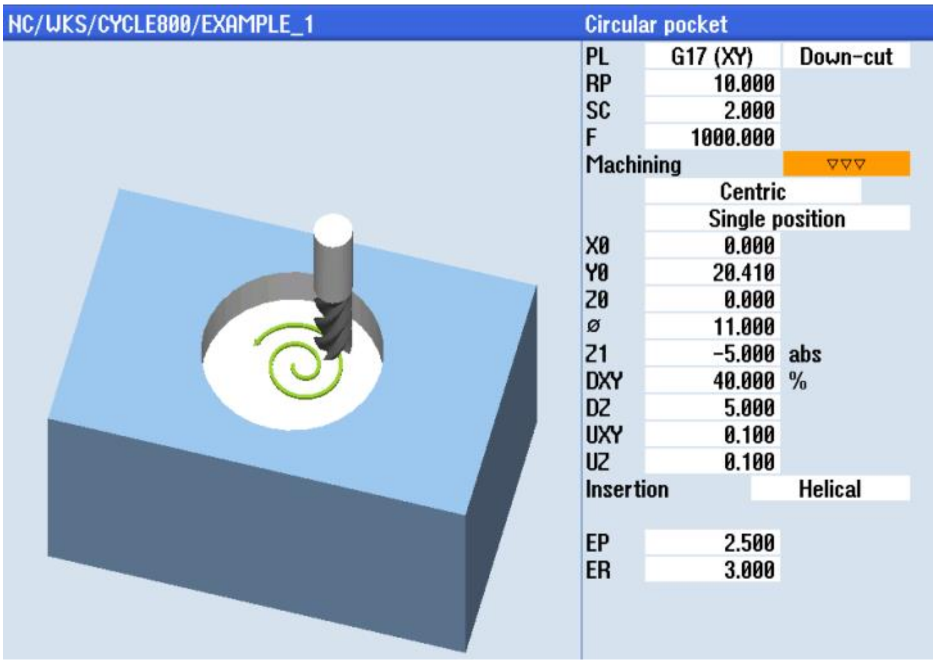


N205 ;***Circular pocket roughing Plane 3***
N206 POCKET4(10,0,2,-5,11,0,20.41,2.5,0.1,0.1,1000,1000,0,21,6,0,,3,1.25,0,,,10100,111,0)

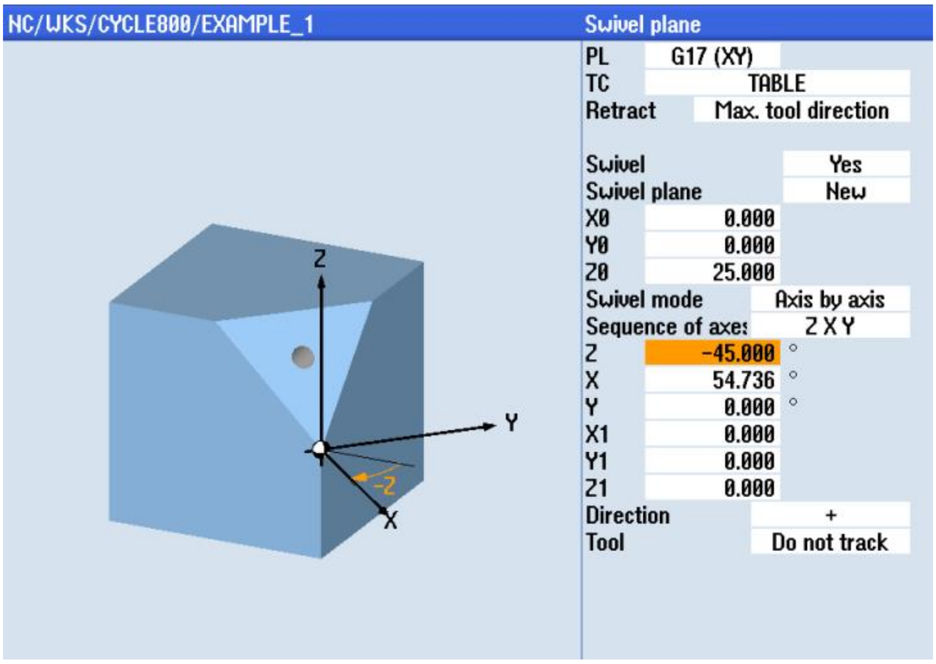


Notes

N207 ;***Circular pocket finishing Plane 3***
N208 POCKET4(10,0,2,-5,11,0,20.41,2.5,0.1,0.1,1000,1000,0,22,6,0,,3,2.5,0,,,10100,111,0)

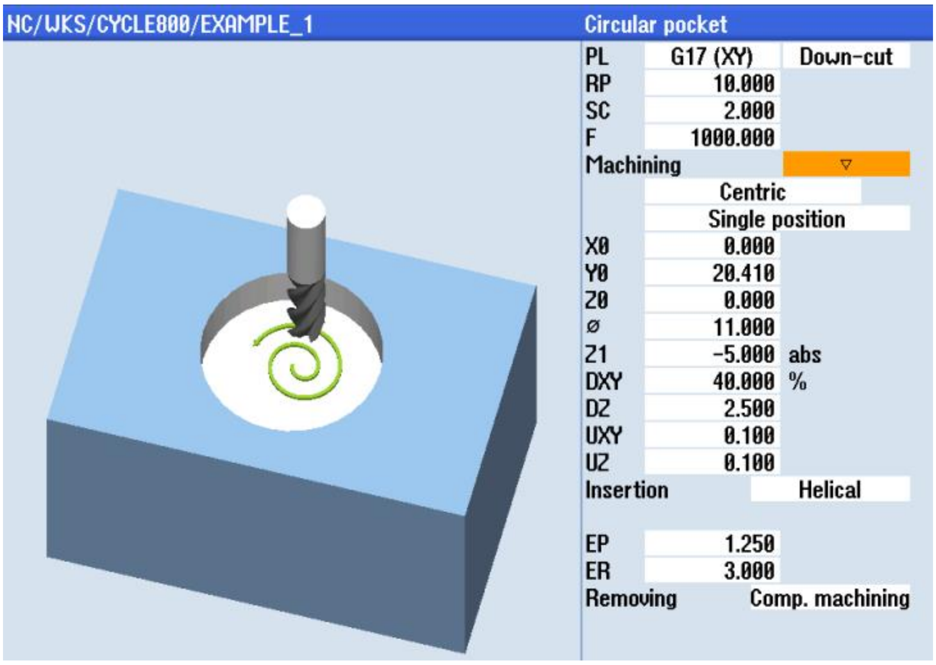


N209 ;*** **Swivel into Plane 2 rotation around Z=-45 X=54.736 deg*****
N210 CYCLE800(1,"TABLE",200000,39,0,0,25,-45,54.736,0,0,0,0,1,)

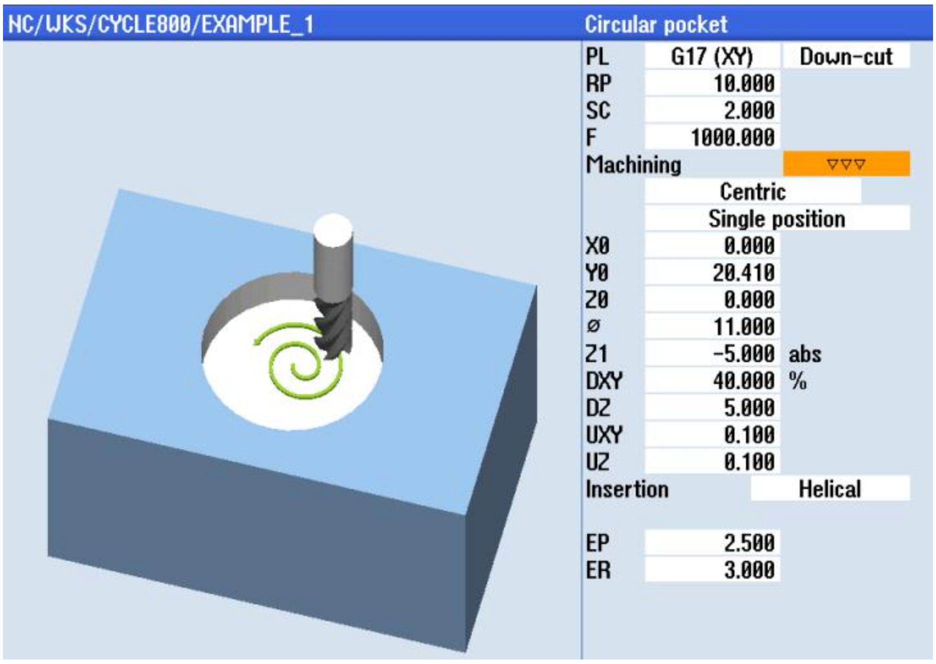


Notes

N211 ;***Circular pocket roughing Plane 2***
N212 POCKET4(10,0,2,-5,11,0,20.41,2.5,0.1,0.1,1000,1000,0,21,6,0,,3,1.25,0,,,10100,111,0)

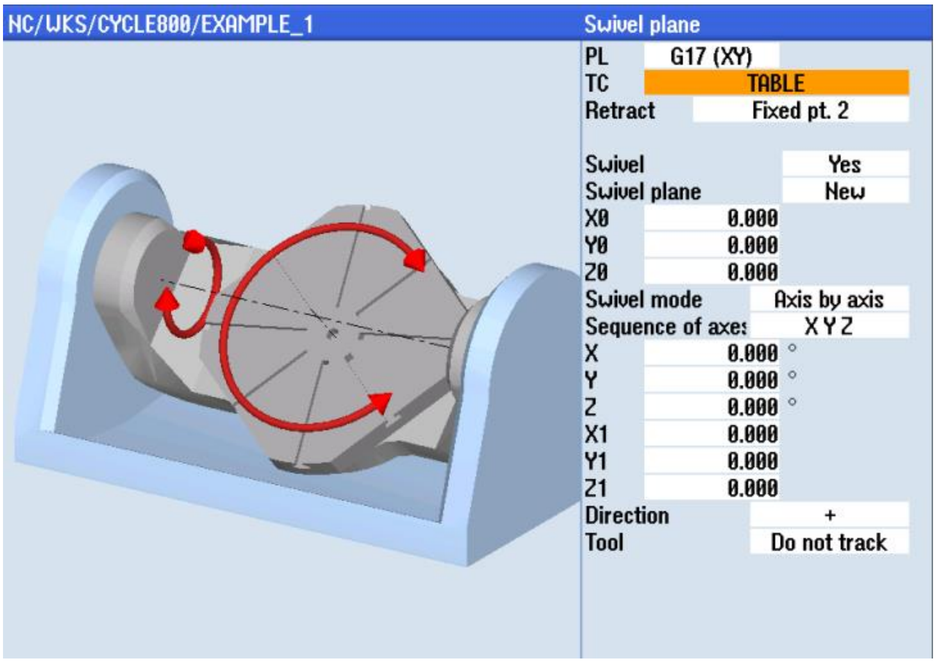


N213 ;***Circular pocket finishing Plane 3***
N214 POCKET4(10,0,2,-5,11,0,20.41,2.5,0.1,0.1,1000,1000,0,22,6,0,,3,2.5,0,,,10100,111,0)

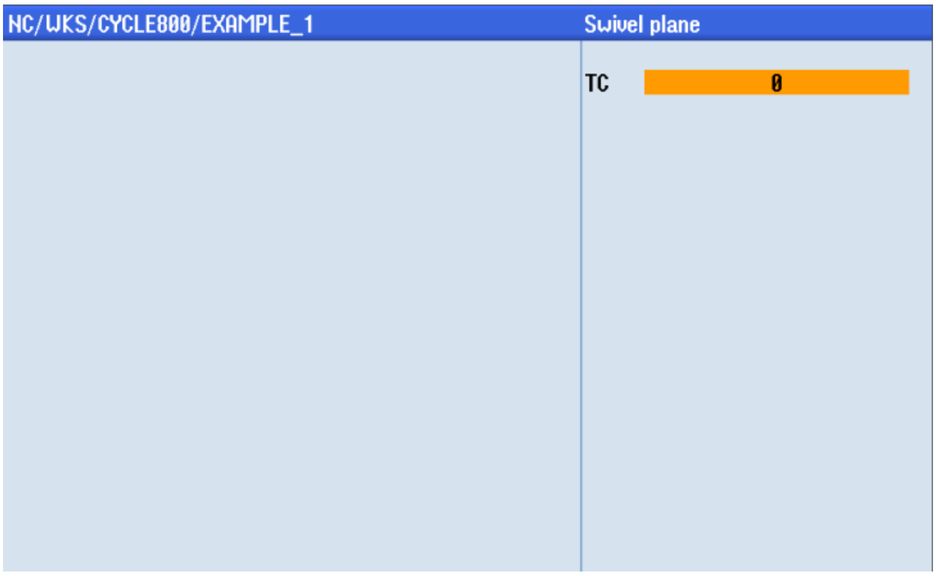


Notes

N215 ;*** **Swivel into basic orientation (initial setting)*****
N216 CYCLE800(4,"TABLE",200000,57,0,0,0,0,0,0,0,0,0,1,,1)



N217 ;***Deselect swivel cycle***
N218 CYCLE800(0,"0",200000,57,0,0,50,0,0,0,0,0,0,0,1,,1)
N219 M30; **END OF PROGRAM**
N211 M30



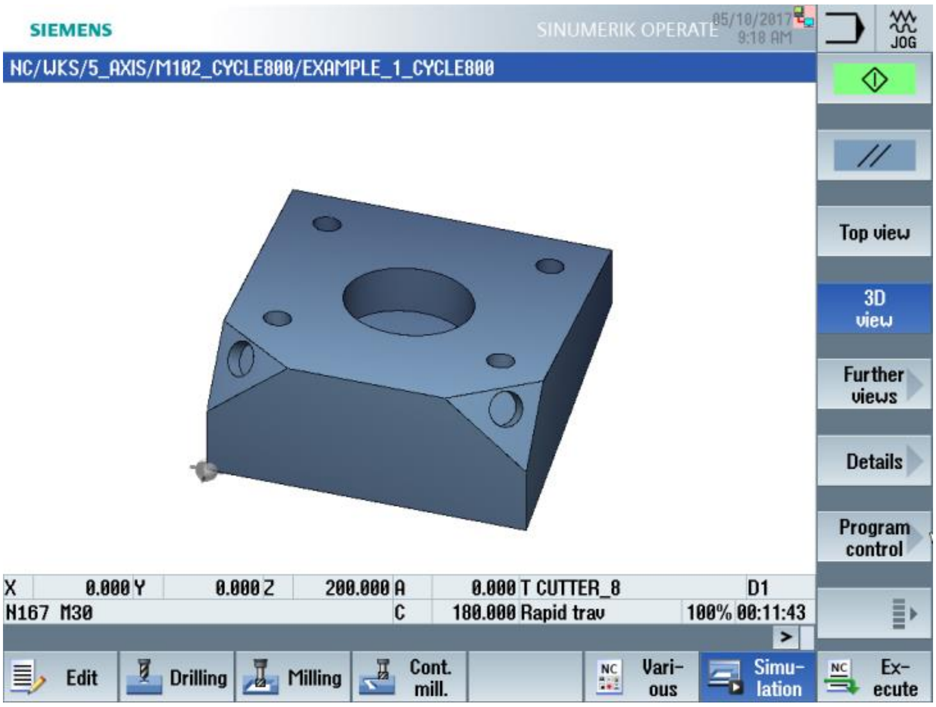
Note:

Alternatively you can deselect CYCLE800 with manual input of **CYCLE800()**.

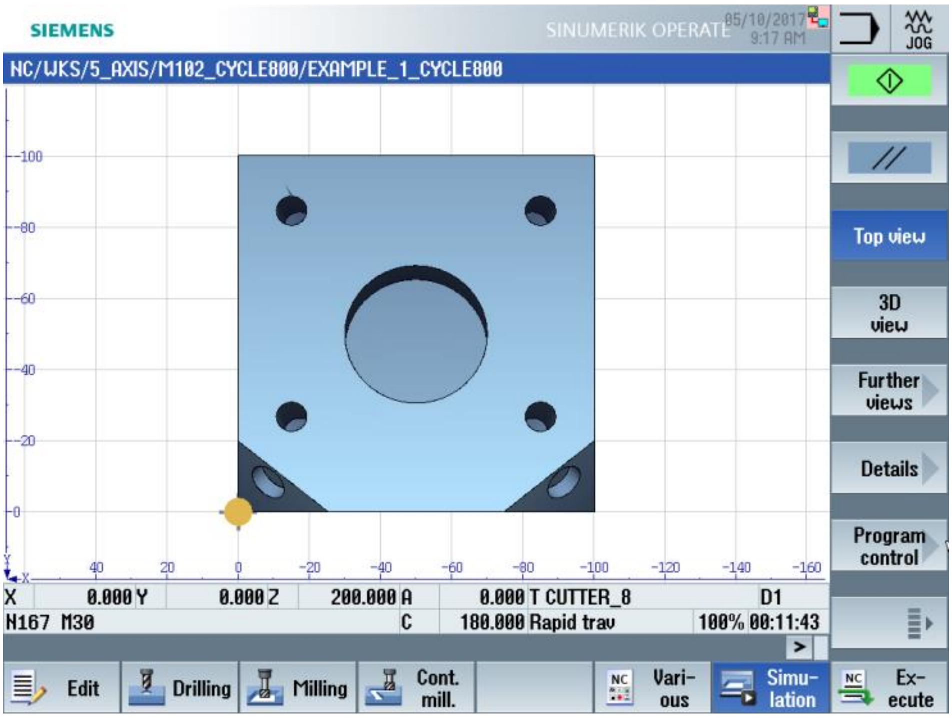
Notes

Simulation PROGRAM

3D view



Top view



Notes

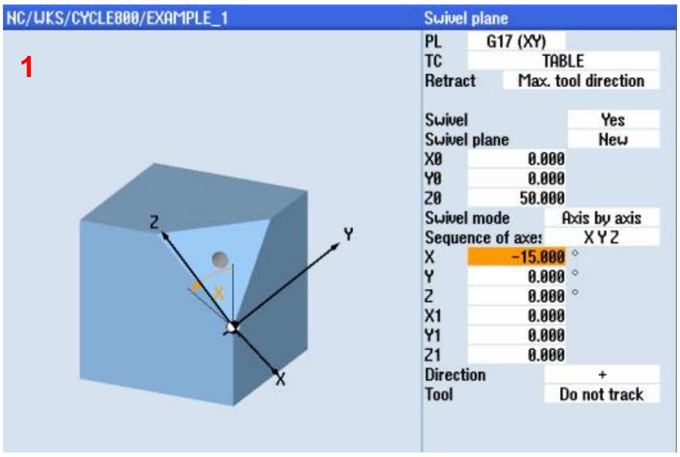
CYCLE800 “Approach tool” in programGUIDE G code

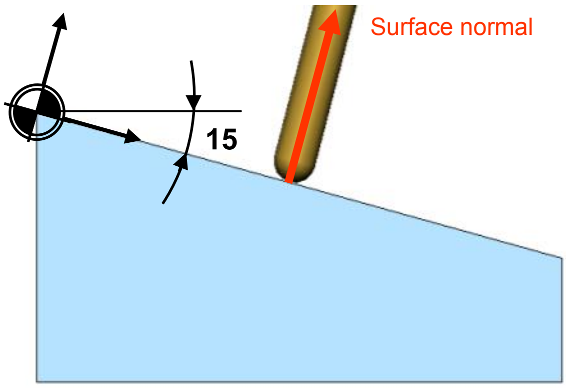
Explanation of the Cycle

When milling with ball nose cutters, it can make technological sense to set the tool at an angle to the surface normal vector with CYCLE800 "**Approach tool**", to improve cutting conditions by means of avoiding to cut with the tool center (zero cutting speed).

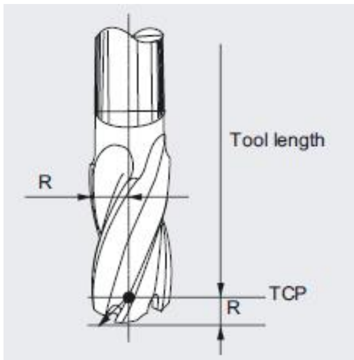
If a program with one or several tool setting angles is to be milled, a "**new**" swivel plane (**1**) must always be set before the subsequent “**Approach tool**” cycle call.

After "**Swivel plane**", the tool orientation is always vertical (normal) to the machining plane (**2**).



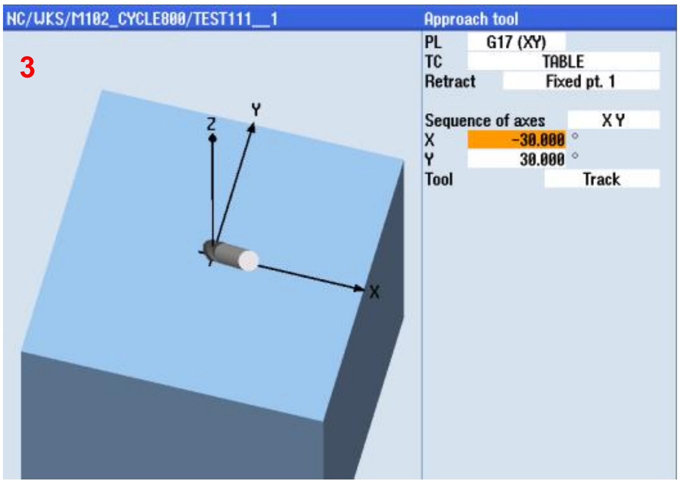


With help of CYCLE800 “**Approach tool**” (**3**) it is possible to change the tool setting angle to the machining plane in various ways. Prerequisite for this, is programming the tool path to the sphere center of the ball mill.

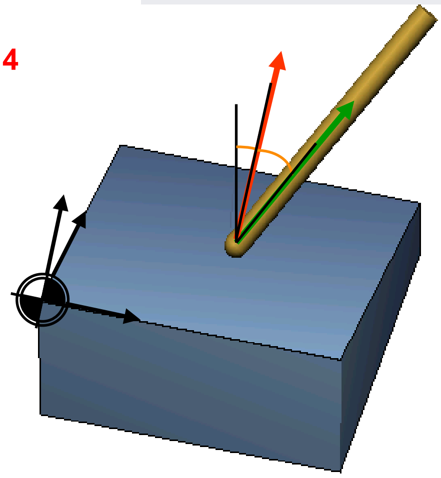


The tool length of the Ball nose cutter is therefore to be set to the sphere center (TCP = Tool Center Point).

The orientation setting angle of the tool is always “**additive**” and acts in reference to the active programmed swivel plane (surface normal) (**4**).

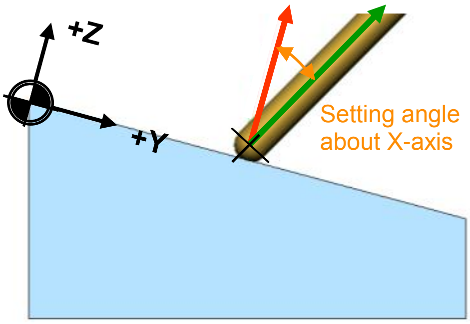


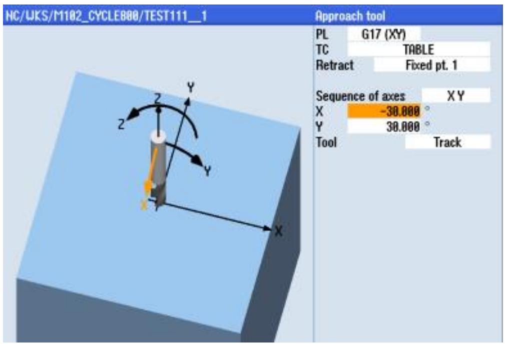
4

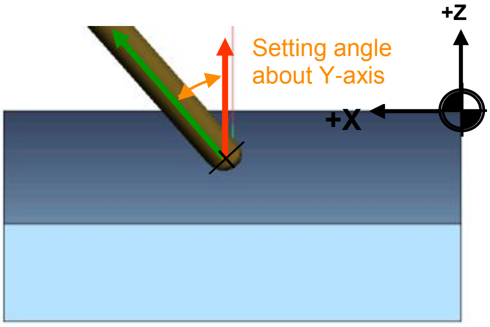


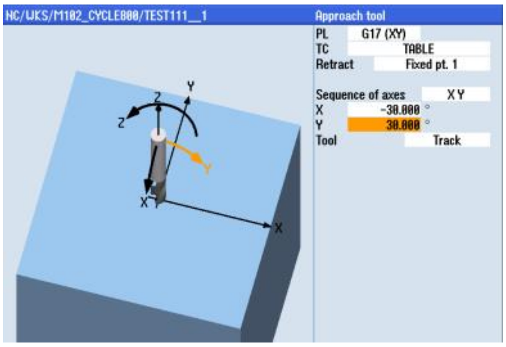
Notes

In the swivel cycle, the setting angle is generated by an axis rotation (max. +/- 90 degrees) in the active swivel plane. When setting the tool orientation with "**Approach tool**", only rotations are displayed in the swivel cycle input screen form. The tool setting angle can be applied about only one axis or two axes depending on the application requirement. The user can select the rotation sequence.

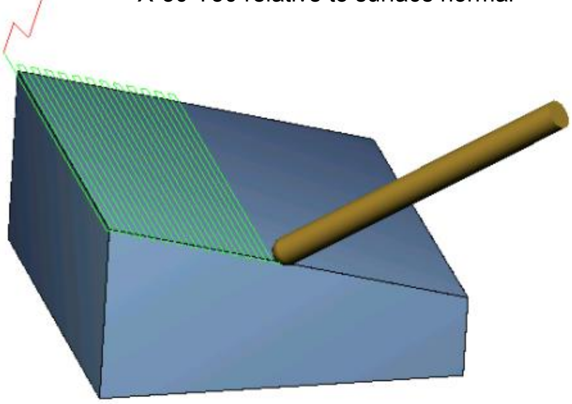


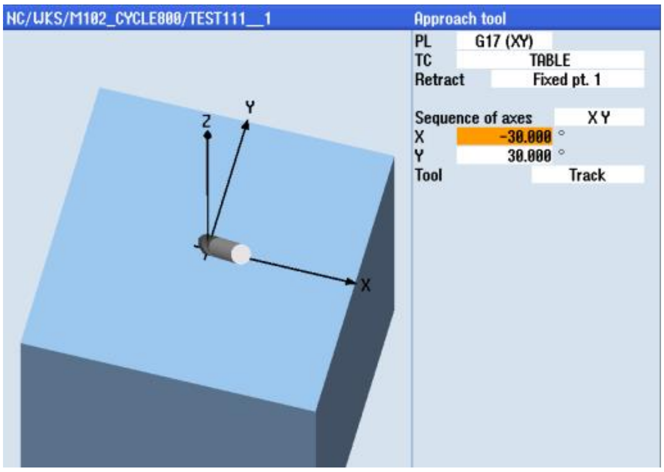






Surface machining with setting angle X-30 Y30 relative to surface normal





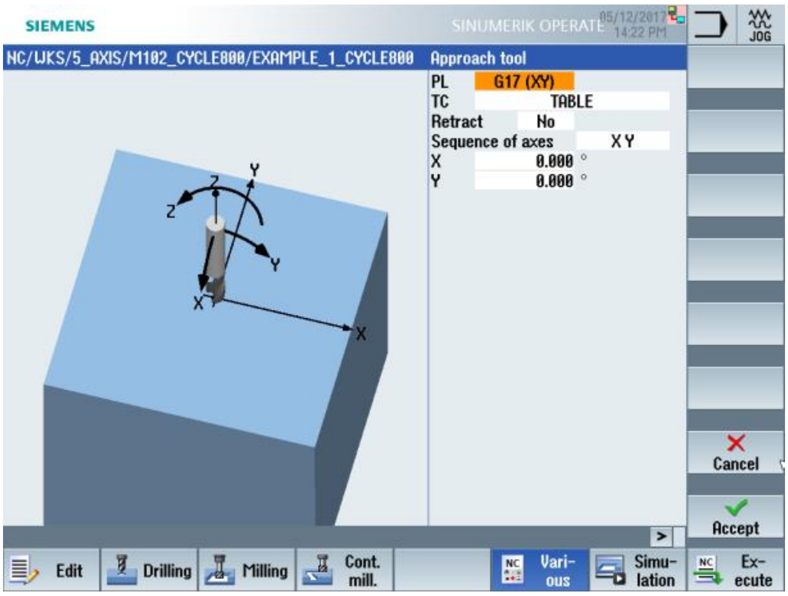
Note:

With CYCLE800 **„Approach tool“** the coordinate system (WCS) is not being swivelled, the effect lies only in a change of the tool setting angle relative to the surface normal.

Notes

Programming procedure

Press the following softkeys to open the Cycle “**Approach tool**”:



► Set a “new” swivel plane (surface normal) as reference for the tool setting angle.
► Set the tool setting angle in reference to the surface normal with CYCLE800 “**Approach tool**”.
► Program machining as usual in the X/Y plane for G17.
► After finishing the operation swivel the coordinate system back to its initial position.

- **PL:**
  Selection of the machining plane for reference of rotation (the sequence of axes changes accordingly with change of the machining plane).
- **TC:**
  Select the name of the swivel data record (e.g. “TABLE” kinematic type P)
- **Retract:**
  Select the retraction mode before swivelling (settable in the swivel data record).
- **Sequence of axes:**
  Select the order in which the axes change tool orientation
- **X, Y:**
  Enter the angle of rotation for about each geometry axis in reference to the active swivel plane. (G17 X and Y)
- **Tool:**
  Select weather the tool tip is tracked or not during positioning of rotary axes
  (can be enabled in the swivel data record).

Notes

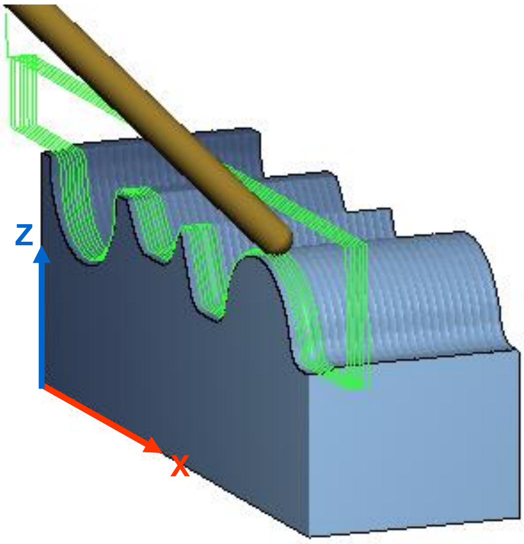
Description of parameters in cycle input mask

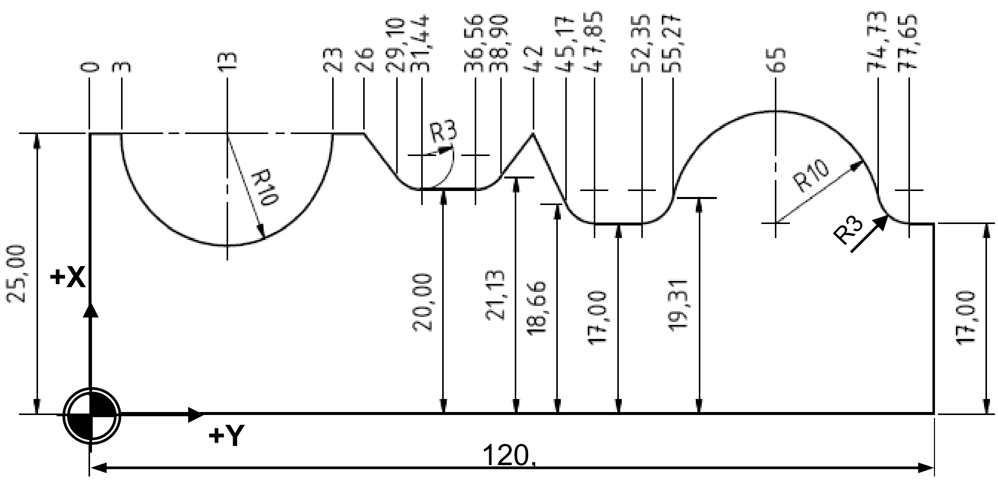
Parameter	Description
PL:	**Machining plane** G17 = (XY) Tool axis Z G18 = (ZX) Tool axis Y G19 = (YZ) Tool axis X Blank = the last programmed machining plane is valid Selection of the machining plane can be enabled in the channel specific machine data ***MD52005 $MCS_DISP_PLANE_MILL*** (0 = Plane selection active in cycle input mask)
TC	Name of the swivel data record (orientable tool carrier TCARR)
Retract	Selection of the retraction variant before changing the tool setting angle (settable in the area “Setup” of the swivel data record) ■ **No:** No retraction before swiveling ■ **Z:** Retraction of machine axis Z ■ **Z,X,Y:** Move machining axis to retraction position before swiveling ■ **Max. tool direction:** Maximum retraction in tool direction (up to software end position)

Parameter	Description
Retract	**Inc. tool direction:** Retraction incremental in tool direction e.g. 100 mm ■ **ZR (for G17):** Retraction path (this field only shows for incremental retraction in tool direction e.g. 100 mm) **Note!** all retraction variants are executed in rapid traverse.
Sequence of axes:	Order in which axes are rotated. Rotation around maximum of 2 axes in the WCS XY or XZ or YX or YZ or ZX or ZY
X	Rotation around X-axis in degrees (WCS for G17)
Y	Rotation around Y-axis in degrees (WCS for G17)
Tool	Tracking of tool tip when swiveling (This field can be enabled/disabled in the swivel data record) • **Track** The position of the tool tip is maintained during swiveling. (The option 5-axis transformation is required and must be setup). • **Do not track** The tool tip position is not maintained during swiveling. (in order to avoid a collision with the workpiece it is recommended to retract the tool to a safe position before swivelling).

Notes

Programming example 2: CYCLE800 "Approach tool" in the programmGUIDE G code





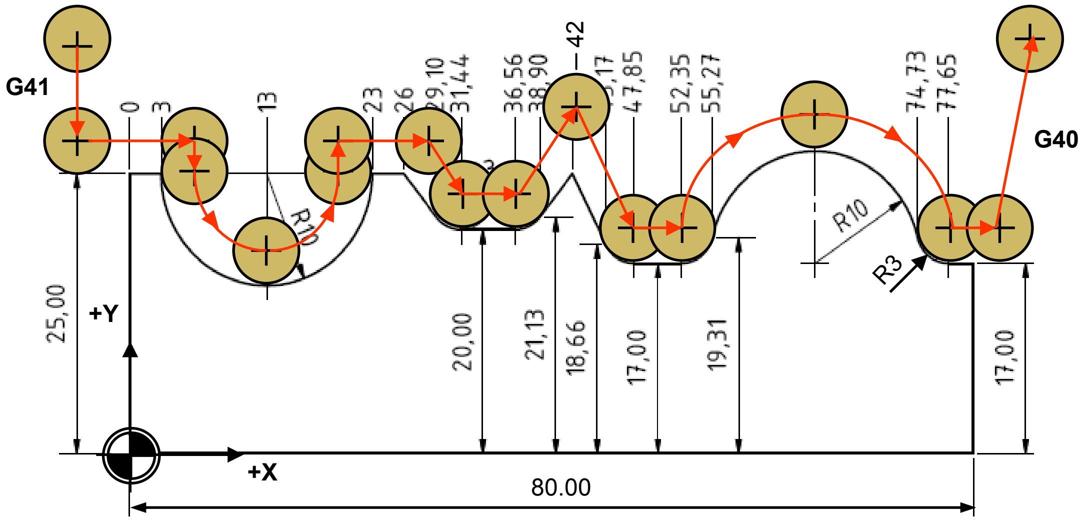
Work steps:

- *Free-form surface without tool approach with roughing allowance*
- *Free-form surface with tool approach to final finishing dimension*

Tools: Ball nose cutter D=5mm
Blank material: Aluminum AlSiMg 120 x 40 x 30

Notes

Programming tool path in swiveled plane (rotation about X axis 90 degree) tool setting angle parallel to surface normal (work plane G17).



Programming notes:

The programmed tool path is offset incrementally in X (R50). The program loop in the main-program after the M30 repeats the subprogram by the calculated number of repeats in the parameter „**P**“.

The subprogram in this example is located in the same directory as the main-program and is called up with the „**CALL**“ command.

The control offers programming of circular interpolation **G2** / **G3** with specification of circle center point **I, J, K** or circle center **CR=...** . Circular interpolation can be executed with change in tool orientation. The work plane remains hereby unchanged in the G17 plane.

The tool path must be programmed to the tool center point (TCP), in order to avoid a contour violation, due to tool orientation about the tool center point with CYCLE800 “Approach tool”. The tool path on the programmed contour is offset to the center with activation of tool radius compensation.
The tool is measured by the machine operator to the tool tip.

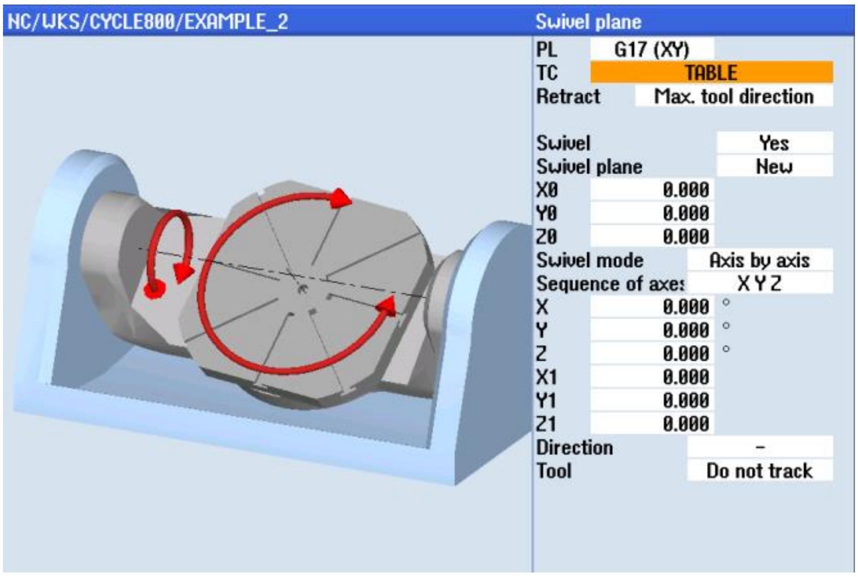
In case of a CAM generated point to point program of the tool path to the center line of the ball sphere center it is possible to offset the tool length in the program with the command TOFFL
(e.g. Ball diameter 5mm TOFFL=-2.5).

Notes

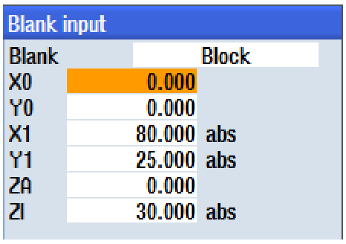
Program: EXAMPLE_2_CYCLE800.MPF

;**** 3+2 AXIS POSITIONING WITH CYCLE800 APPROACH TOOL ***

N100;*** **SWIVEL INTO BASE ORIENTATION (INITIAL SETTING)** ***
N100 CYCLE800(4,"TABLE",200000,57,0,0,0,0,0,0,0,0,0,1,,1)

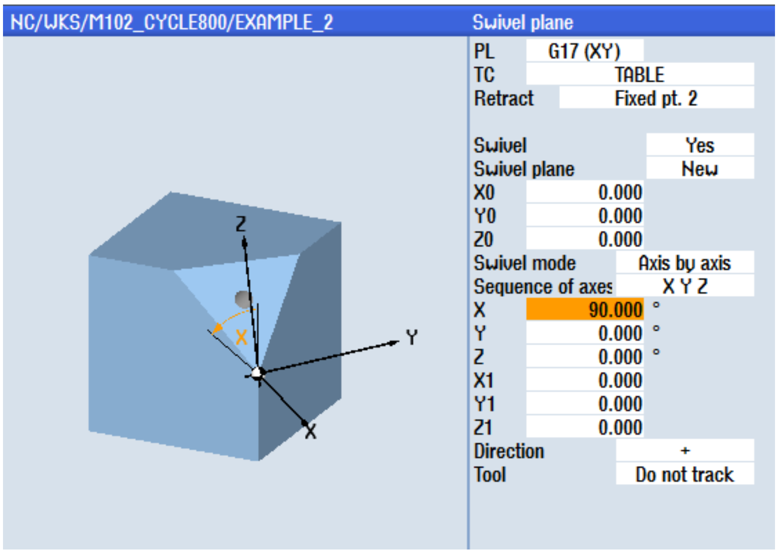


N102 G54 G17
;********* **BLANK INPUT (SIMULATION)** *********
N103 WORKPIECE(,,,"BOX",112,0,30,-80,0,0,100,80)

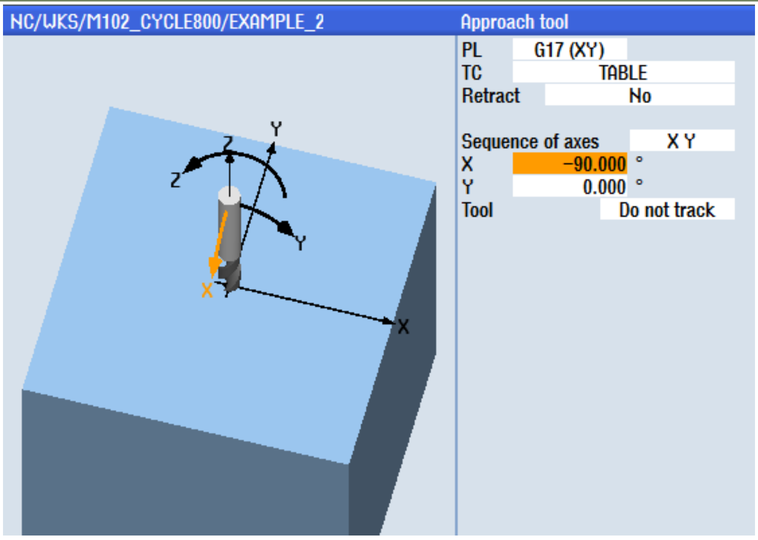


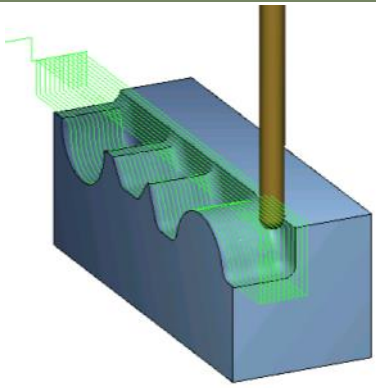
Notes

;****************** **SET SURFACE NORMAL** ******************
N114 CYCLE800(2,"TABLE",200000,57,0,0,0,90,0,0,0,0,0,1,,1)



;************* **SET TOOL ORIENTATION VERTICAL** **********
N115 CYCLE800(0,"TABLE",101,57,,,,-90,0,,,,,-1,100,1)
N121 REPEAT MARKER_1 MARKER_2 P=R53 ;Jump to the subprogram call after M30

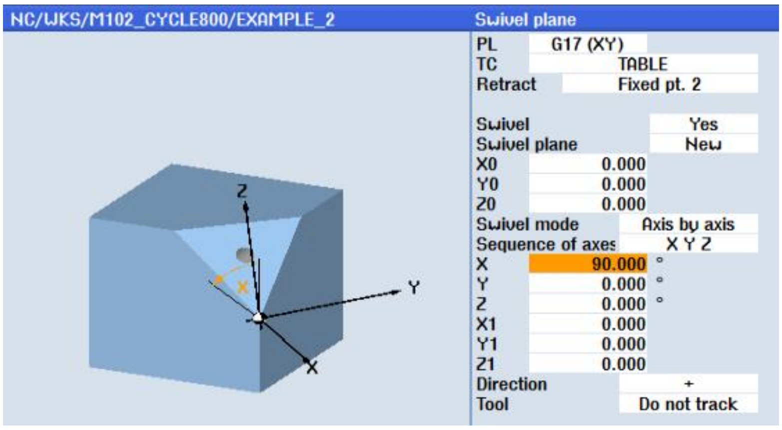




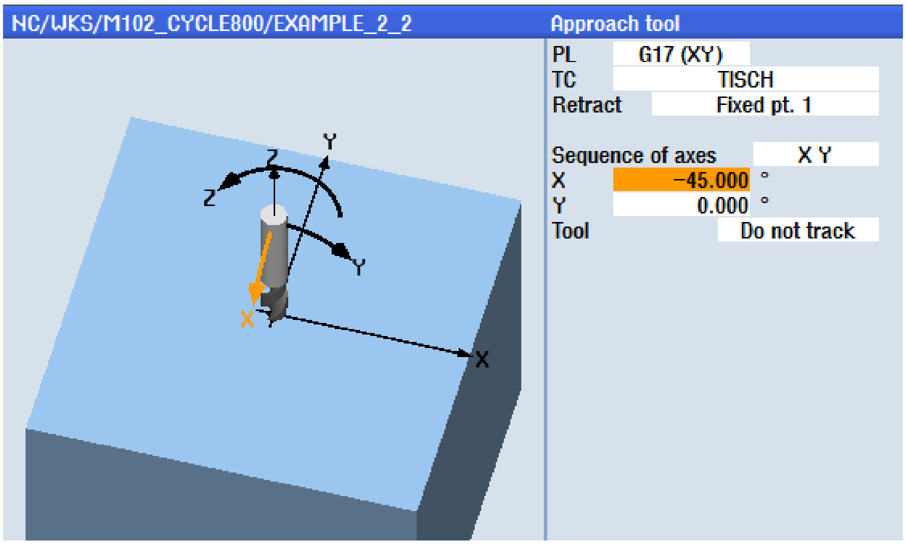
;* TECHNOLOGY 3+2 AXIS FINISHING WITH APPROACH TOOL*****
N123 ;**
N124 R50=0.5 ;INFEED IN X
N125 R51=100 ;WORKPIECE LENGTH
N126 R52=3 ;BALL RADIUS
N127 R53=R51/R50 ;CALCULATION OF THE REPEATS
N107 R54=1000; F APPROACH
N108 R55=3000; F XY
N109 R56=1000; F RETRACT
N128 ;***

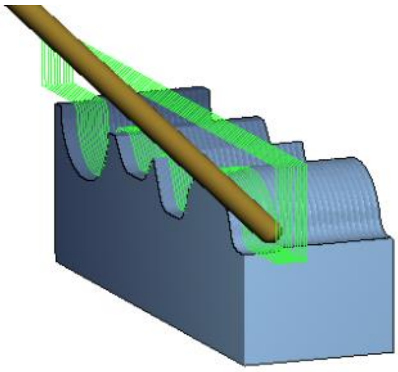
Notes

;*********************** **SET SURFACE NORMAL** ************************
N128 CYCLE800(1,"TABLE",200000,57,0,0,0,90,0,0,0,0,0,1,,1)



;********* **SET TOOL ORIENTATION 45 DEG** **********
N129 CYCLE800(0,"TABLE",101,57,,,,-45,0,,,,,-1,100,1)





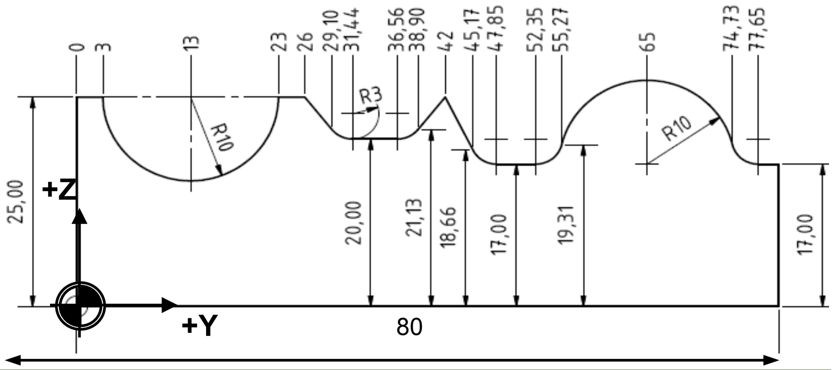
Notes

NC/WCS/DOCU_TOOL_APPROACH/EXAMPLE_1

```
N10 CYCLE800(2,"TABLE",200000,57,0,0,0,0,0,0,0,0,0,1,,1)
N20 CYCLE800(0,"0",200000,57,0,0,0,0,0,0,0,0,0,1,,1)
N30 G54 G17
N40 WORKPIECE(,"",,"BOX",112,0,30,-80,0,0,40,120)
N60 T="BALL_D6"
N70 M6
N80 S12000 M3 M8 F2000
N90 G0 X0 Y-10
N100 Z35
N110 TOFFL=-3 ;TOOL LENGTH TO THE BALL CENTER
N120 ;************************************************
N130 R50=1 ;INFEED IN X
N140 R51=40 ;TOOL LENGTH
N150 R52=3+0.2 ;BALL RADIUS + ALLOWANCE
N160 R53=R51/R50 ;CALCULATION OF THE REPEATS
N170 R54=1000
N180 R55=2500
N190 R56=3000
N200 ;*************************************************
N210 CYCLE832(0.02,_ROUGH,1)
N220 REPEAT MARKER_1 MARKER_2 P=R53
N230 ;************************************************
N240 R50=0.5 ;INFEED IN X
N250 R51=40 ;TOOL LENGTH
N260 R52=3 ;BALL RADIUS
N270 R53=R51/R50 ;CALCULATION OF THE REPEATS
N280 R54=1000
N290 R55=2500
N300 R56=3000
N310 ;*************************************************
N320 CYCLE832(0.005,_FINISH,1)
N330 ;****TOOL APPROACH*****
N340 CYCLE800(1,"TABLE",200000,57,0,0,0,0,0,0,0,0,0,1,,1)
N350 CYCLE800(0,"TABLE",101,54,,,,-45,0,,,,,-1,100,1)
N360 G0 X0 Y-10
N370 Z35 M8
N380 REPEAT MARKER_1 MARKER_2 P=R53
N390 CYCLE800(2,"TABLE",200000,57,0,0,0,0,0,0,0,0,0,1,,1)
N400 M30
N410 MARKER_1:
N420 CALL "PROFILE_1" ;#SM;*RO*
N430 MARKER_2:
```

Subprogram: PROFILE_1.SPF

```
N10 G1 Z=25+R52 F=R54
N20 G1 X=IC(R50)
N30 Y=3+R52 F=R55
N40 Z=25
N50 G3 Y=23-R52 Z25 J=AC(13) K=AC(25)
N60 G1 Z=25+R52
N70 G1 Y=26+R52
N80 G1 Y=31.44 Z=20+R52
N90 G1 Y=36.56
N100 G1 Y=42 Z=25+R52
N110 G1 Y=47.85 Z=17+R52
N120 G1 Y=52.35
N140 G2 Y=77.65 Z=17+R52 J=AC(65)
K=AC(17)
N150 G1 Y=125
N160 G0 Z=35+R52
N170 Y-10
N180 Z30 F=R56
N190 M17
```



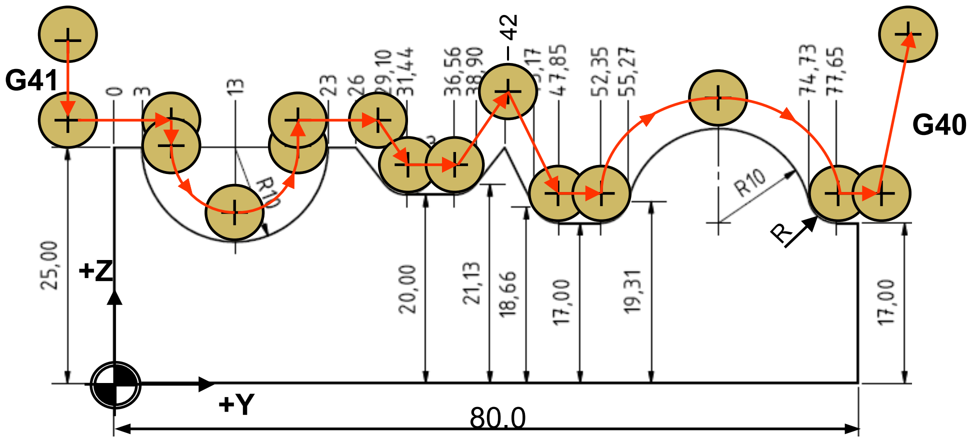
Notes

NC/WCS/DOCU_TOOL_APPROACH/EXAMPLE_2

```
N10 CYCLE800(2,"TABLE",200000,57,0,0,0,0,0,0,0,0,0,1,,1)
N20 CYCLE800()
N30 G54 G90 G17 G64
N40 WORKPIECE(,"",,"BOX",112,0,30,-80,0,0,40,120)
N50 T="BALL_D6"
N60 M6
N70 S12000 M3  F2000
N80 G0 X-5 Y-10
N90 GOTOF TEST
N100 Z35
;************************************************
N110 TOFFR=0.2 ;TOOL RADIUS OFFSET CORRECTION
N120 R50=1 ;STEP OVER PATH
N130 R51=45 ;WORKPIECE LENGTH
N140 R53=R51/R50 ;NUMBER OF REPEATS
N150 R54=1000 ;IN-FEED
N160 R55=2500 ;PLANE-FEED
N170 R56=3000 ;RETRACT-FEED
;*******************************************************
;************** SET SURFACE NORMAL ******************
N180 CYCLE800(0,"TABLE",220000,54,0,0,0,90,0,0,0,0,0,0,,1)
;*******************************************************
;******* SET TOOL ORIENTATION VERTICAL **************
N190 CYCLE800(0,"TABLE",101,54,,,,-90,0,,,,,-1,100,1)
N200 G0 Y-10 X-40
N210 X-35
N220 G1 Z-2 F=R54
N230 REPEAT MARKER_1 MARKER_2 P=R53
;************************************************
N240 TEST:
N250 TOFFR=0 ;TOOL RADIUS OFFSET CORRECTION
N260 R50=0.5 ;STEP OVER PATH
N270 R51=45 ;WORKPIECE LENGTH
N280 R53=R51/R50 ;NUMBER OF REPEATS
N290 R54=1000 ;IN-FEED
N300 R55=2500 ;PLANE-FEED
N310 R56=3000 ;RETRACT-FEED
;*******************************************************
;************** SET SURFACE NORMAL ******************
N340 TOFFL=-$P_TOOLR
N320 CYCLE800(1,"TABLE",220000,54,0,0,0,90,0,0,0,0,0,0,,1)
;*******************************************************
;********* SET TOOL ORIENTATION 45 DEG **************
N350 ATRANS X=-$P_TOOLR
N330 CYCLE800(0,"TABLE",101,54,,,,-45,0,,,,,-1,100,1)
N360 G0 Y-10 X-40
N370 X-35
N380 Z-2
N390 REPEAT MARKER_1 MARKER_2 P=R53
N400 CYCLE800(2,"TABLE",200000,57,0,0,0,0,0,0,0,0,0,1,,1)
N410 M30
N420 MARKER_1:
N430 CALL "PROFILE_2" ;#SM;*RO*
N440 MARKER_2:
```

Subprogram: PROFILE_2.SPF

```
N10 G1 Y-10 X-35 F=R56
N20 G1 Z=IC(R50) F=R54
N30 g1 g41 Y-10 X-25 F=R55
N40 Y3
N50 g3 Y23 X-25 i=ac(-25) j=ac(13)
N60 g1 Y26
N70 Y29.1 X-21.13
N80 g3 Y31.44 X-20 cr=3
N90 g1 Y36.56
N100 g3 Y38.9 X-21.13 cr=3
N110 g1 Y42 X-25
N120 g1 Y45.17 X-18.66
N130 g3 Y47.85 X-17 cr=3
N140 g1 Y52.35
N150 g3 Y55.27 X-19.31 cr=3
N160 g2 Y65 X-27 cr=10
N170 g2 Y74.73 X-19.31 cr=10
N180 g3 Y77.65 X-17 cr=3
N190 g1 Y120
N200 G0 g40 Y125 X-35
N210 Y-10
N220 m17
```



Notes

Set-up CYCLE800

Configuration of machine kinematics

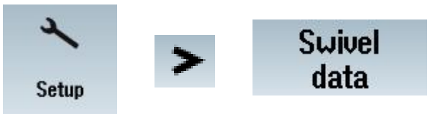
General explanation

The commissioning of the kinematic chain of the machine is a mandatory requirement for the correct swivelling functionality of CYCLE800. The kinematic chain is parameterized and stored in a swivel data record, that consists of 40 tool carrier system variables $TC_CARR1[n] to $TC_CARR40[n].
(or 65 tool carrier system variables $TC_CARR1[n] to $TC_CARR40[n] if the fine offset vectors are activated)

The input of the individual tool carrier variables are supported by a input mask in order to simplify the set-up process.

Set-up of a swivel data record

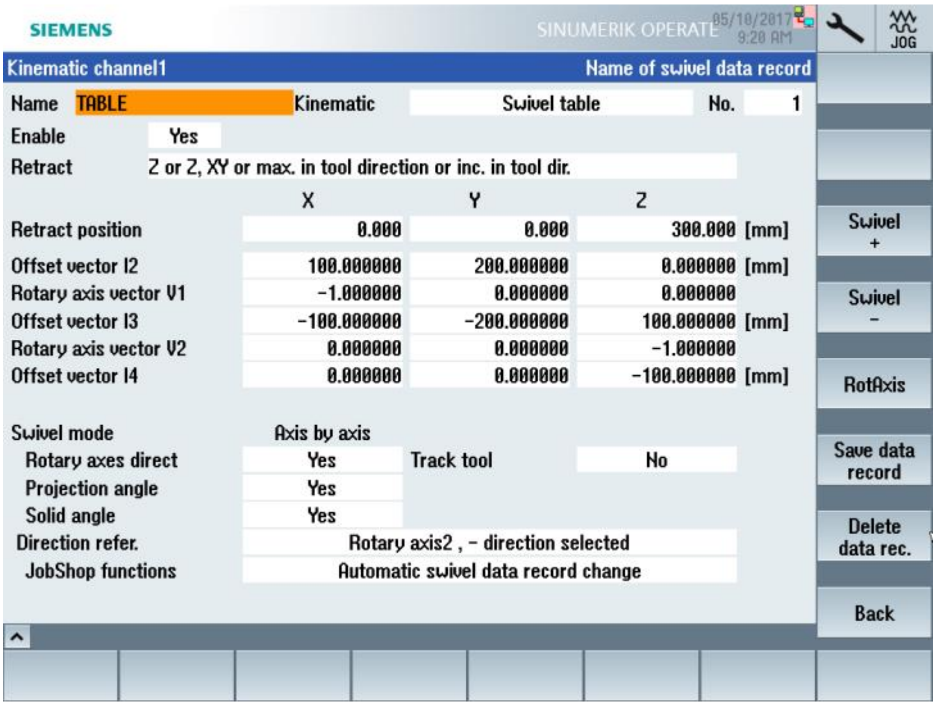
Press the following softkeys to excess the setup area of the swivel data record:



Note: Manufacturer Password must be set

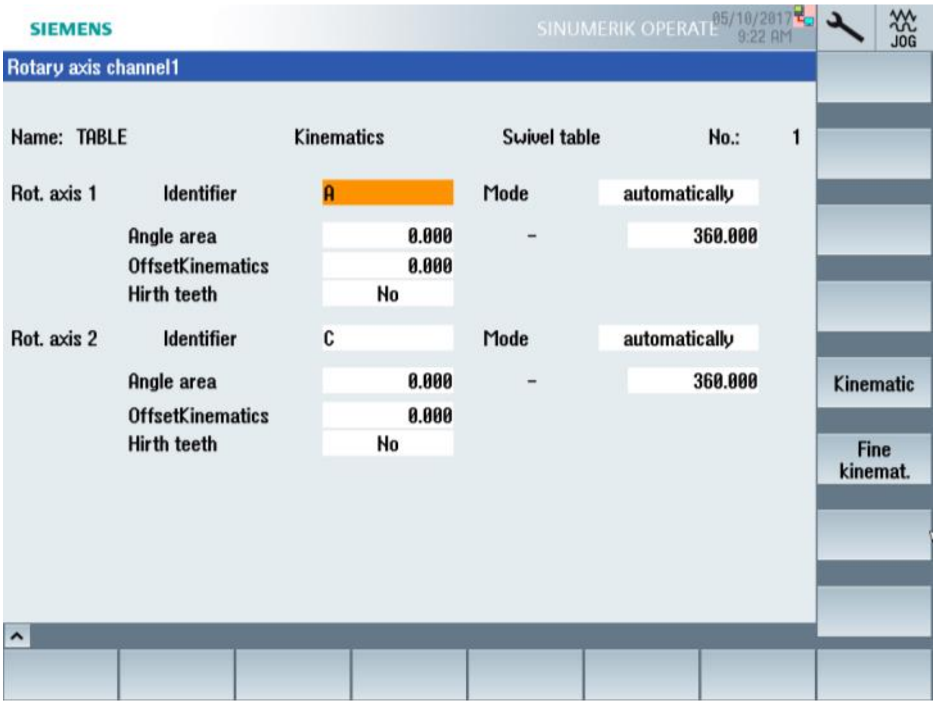
The swivel data record is hereby split into two sections:

- ***Basic Kinematic settings in channel [n]***

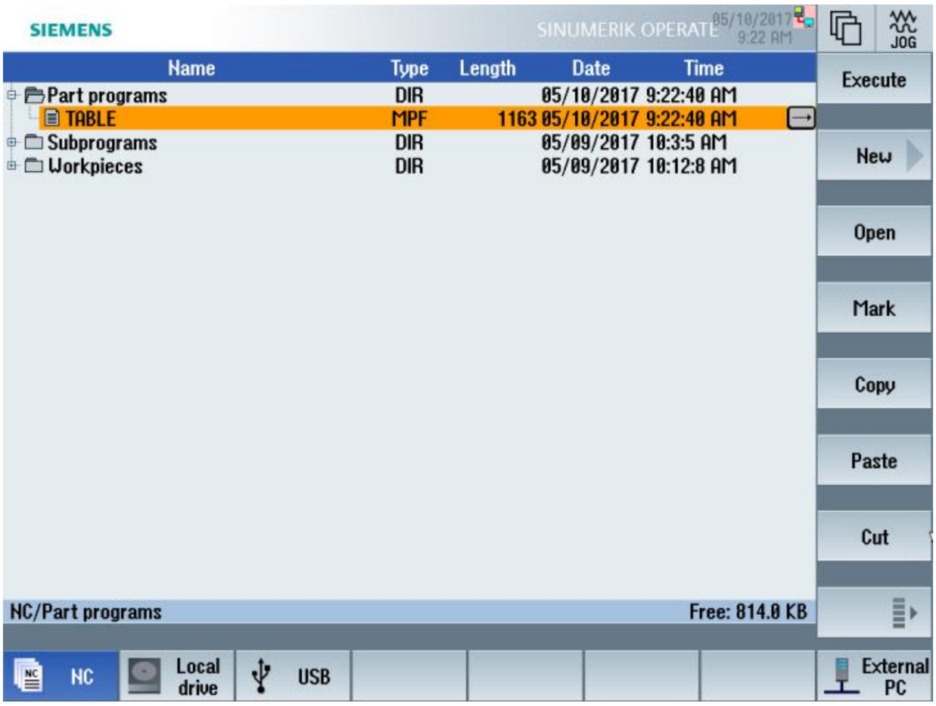


Notes

- ***Rotary axis settings in channel [n]***



Pressing the Softkey “**save data record**” creates a “**MPF**” file in the directory “**Part programs**” with all tool carrier system variables and their assigned values/names of the swivel data record.



Notes

Example:** Swivel data record for AC Swivel rotary table kinematic type **P

```
$TC_CARR1[1]=0              ;Offset vector I1 (X)
$TC_CARR2[1]=0              ;Offset vector I1 (Y)
$TC_CARR3[1]=0              ;Offset vector I1 (Z)
$TC_CARR4[1]=250            ;Offset vector I2 (X)
$TC_CARR5[1]=200            ;Offset vector I2 (Y)
$TC_CARR6[1]=150            ;Offset vector I2 (Z)
$TC_CARR7[1]=-1             ;Rotary axis vector V1 (X)
$TC_CARR8[1]=0              ;Rotary axis vector V1 (Y)
$TC_CARR9[1]=0              ;Rotary axis vector V1 (Z)
$TC_CARR10[1]=0             ;Rotary axis vector V2 (X)
$TC_CARR11[1]=0             ;Rotary axis vector V2 (Y)
$TC_CARR12[1]=-1            ;Rotary axis vector V2 (Z)
$TC_CARR13[1]=0
$TC_CARR14[1]=0
$TC_CARR15[1]=0             ;Offset vector  I3 (X)
$TC_CARR16[1]=0.02          ;Offset vector  I3 (Y)
$TC_CARR17[1]=-150.02       ;Offset vector  I3 (Z)
$TC_CARR18[1]=-250          ;Offset vector  I4
$TC_CARR19[1]=-199.99       ;Offset vector  I4
$TC_CARR20[1]=0.02          ;Offset vector  I4
TC_CARR21[1]=X
$TC_CARR22[1]=X
```

```
$TC_CARR23[1]="P"           ;Kinematic type
$TC_CARR24[1]=0
$TC_CARR25[1]=0
$TC_CARR26[1]=0
$TC_CARR27[1]=0
$TC_CARR28[1]=0
$TC_CARR29[1]=0
$TC_CARR30[1]=-100          ;1.Rotary axis min. range
$TC_CARR31[1]=0             ;2.Rotary axis min. range
$TC_CARR32[1]=100           ;1.Rotary axis max. range
$TC_CARR33[1]=360           ;2.Rotary axis max. range
$TC_CARR34[1]="TABLE"       ;Name of swivel data record
$TC_CARR35[1]="A"           ;Rotary axis 1 identifier
$TC_CARR36[1]="C"           ;Rotary axis 2 identifier
$TC_CARR37[1]=415018003     ;Display variants swivel cycle
$TC_CARR38[1]=200           ;Retract position X
$TC_CARR39[1]=200           ;Retract position Y
$TC_CARR40[1]=300           ;Retract position Z
M30
```

Notes

Explanation of softkeys

Swivel +/-:

These softkeys allow to page through all existing swivel data records that are Set-up on the machine, hence various machine kinematics (e.g. TABLE, HEAD, HEAD_TABLE, HEAD_TABLE_45, TABLE_45, HEAD_HEAD_45, TURNING, ….).

If you wish to create a new swivel data record, continue to press this softkey until you reach to a tool carrier number that is not yet configured.
The maximum number of available tool carriers is set in **MD18088**

(**Attention !** This machine data reorganizes the buffered memory SRAM)

RotAxis:

This softkey opens the next section for parameterization of the rotary axis settings in channel

Kinematic:

This softkey appears in section “RotAxis”, and returns when pressing to the menu of the main kinematic settings of the channel.

Save data record:

Attention ! This softkey does not store any changes made in the swivel data input mask, but only creates a MPF file in the “Part program” directory with the name assigned in the swivel data record. This file contains all TCARR system variables which define a orientable tool carrier (see example of swivel data record in section 4.2).

Delete data rec.:

By pressing this softkey you delete the current data record from the NC.

Back:

This softkey closes the swivel data record input mask and returns to the previous menu screen.

Note:

Any changes made in the swivel data input mask are immediately active and stored to the NC.

Notes

Description of the TCARR parameters

Name:
$TC_CARR34[n]
n = Swivel data record number

Shows the names of all existing swivel data records that are Set-up on the machine, hence various machine kinematics.

If several swivel data records are declared in each NC channel, then a name is assigned to each swivel data record. No name needs to be specified if the swivel-mounted tool holder is not exchangeable (i.e. one swivel data record per channel).

Switching to the next swivel data record is done with the **"Swivel +"** or **"Swivel -"** softkey.

Note:

Swivel data record names may only contain permitted NC programming characters (A...Z, 0..9 and _)!

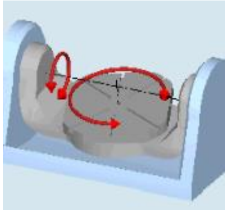
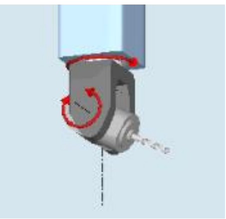
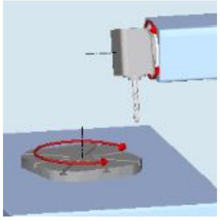
The names can be freely chosen by the machine tool manufacturer to suit the particular kinematic design of his machine.

Kinematics:
$TC_CARR23[n]
n = Swivel data record number

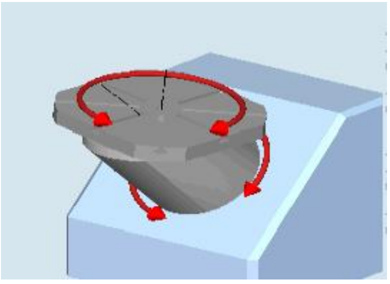
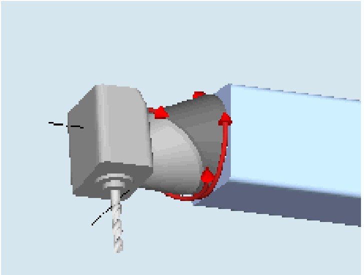
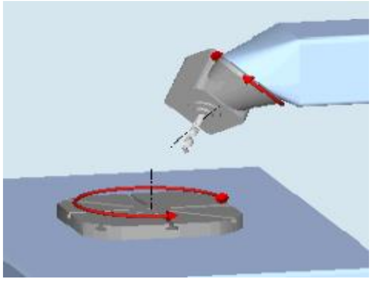
The basic machine kinematics of the machine tool are defined in this field.
The choice of selection refers to the machine types T, P, M.

Selection is made with the "**SELECT**" key (toggle).

Following selection can be made in this field:
- Swivel table (type P)
- Inclined head (type T)
- Inclined head + swivel table (type M)

TABLE (Swivel table)	HEAD (Inclined head)	HEAD + TABLE (Swivel head + Swivel table)
Orientable tool carrier (type P)	Orientable tool carrier (type T)	Mixed orientable tool carrier (type M)
		

Notes

TABLE (Swivel table)	HEAD (Swivel Head)	HEAD + TABLE (Swivel head + Swivel table)
Nutated orientable tool carrier (type P)	Nutated orientable tool carrier (type T)	Nutated orientable tool carrier (type M)
		

Note:

The distinction between a nutated rotary axis and a parallel swivel rotary axis is made by the definition of the components of the rotary axes vectors V1 and V2 in the input mask of the swivel data record.

No.:
TCARR=...

Displays the number of the swivel data record (orientable tool carrier).

The maximum number of definable tool carriers can be set in
MD18088 $MN_MM_NUM_TOOL_CARRIER.
*(**Attention !** This machine data reorganizes the buffered memory SRAM)*

The number in this field corresponds with the index [n] in the system variables for the tool carriers.

In general there is usually only one swivel data record defined that describes the particular machine tool kinematics, except machines with exchangeable tool carrier or mill/turn machines that have to kinematics, one for turning mode and the second for milling with B-axis.

Notes

Retraction mode:

$TC_CARR37[n].

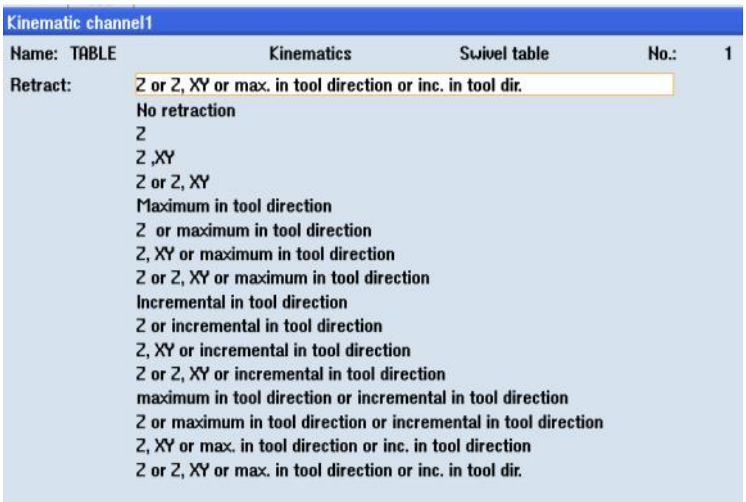
n = Swivel data record number

In this field you can choose between various retraction modes before swivelling, which are displayed in the input mask of the swivel cycle.

Selection is made with the “**SELECT**” key (toggle).

This input field corresponds with the **ONE MILLION** and **TEN MILLION** positions (Bit 6 and 7) of the system variable $TC_CARR37[n].

A selection of the following retraction modes can be made:



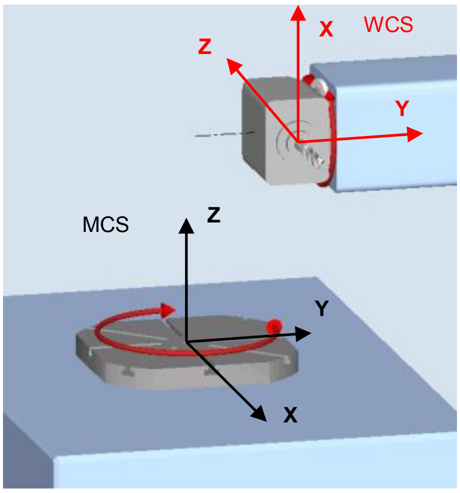
Note:

- Maximum and incremental retraction in tool direction always occurs in the positive tool direction (away from workpiece).
- Retraction in tool direction is always executed prior to swivelling the rotary axes with the current tool orientation. If the swivel plane is undefined at the start of a program, the axis should first be traversed to a safe position or be safely prepositioned using retraction Z.
- Maximum and incremental retraction in tool direction is particularly suitable for machines with swivel heads. In the case of a machine with a swivel head and an active swivel frame.
- In case of a machine with a swivel head (type T) or mixed kinematics (type M) and active swivel frame (rotation in workpiece reference not equal to 0), traversing of multiple machine axis can occur, when traversing the tool axis (retract or approach).

Notes

Example:

In the case of a machine with a swivel head, the head and the WCS in the G17 plane is swivelled -90 degrees around the X -axis. Selection of Retraction mode “**max. tool direction**” for the Z-axis in the swivel cycle, results in traversing the machine axis **X** to the positive software end position.



Retract position:
$TC_CARR38[n] .. to $TC_CARR40[n]..
n = Swivel data record number

In this field you can specify a absolute retract position in X Y Z (Fixed point in the MCS). The position input in these fields for X, Y, and Z refer only to the retraction modes “Z” and “Z, XY” (or Fixed point 1 and 2).

$TC_CARR38[n]	Retract position for X-axis
$TC_CARR39[n]	Retract position for Y-axis
$TC_CARR40[n]	Retract position for Z-axis

This input field corresponds with the following TCARR system variables:

Modification of the retraction type is done in the user CUST_800 cycle. If no changes are made in the CUST_800.SPF, then retraction takes place either along the Z axis or retracting the axes Z, XY to a absolute machine position defined in the pa-

Note:

Alternatively the display text for retract variants “Z” and “Z,XY” can be changed to „fixed point 1“ and „fixed point 2“ in channel setting data ***SD55221$SCS_FUNKTION_MASK_SWIVEL_SET***.

Notes

<u>Offset vectors I1 - I4 and rotary axes vectors V1 and V2;</u>
$TC_CARR1[n] …. $TC_CARR20[n]
n = Swivel data record number

The Vectors always contain three components, which represent the reference to the X, Y, Z machine axes. Depending on the kinematic type, vector chains always have to be closed with a 3rd Vector.

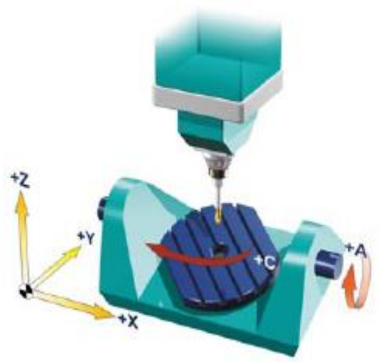
Offset vectors I1 to I4 refer to the non-swivelled state of the rotary axes (initial setting of machine kinematics).

The offset vectors do not have to point to the pivot point of the rotary axes. The important thing is that they point to a point on the direction of rotation (intersection point of rotary axes).

The sign of the offset vectors (I1 to I4) and the rotary axis vectors (V1, V2) result from the specifications of the axis directions according to ISO or DIN (right-hand rule). In the case of kinematics that move the workpiece (rotary table), the motion of the workpiece relative to the tool must be taken into account.

Kinematic types and their offset-/rotary axes vectors

Swivel table (kinematic type P)	
Description	**Tool carrier system variable**
Kinematic type	$TC_CARR23[1]="**P**"
Offset vector I2	$TC_CARR4...6[n] (x, y, z)
Offset vector I3	$TC_CARR15...17[n] (x, y, z)
Offset vector I4	$TC_CARR18...20[n] (x, y, z)
Rotary axis vector V1	$TC_CARR7...9[n] (x, y, z)
Rotary axis vector V2	$TC_CARR10...12[n] (x, y, z)



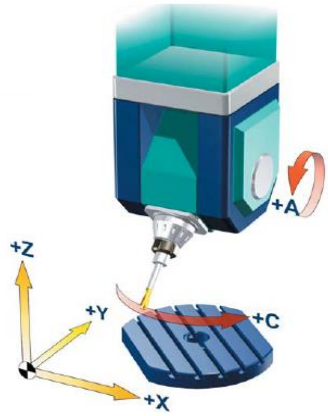
- I2 distance from the machine reference point to the pivot point/intersection of the 1st rotary axis
- I3 distance from the pivot point/intersection of the 1st rotary axis to the pivot point/intersection of the 2nd rotary axis (or to the reference point of the tool adapter)
- I4 closure of the I4=-(I2+I3) vector chain, if the swivel table cannot be changed

<u>Note:</u>

With a swivel table kinematics, the 1st rotary axis is always based on the 2nd rotary axis. This results in the following rotary axes vectors:
V1=-1 (1st Rotary axis A rotates about X)
V2=-1 (2nd Rotary axis C rotates about Z)
The rotary axes vectors must be inverted (value –1) since both rotary axes move the workpiece .

Notes

Inclined head + rotary table (kinematic type M)	
Description	**Tool carrier system variable**
Kinematic type	$TC_CARR23[1]="**M**"
Offset vector I1	$TC_CARR1...3[n] (x, y, z)
Offset vector I2	$TC_CARR4...6[n] (x, y, z)
Offset vector I3	$TC_CARR15...17[n] (x, y, z)
Offset vector I4	$TC_CARR18...20[n] (x, y, z)
Rotary axis vector V1	$TC_CARR7...9[n] (x, y, z)
Rotary axis vector V2	$TC_CARR10...12[n] (x, y, z)



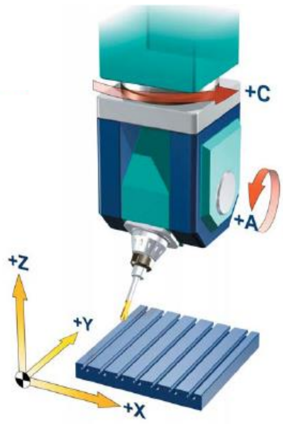
- I2 distance from the tool adapter to the pivot point/intersection of the 1st rotary axis
- I1 closure of the I1=-I2 vector chain, if the swivel head cannot be changed
- I3 distance from the machine reference point to the pivot point/intersection of the 2nd rotary axis (or to the reference point of the tool adapter)

Note:

With mixed kinematics, the 1st rotary axis is always the axis that changes the tool orientation. This results in the following rotary axes vectors:
V1=1 (1st Rotary axis A rotates about X)
V2=-1 (2nd Rotary axis C rotates about Z)
Since the 2nd rotary axis moves the workpiece the vector must be inverted.

- I4 closure of the I4=-I3 vector chain, if the swivel table cannot be changed

Inclined head (kinematic type T)	
Description	**Tool carrier system variable**
Kinematic type	$TC_CARR23[1]="**T**"
Offset vector I1	$TC_CARR1...3[n] (x, y, z)
Offset vector I2	$TC_CARR4...6[n], (x, y, z)
Offset vector I3	$TC_CARR15...17[n] (x, y, z)
Rotary axis vector V1	$TC_CARR7...9[n] (x, y, z)
Rotary axis vector V2	$TC_CARR10...12[n] (x, y, z)



- I3 distance from the tool adapter to the pivot point/intersection of the 2nd rotary axis.
- I2 distance from the pivot point/intersection of the 2nd rotary axis to the pivot point/intersection of the 1st rotary axis.
- I1 closure of the I1=-(I2+I3) vector chain, if the swivel head cannot be changed.

Note:

With a swivel head kinematics, the 2nd rotary axis is always based on the 1st rotary axis. This results in the following rotary axes vectors:
V1=1 (1st Rotary axis C rotates about Z)
V2=1 (2nd Rotary axis A rotates about X)

Notes

Swivel mode:
$TC_CARR37[n]
n = Swivel data record number

In this section it is possible to activate various methods of swivelling, that are being displayed in the cycle input mask of the swivel cycle. Whereby swivel mode axis by axis is set active as preferred default and can not be deselected.

This input field corresponds with the **ONES** position of the system variable $TC_CARR37[n].

Selection is made with the "**SELECT**" key (toggle between "yes" and "no" to activate or deactivate, exception swivel mode axis by axis).

Following swivel modes are available and can be selected or deselected:

- Axis by axis (standard setting can not be deselected)
- Rotary axis direct
- Projection angle
- Solid angle

Note:

Selection **"directly",** allows rotary axis positions to be entered directly. After rotary axes have been positioned (or manual rotary axes have been rotated), a swivel frame is calculated for these positions. In swivel mode **"directly"**, input fields with the names of the rotary axes of the active swivel data record are displayed instead of "Rotation about" input fields.

In JOG mode the vertical softkey "**Teach-in**" is provided take over the actual rotary axes positions into the input mask of the swivel function.

Track tool:
$TC_CARR37[n].
n = Swivel data record number

In this field the tool tracking function with the 5-axis transformation function **TRAORI** can be enabled in the cycle input mask of the swivel cycle.
If **"yes"** is selected there is a dynamic tracking of the tool tip active, with the movement of the rotary axes the current position is maintained by the tip of the tool relative to the workpiece.

This input field corresponds with the **TEN THOUSAND** position of the system variable
$TC_CARR37[n]

Selection is made with the "**SELECT**" key (toggle).

A selection of the following choice can be made:

- No (Display field for tool tracking is not displayed).
- Yes (Display field for tool tracking is displayed).

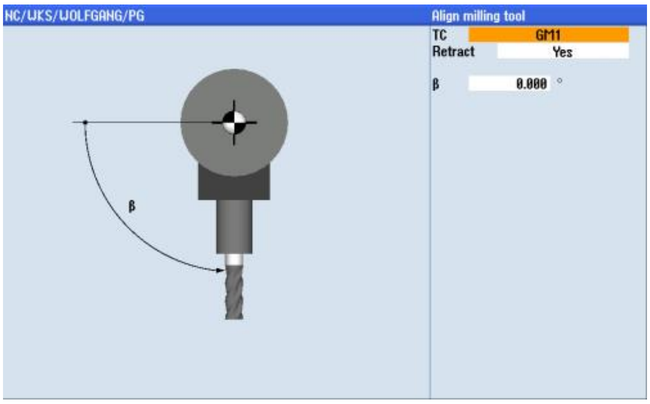
Note:

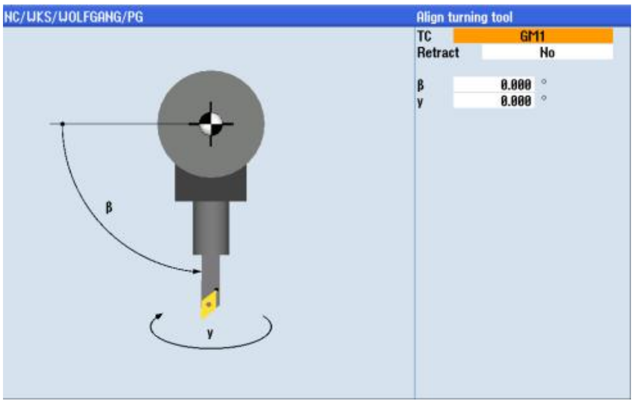
TRAORI requires the option 5-axis transformation.

Notes

B-axis kinematics:
$TC_CARR37[n].
n = Swivel data record number

The selection in this field activates for mill turn or turn mill machines the "**Align tool**" function for milling and turning tools (see picture below).





In the case of "**Align tool**" in the turning area, activation of NC function CUTMOD ensures, that the correct tools for tool point direction or tool clearance angle of a turning tool are made available.

This input field corresponds with the **TEN THOUSAND** position of the system variable $TC_CARR37[n].

Explanation of function:

Up to software version V2.6 this functionality was designed for use with a specific configuration of turn/mill or mill/turn machines on which the tool orientation is implemented by a swivel axis B (around Y) with associated milling spindle (SP2). Milling as well as turning tools can be used.

With software version V2.7 Machine kinematics for mill/turn with Swivel table and nutated Head kinematics are also supported.

Notes

Direction reference:
$TC_CARR37[n].
n = Swivel data record number

In this field you can specify, the rotary axis to be affected by the direction reference. Where by a default setting for the direction of the initial setting can be made.

This input field corresponds with the **THOUSAND** position of the system variable $TC_CARR37[n].

Selection is made with the "**SELECT**" key (toggle).

One of the following direction references can be selected:

- No: no display, - direction (1)
- Rotary axis 1, - direction selected (2)
- Rotary axis 2, - direction selected (3)
- No: no display, + direction (4)
- Rotary axis 1, + direction selected (5)
- Rotary axis 2, + direction selected (6)

What is the 1st or 2nd rotary axis of a swivel data set ?

Rule for Typ P: Rotary axis 2 (C) is based on rotary axis 1 (A).
Rule for Typ T: Rotary axis 2 (A) is based on rotary axis 1 (C).
Rule for Typ M: With mixed kinematics, rotary axis 1 is always the axis for the tool orientation.

Explanation of the function:

The NCU calculates two possible solutions using the angle range of the rotary axes of the machine kinematics (set in $TC_CARR30[n] .. $TC_CARR33[n]). Usually, only one of these solutions is technologically suitable.

The rotary axis to which the two solutions are to refer is selected in this field. The solution to be applied is selected in the input screen form of the swivel cycle.

Note:

Selection **1** or **4** are made for machine kinematics that only have one solution to reach the programmed plane. The direction field in the cycle input mask of the swivel cycle is not displayed.

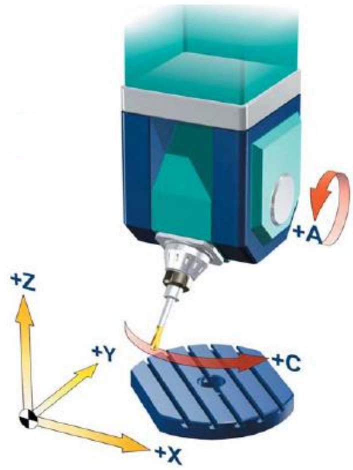
Selection **2** or **3** are made to control the preferred direction reference for rotary axis 1 or 2 in „**Minus**" direction as basic setting of the machine kinematics (initial setting).

Selection **5** or **6** are made to control the preferred direction reference for rotary axis 1 or 2 in „**Plus**" direction as basic setting of the machine kinematics (initial setting).

Notes

Example:

Effect of parameter "**Direction reference**" with mixed machine kinematics “**type M**”.



1st rotary axis
Swivel head **A** rotates around machine axis **X**
Angle range of 1st rotary axis –30° to +90°

2nd rotary axis
Rotary axis **C** rotates around machine axis **Z**
Angle range of 2nd rotary axis 0° to +360° (modulo axis)

Program 1:
Swivel CYCLE800 (... Rotation about X = 20°, ... minus direction)
A-axis rotates to - 20° and C-axis to 180°.

Program 2:
Swivel CYCLE800 (... Rotation about X = 20° degrees, ... plus direction)
A-axis rotates to 20° and C-axis to 0°.

Program 3:
Swivel CYCLE800 (... Rotation about X = 45° degrees, ... minus direction)
A-axis rotates to 45° and C-axis to 0°.

Program 4:
Swivel CYCLE800 (... Rotation about X = 45°, ... plus direction)
A-axis rotates to 45° and C-axis to 0°

Note:

The “**direction reference**” parameter has no effect in programs 3 and 4 because two solutions are not produced for the kinematics starting at 30° around X or Y due to the limited angle range of the A-axis (-30° to +90°) and therefore the first solution is always applied.

Notes

Enable:
$TC_CARR37[n]
n = Swivel data record number

This field enables or disables the selection of the swivel data record (orientable tool carrier TCARR) in the swivel cycle.

This input field corresponds with the **HUNDRED MILLION** position of the system variable $TC_CARR37[n].

Activation is made with the blue "**SELECT**" key (Toggle).

The following selection can be made:

- No: Swivel data record not released
- Yes: Swivel data record released

JobShop functions:
$TC_CARR37[n].
n = Swivel data record number

This field handles the Swivel data record change and tool change for **ShopMill** or **ShopTurn**.

This input field corresponds with the **HUNDRED MILLION** position of the system variable $TC_CARR37[n].

Activation is made with the blue "**SELECT**" key (Toggle).

The following selection can be made:

- Automatic swivel data record change
- Manual swivel data record change

For kinematic type T and M the following additional selection can be made:

- Automatic tool change
- Manual tool change

Explanation of the function:

The JobShop functions are used for changeable tool carriers such as angle head attachments for boring machines, which are changed into the spindle adapter either by hand or automatically.

ShopMill or Shop turn supports applications with angle head attachments and offers the functionality to activate the swivel data record automatically upon tool change.

Attention ! Adoptions must be made at the appropriate markers in the manufacturer cycle CUST_800 by the machine tool manufacturer.

Notes

Rot. axis 1 / Rot. axis 2 Identifier:
$TC_CARR35[n]; $TC_CARR36[n]
n = Swivel data record number

The following identifiers should be chosen preferable, if possible*:*

- A: Axis rotates around machine axis MX1
- B: Axis rotates around machine axis MY1
- C: Axis rotates around machine axis MZ1

For automatic rotary axes, the channel names of the corresponding NC rotary axes must be entered.

For manual (manually adjustable) and semi-automatic rotary axes, you can use any axis identifier (up to 6 letters or digits).

Mode:
$TC_CARR37[n]
n = Swivel data record number

In this filed you select the type of rotary axis on your machine tool for rotary axis 1 and rotary axis 2 in our example A and C.

This input field corresponds with the **TEN** position of the system variable $TC_CARR37[n].

Selection is made with the blue "**SELECT**" key (Toggle).

The following selection can be made:

- Automatically (NC rotary axes are traversed automatically)
- Manual (Rotary axes are manually positioned by the operator e.g. manual adjustable rotary axes)
- Semi-auto (Not a NC axis but indexing axis only)

Angle range:
$TC_CARR30[n] .. $TC_CARR33[n]
n = Swivel data record number

This field you can set the maximum and minimum traversing range of the available rotary axes of the machine

A valid angle range (e.g., -90 to +90 degrees) must be assigned to each rotary axis. This does not have to be the software end position range of the rotary axis in question. With modulo axes, you must enter a traversing range of between 0 and 360 degrees.

$TC_CARR30[n]	Angle range of rotary axis 1 (minimum value).
$TC_CARR31[n]	Angle range of rotary axis 2 (minimum value).
$TC_CARR32[n]	Angle range of rotary axis 1 (maximum value).
$TC_CARR33[n]	Angle range of rotary axis 2 (maximum value).

This input field corresponds to the following system variables:

Notes

__Offset kinematics:__
$TC_CARR24[n]; $TC_CARR25[n]
n = Swivel data record number

In this field an offset value is entered for rotary axis 1 or 2 when the position of the rotary axes is not equal to 0 in the initial setting of the kinematics. In the initial setting of the kinematics, the tool orientation to a geometry axis (X, Y, Z) must be parallel.

This input field corresponds to the following system variables:
__Hirth tooth system:__

$TC_CARR24[n]	Kinematics offset of rotary axis 1.
$TC_CARR25[n]	Kinematics offset of rotary axis 2.

$TC_CARR26[n]... $TC_CARR29[n]
n = Swivel data record number

In diesem Feld wird ausgewählt ob die Maschinenkinematik über eine Hirthverzahnung verfügt.

The following selection can be made with the blue “**SELECT**” key (Toggle):

- Yes: The fields “Angle offset” and “Angle grid” are displayed
- No: Subsequent fields are concealed

This input field corresponds to the following system variables:

$TC_CARR26[n]	Angular offset of Hirth gearing at the start of gearing of rotary axis 1.
$TC_CARR27[n]	Angular offset of Hirth gearing at the start of gearing of rotary axis 2.
$TC_CARR28[n]	Angle grid of the Hirth gearing for rotary axis 1.
$TC_CARR29[n]	Angle grid of the Hirth gearing for rotary axis 2.

__Kinematic fine offset:__
$TC_CARR41..65[n]
n = Swivel data record number

The fine offsets can be applied for temperature compensation of machine kinematics. For this purpose, the fine offsets can be written, activated or deactivated in a corresponding manufacturer cycle.

In TC_CARR41...65 **"Kinematics fine offset"** the vectors of machine kinematics I1 to I4 or the offset values of the rotary axes can be entered .

The fine offsets are activated by the following setting data:
SD 42974: $SC_TOCARR_FINE_CORRECTION = 1.

The fine offsets act in addition to the corresponding base vectors when the function "swivelling"

__Note:__

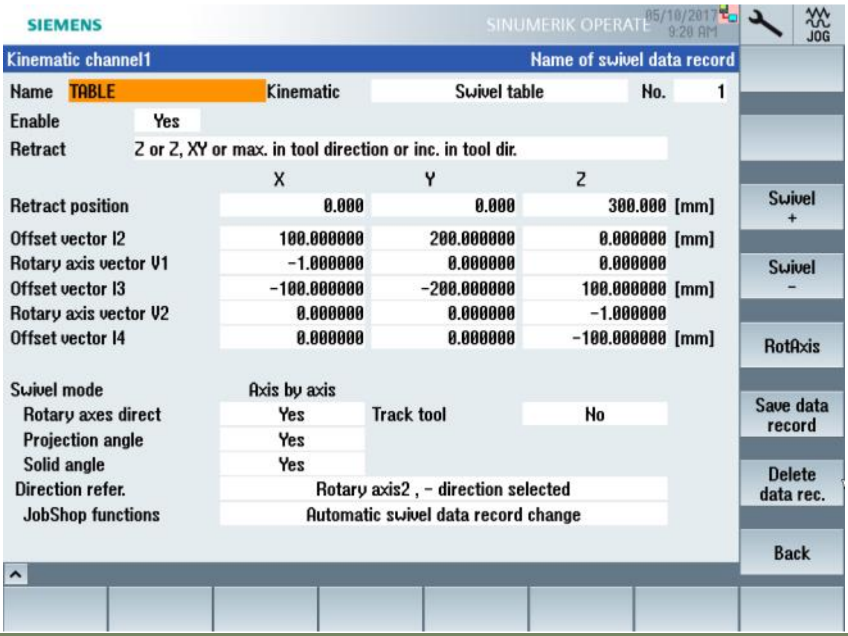
The tool carrier system variables for the kinematic fine offset are **not** part of the swivel data mask.

Notes

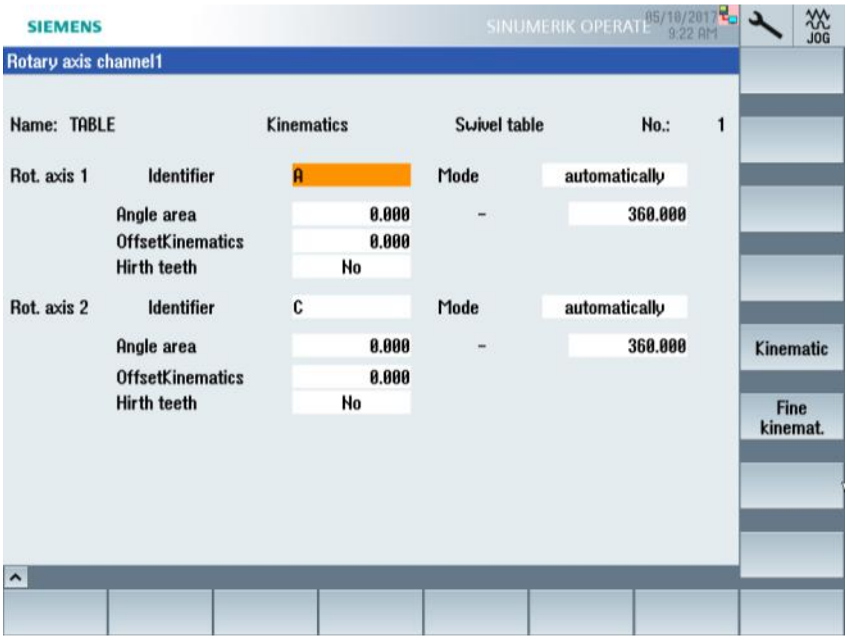
Set-up example of a swivel table kinematik type P

Swivel data record

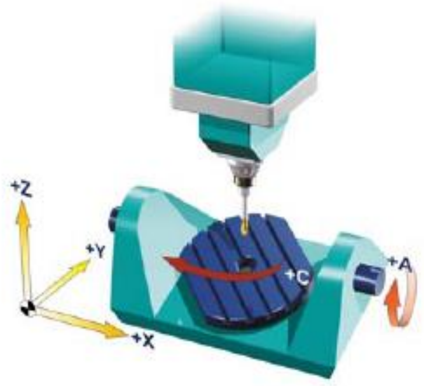
- Parameterization of offset-/rotary axes vectors and swivel functions



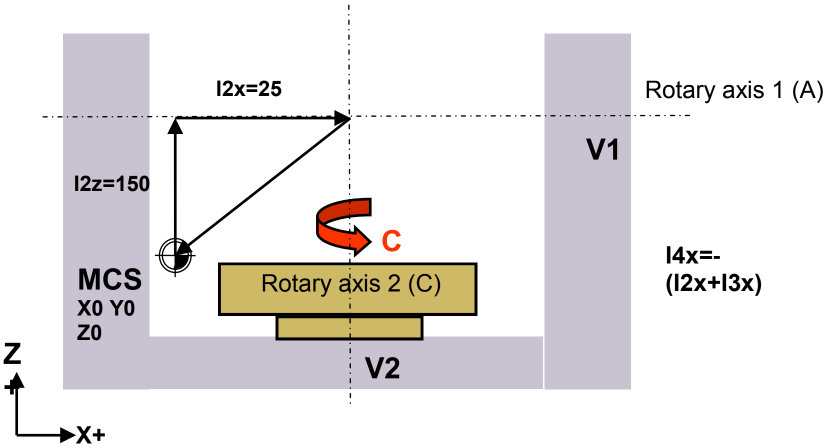
- Parameterization of rotary axes types



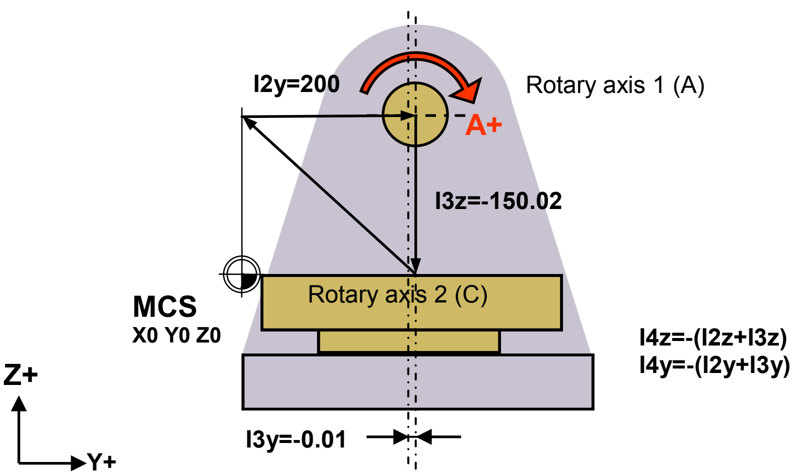
Notes



Front view in +Y direction of the machine



Front view in +X direction of the machine



Notes

Configuration of machine data

Activation of swivel cycle:

To activate the swivel function, there must be at least one orientable tool carrier (swivel data record) in the NCK and the workpiece, tool and rotary table reference system frames must be activated:

- Activate number of tool carrier via **MD18088**

MD18088 $MN_MM_NUM_TOOL_CARRIER	Maximum number of definable orientable tool carriers (number of swivel data records) (SRAM).
= 0	Do not declare any swivel data records
> 0	Number of definable swivel data records

MD28085 $MC_MM_LINK_TOA_UNIT	Assignment of orientable tool carrier (TCARR) to a channel (SRAM).

If several channels are declared in the NCU, the number of swivel data records can be split under consideration of this machine data as follows:

- The number of orientable tool carriers in MD18088 split by the link unit in MD28085 per channel.
- Declare all defined orientable tool carriers to one channel only.

Example: MD18088 $MN_MM_NUM_TOOL_CARRIER = 4
Number of channels = 2.
Assignment of the TO areas: ***MD28085 $MN_MM_LINK_TOA_UNIT= 2***, this results in two swivel data sets per channel.

MD28082 $MC_MM_SYSTEM_FRAME_MASK	System frames (SRAM)
= 3DH	
Bit 2 = 1	TCARR and PAROT (Rotary table reference)
Bit 3 = 1	TOROT and TOFRAME (Tool reference)
Bit 4 = 1	Workpiece reference point (Workpiece reference)

- Set system frame mask via **MD28082** accordingly

- Activate and enable CYCLE800 on the user interface via the channel-specific machine data

MD52212 $MCS_FUNCTION_MASK_TECH	Cross-technology function mask
Bit 0 = 1	Enable swivel

Note:

Changing machine data **MD18088**, **MD28082, MD28085** cause the buffered memory (SRAM) to be reorganized. After changing these MD's it is recommended to create a NC data backup, otherwise NC data will be lost.

Notes

Setting data for configuration of cycle input mask

SD55221 $SCS_FUNCTION_MASK_SWIVEL_SET	Function mask CYCLE800
Bit 0	Input field "No swivelling"
= 0	Hide
= 1	Display
Bit 1	Text displayed for retract the tool axis.
= 0	Display text Z = "Z", display text Z, XY = "Z,XY"
= 1	Display text Z = "Fixed point 1", Display text Z, XY = "Fixed point 2". If you wish to modify the retraction version "Z" or "Z, XY" via the manufacturer cycle CUST_800.SPF, the neutral text "Fixed point 1" and "Fixed point 2" can be displayed.
Bit 2	Deselecting the active swivel set.
= 0	If deselection is not permitted, the "Swivel data set" (TC) selection field is not displayed in the "Swivel" input screen.
= 1	Deselection permitted see also swivel data set parameter $TC_CARR37 HUNDRED MILLIONS position.
Bit 3	Displays the active swivel plane under swivel in JOG. The setting in the swivel function screen acts on all swivel data sets.
Bit 4	Optimized swivelling in basic position (pole position) of the kinematics.

Setting the behavior of settable frame with orientable tool carrier

In order to use the swivel cycle, set the following machine data, different to the default value, at least as follows:

MD10602 $MN_FRAME_GEOAX_CHANGE_MODE	
=1	The actual total frame (work offsets) is recalculated when switching over geometry axes (selecting/deselecting TRAORI).

MD21186 $MC_TOCARR_ROT_OFFSET_FROM_FR	Rotary axes offset in WO with active TCARR
=1	Rotary axes offset for orientable tool carrier is automatically accepted for active work offset on activation of the orientable tool carrier for the rotary axes.

MD20196 $MC_TOCARR_ROTAX_MODE	Rotary axis setting with undefined position with active tool carrier (TCARR). (Default: 2H)
Bit 1= 0	The position of the rotary axis will be “0“ (C0); a possible necessary rotation is preformed through the specified frame.
Bit 1= 1	The rotation is performed by means of the rotary axis of the orientable tool carrier. The resulting frame will no longer include a rotation

Notes

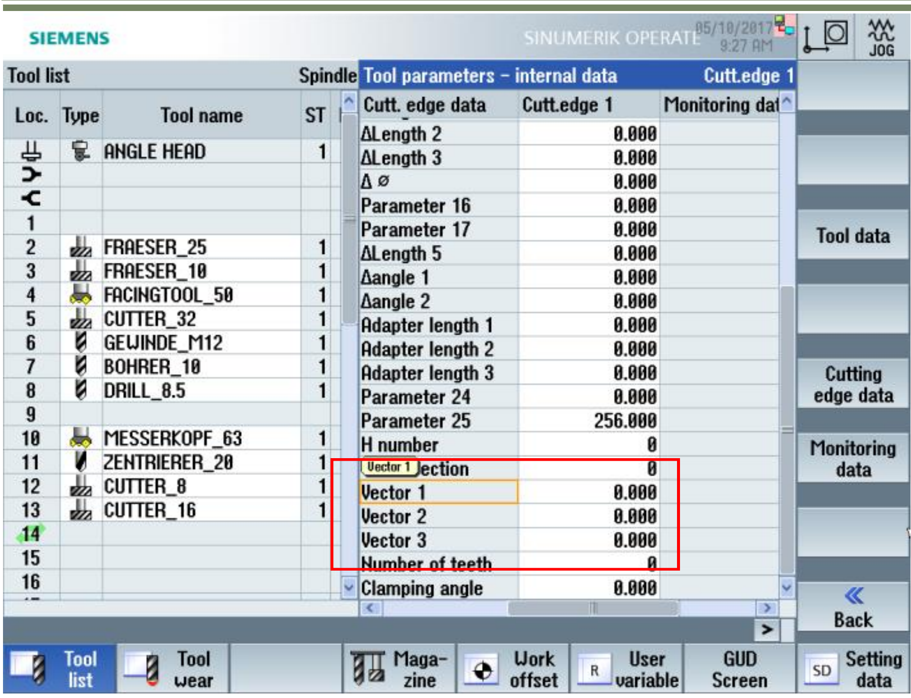
Working with angular tools

Angular tools can be created and managed in 840D sl SINUMERIK Operate with tool type130.

The tool lengths for angular tools of type 130 are entered in tool list. If an angular tool is also used on a swiveled machining plane in combination with CYCLE800, the basic tool orientation must be enabled with the following machine data:
MD18114 $MM_ENABLE_TOOL_ORIENT = 2.

This enables a direction vector to be transferred to the tool in the tool parameters $TC_DPV3[n] to $TC_DPV5[n] (n = internal tool number).

The G-code commands TOROT (G17), TOROTY (G18), and TOROTX (G19) must be programmed after the tool change so that the basic tool orientation is updated.



Note:

Changing machine data **MD18114** cause the buffered memory (SRAM) to be reorganized. After changing this MD it is recommended to create a NC data backup, otherwise NC data will be lost.

Notes

MD18114 $MM_ENABLE_TOOL_ORIENT	Assignment of orientation to cutting edges (Basic tool orientation).
= 0	No basic tool orientation active
= 2	Basic tool orientation active (used for swiveling with angle head tools).

MD20110 $MC_RESET_MODE_MASK	Initial control setting for RESET and end of part program
Bit 14 = 1	Calculation of the Basic– and System frames, Example: 4041H.

MD20112 $MC_START_MODE_MASK	Basic control setting for START and part program.
400 H	

MD20126 $MC_TOOL_CARRIER_RESET_VALUE	Active tool carrier at RESET.
= 0	No active tool carrier on RESET
= > 0	(n) Tool carrier with number n active at RESET. MD20126 is written to in CYCLE800. CYCLE800() corresponds to deselecting tool holder (MD20126 = 0).

MD20150 $MC_GCODE_RESET_VALUES[ ]	Delete position, G group.
[41] = 1	Delete position, G group 42 on TCOABS
[51] = 2	Delete position, G group 52 on PAROT
[52] = 1	Delete position, G group 53 on TOROTOF
[52] = > 1	Delete position, G group 53 on TOROT, TOROTY or TOROTX Used for machine kinematics, types "T" and "M". See parameter $TC_CARR34

Note:

If, after a RESET from the NC, a frame must be calculated in the tool direction, then ***MD20150 $MC_GCODE_RESET_VALUES[52]*** can be set to a value > 1.

Applications:

- Machine kinematics with Hirth gearing
- Angular tool with basic tool orientation

Notes

Setting the workpiece, tool and rotary table reference

Using the following machine data, you can set the workpiece, tool and rotary table reference system frames or you can influence the behavior of the system frames at reset or power on/off.

Application:

System frames can be active after a reset or Power On in order, for example, to retract a drill from a swiveled position without causing a collision. When using measuring or swivel in JOG, the workpiece reference must be active at RESET and not deleted.

MD24006 $MC_CHSFRAME_RESET_MASK		Active system frames after RESET
Bit 4	System frame workpiece reference	
= 0	Not active	
= 1	Remains active	

MD24007 $MC_CHSFRAME_RESET_CLEAR_MASK		Clear system frames after RESET
Bit 4	System frame workpiece reference	
= 0	Do not delete	
= 1	Delete	

MD24008 $MC_CHSFRAME_POWERON_MASK		Reset system frames after power on.
Bit 2	System frame rotary table reference (PAROT)	
= 0	Do not reset	
= 1	Reset	
Bit 3	System frame tool reference (TOROT,..)	
= 0	Do not reset	
= 1	Reset	
Bit 4	System frame workpiece reference	
= 0	Do not reset	
= 1	Reset	

Notes

MD24080 $MC_USER_FRAME_POWERON_MASK	Settings for settable frames.
Bit 0	System frame workpiece reference
= 0	Settable work offset via power on not active.
= 1	Last active settable work offset remains active after power on if MD20152 $MC_GCODE_RESET_MODE[7] = 1.

Application:
Work offset G5xx, including all rotations, should remain active after Power On.

MD28082 $MC_MM_SYSTEM_FRAME_MASK	Setting-up system frames (SRAM)
Bit 2 = 1	Rotary table reference ($P_PARTFRAME)
Bit 3 = 1	Tool reference ($P_TOOLFRAME)
Bit 4 = 1	Workpiece reference ($P_WPFRAME)

MD28083 $MC_MM_SYSTEM_DATAFRAME_MASK	Setting-up data management, system frames (SRAM)
Bit 2 = 1	Rotary table reference ($P_PARTFRAME)
Bit 3 = 1	Tool reference ($P_TOOLFRAME)
Bit 4 = 1	Workpiece reference ($P_WPFRAME)

SD42980 $SC_TOFRAME_MODE	Setting, frame definition for TOROT, PAROT
= 2000	Swivel (default value)

SD42974 $SC_TOCARR_FINE_CORRECTION	Fine offset TCARR (swivel data record)
= 0	No fine offset of the swivel data set vectors.
= 1	Fine offset of the swivel data set vectors. The parameters of the swivel data set from $TC_CARR41[n] n…number of the swivel data set and higher apply.

Setting for block search in AUTOMATIC mode

MD11450 $MN_SEARCH_RUN_MODE	Settings for block search
Bit1 = 1	Activate PROG_EVENT.SPF after block search. This means that for a block search, the rotary axes of the active swivel data set are pre-positioned.

Notes

Axis machine data for modulo rotary axes

MD30455 $MA_MISC_FUNCTION_MASK	Axis functions
Bit 0	Modulo rotary axis programming
= 0	No modulo rotary axis programming (e.g. 0 to 359.999 degrees)
= 1	Modulo rotary axis programming (e.g. -180 to 180 degrees)
Bit 2	Positioning, rotary axis
= 0	As programmed
= 1	Along the shortest path Application: With the setting, bit 2=1 then e.g. for G90 with DC, rotary axis C travels along the shortest path. Additional information can be taken from the Chapter "Manufacturer cycle CUST_800.SPF".

Display or hide cycle alarms for CYCLE800

MD55410 $MC_MILL_SWIVEL_ALARM_MASK	Activate fault evaluation CYCLE800 (Default 0H)
Bit 0	Activates fault 61186
= 0	Hide fault 61186 "Active work offset G%4 and basic (basic reference) contains rotations" (default).
= 1	Display fault 61186
Bit 1	Activate fault 61187
= 0	Hide fault 61187 "Active basis and basis reference (G500) contain rotations" (default).
= 1	Display fault 61187

Swivel in JOG mode

MD32010 $MA_JOG_VELO_RAPID[AX]	Rapid traverse in JOG, rotary and machine axes, that should travel for swivel function in JOG (AX = axis name).
= 10000	Rapid traverse for swivel function in JOG

MD11602 $MN_ASUP_START_MASK	Ignore stop conditions for ASUB
Bit 0= 1	ASUB, self-locking (used for the swivel function in JOG)

MD11604 $MN_ASUP_START_PRIO_LEVEL	Priorities $MN_ASUP_START
= 64	Corresponds to 100 (used for the swivel function in JOG)

Notes

Manufacturer cycle CUST_800.SPF

Customization for CYCLE800

General explanation

This cycle is designed for the machine tool manufacturer or end user to customize CYCLE800, based on the requirements of the machine tool.
Events before and after swivelling can be modified upon requirement.

During swivelling, all axis positions are traversed using the CUST_800.SPF program which is always called from CYCLE800 swivel cycle or E_TCARR (ShopMill) or F_TCARR (ShopTurn).

In cycle CUST_800.SPF, the function markers (_M2: to _M59) are prepared and documented. Also refer to the diagram “Structure of swivel cycles” section 5.31 – 5.32

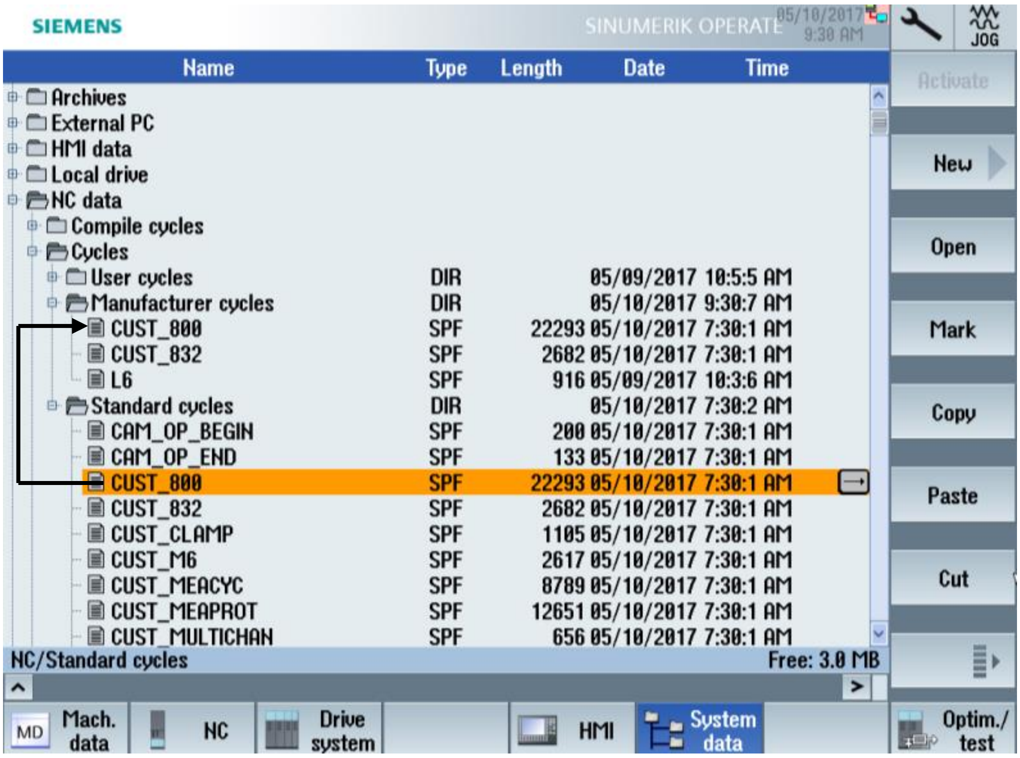
Procedure for customization

If you modify the cycle CUST_800.SPF, proceed as follows:

For a modification by the machine manufacturer copy **CUST_800.SPF** from the standard cycles directory into the manufacturer cycles directory.

For a modification by the user copy **CUST_800.SPF** from the standard cycles directory into the user cycles directory.

On start of CYCLE800.SPF, the program looks first in the user cycles directory for the subprogram,
if not there, then in manufacturer cycles and last in the standard cycles directory.



Notes

User variables of the CUST_800.SPF manufacturer cycle

CUST_800 (INT _MODE, INT _TC1, REAL _A1, REAL _A2, INT _TC2, REAL _T_POS)
SAVE DISPLOF

Local user variables LUD	Description
_MODE	A jump is made to markers _M2 to _M59
_TC1	Number of the new swivel data record
_A1	Calculated angle of rotary axis 1 in CYCLE800. Also used as transfer parameter for E_SWIV_H for positioning of the first rotary axis
_A2	Calculated angle of rotary axis 2 in CYCLE800. Also used as transfer parameter for E_SWIV_H for positioning of the second rotary axis
_TC2	Number of the new swivel data record for automatic swivel data change with JobShop function, otherwise only feed evaluation in percent (%) for swivelling in JOG mode
_T_POS	Incremental position during retraction in the incremental tool direction (see marker M44, _M45)

Channel-specific user variables GUD	
_TC_FR	ONES-position: Retract mode: =0: no retract =1: Z or fixed point 1 =2: Z, XY or fixed point 2 =4: Tool direction max =5: Tool direction inc THAUSANDS-position: =0: Swivel in AUTO =1: Swivel in JOG
_TC_N_WZ	=0: Tool tracking off =1: Tool tracking on
_TC_A_WZ	=0: Do not align tool with swivel axes =1: Align tool with swivel axes
System variables	
$P_EP[_AX1]	System variable for actual rotary axis position WCS of 1st rotary axis
$P_EP[_AX2]	System variable for actual rotary axis position WCS of 2nd rotary axis

Notes

Example retract prior to swivelling:

If the CUST_800.SPF cycle is not modified, the Z-axis (marker _M41) or the Z-axis followed by the X-, Y-axes (marker _M42) are first traversed in the Machine to the positions when retracting prior to swivelling. The freely available position values are specified in the system variables $TC_CARR38[n] to $TC_CARR40[n]. When retracting, the active tool cutting edge is deselected (D0) and is reselected after retraction.

If retraction in the tool direction had been declared, the tool axis is retracted to the software end position (marker _M44) or by an incremental distance away from the tool in the tool direction (marker _M45). The tool lengths are taken into account accordingly.

Section from CUST_800

```
_M41: ; Axis Z MCS-Fixpoint
 _DD=$P_TOOL
 IF _TC2>0
  DO $AC_OVR=_TC2
 ENDIF
 SBLON
 N800410 SUPA D0 G0 G40 G60 Z=AC($TC_CARR40[_TC1])
 D=_DD
 SBLOF
 ;STOPRE
 GOTOF _MEND
 ;
```

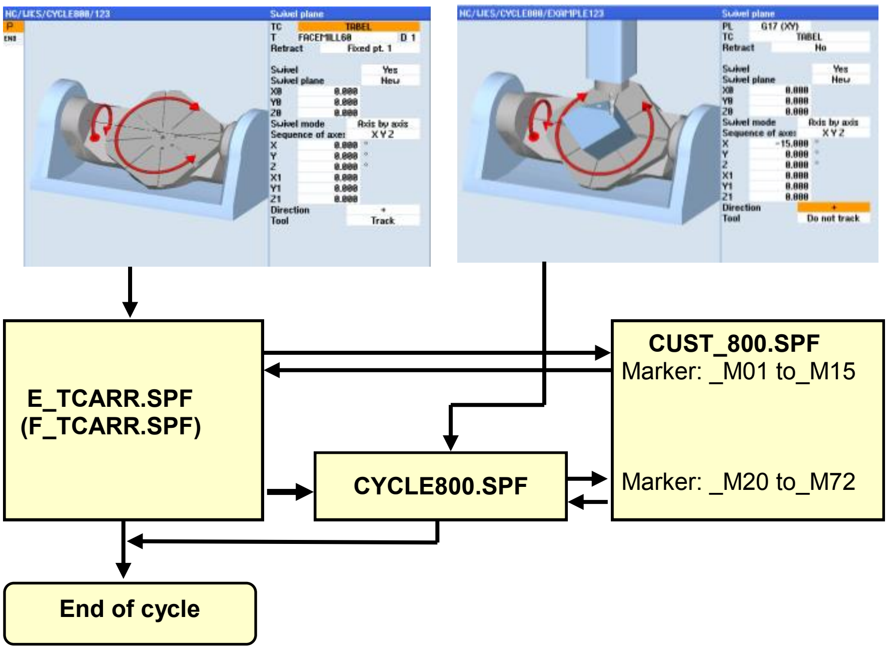
```
_M42: ; Axis Z and after axes XY MCS-Fixpoint
 _DD=$P_TOOL
 IF _TC2>0
  DO $AC_OVR=_TC2
 ENDIF
 SBLON
 N800420 SUPA D0 G0 G40 G60 Z=AC($TC_CARR40[_TC1])
 IF _TC2>0
  DO $AC_OVR=_TC2
 ENDIF
 N800421 SUPA X=$TC_CARR38[_TC1] Y=$TC_CARR39[_TC1]
  D=_DD
 SBLOF
 ;STOPRE
 GOTOF _MEND
 ;
_M44:  ; Tool direction retract maximal
 IF _TC2>0
  DO $AC_OVR=_TC2
 ENDIF
 SBLON
 N800440 G0 G60 AX[$P_AXN3]=AC(_T_POS)
 SBLOF
 ;STOPRE
 GOTOF _MEND
………..
```

Notes

Basic structure of swivel cycles

Input screen form

CYCLE800 ShopMill/ShopTurn ***CYCLE800 ProgramGUIDE G-Code***



Note:

The parameters are only valid in conjunction with the relevant jump label in CUST_800.SPF (see program structure on following page).

If Inch is the basic system of the machine, the CUST_800.SPF program must be modified accordingly (note the machine manufacturer's instructions).

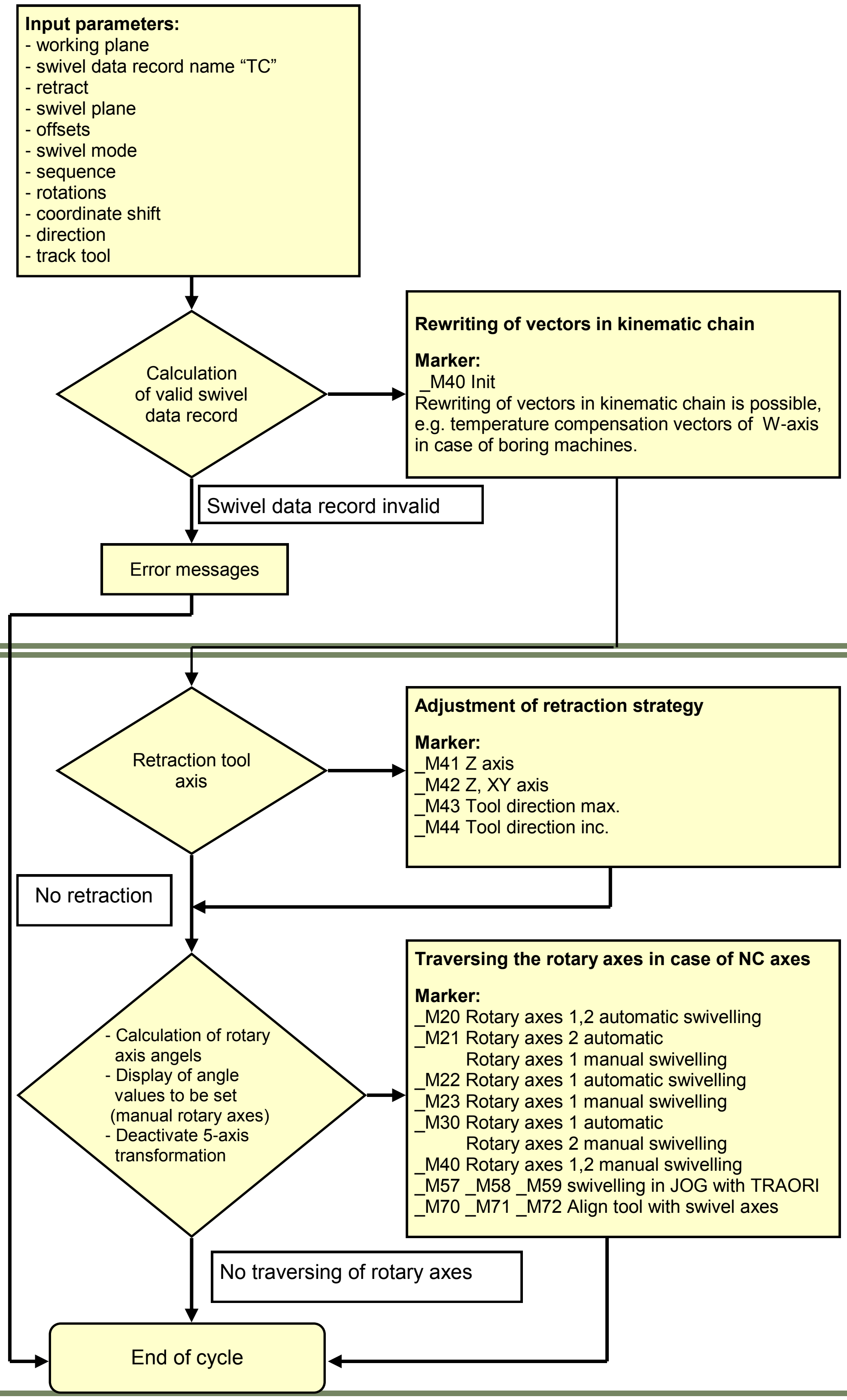
The subprograms **E_SWIV_H, E_TCARR, E_SP_RP** apply only for ShopMill.
The subprograms **F_SWIV_H, F_TCARR, F_SP_RP** apply only for ShopTurn.

(also see structure diagram on next page)

Notes

Structure diagram swivel cycles

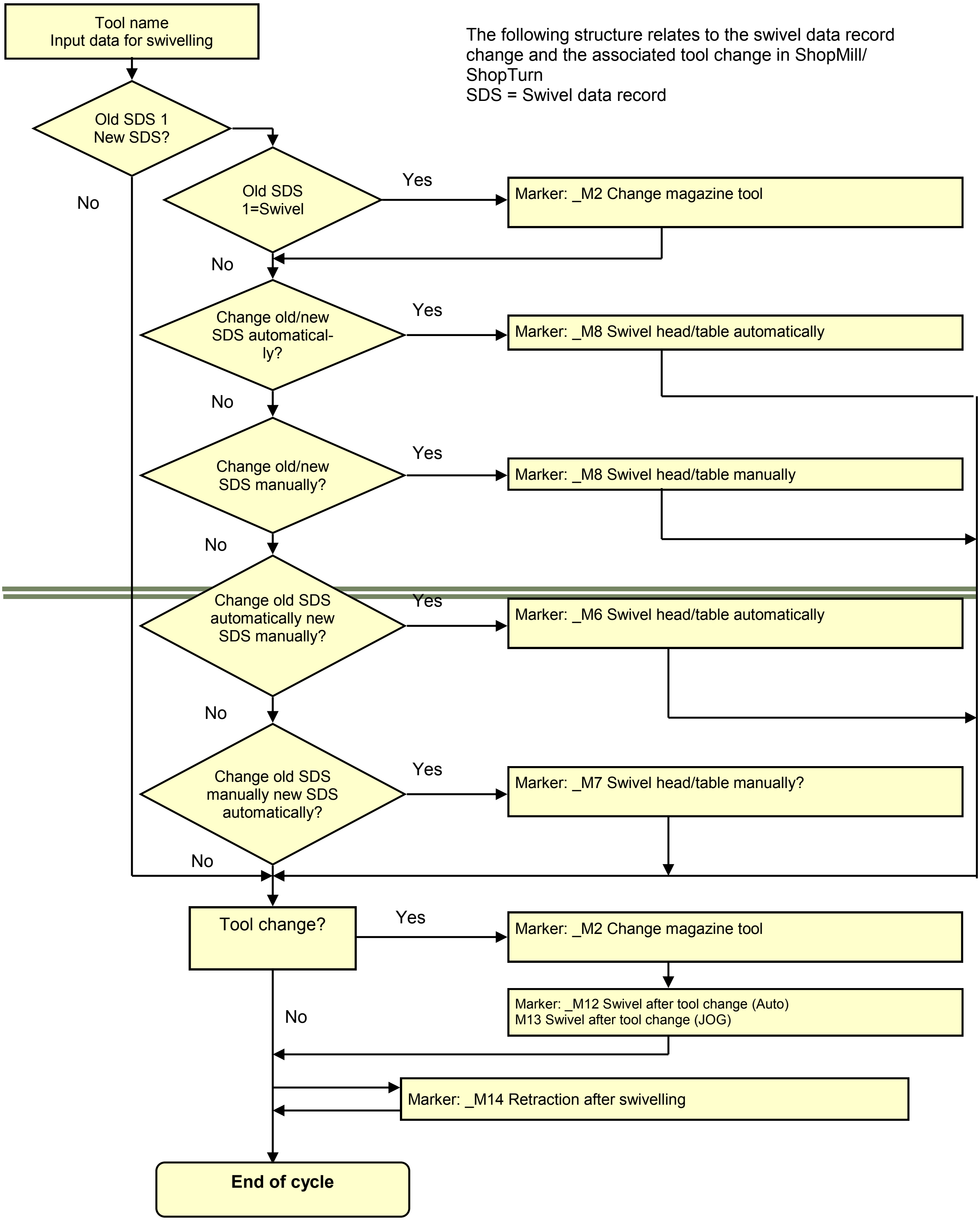
CYCLE800.SPF **CUST_800.SPF**



Notes

E_TCARR.SPF (F_TCARR.SPF) **CUST_800.SPF**

The following structure relates to the swivel data record change and the associated tool change in ShopMill/ ShopTurn
SDS = Swivel data record



Notes

Notes on markers

Markers _M2 to _M13 (only relevant for ShopMill/ShopTurn)

If the swivel data set (SDS) or tool is changed, the linear axes are retracted using the last retraction mode (modal).

If this behavior is not desired in ShopMill/ShopTurn, the corresponding calls must be commented out with a semicolon (;). The **E_SWIV_H** or **F_SWIV_H** cycle is called in milling/turning (see markers _M2 to _M9) in the CUST_800.SPF manufacturer cycle.

Parameter E_SWIV_H (Par1, Par2, Par3)

- Par1: Number of swivel data set (_TC1)
- Par2: Angle of 1st rotary axis
- Par3: Angle of 2nd rotary axis

Modification examples

If the rotary axes (swivel head/table) are not to be positioned during swivel data change / tool change, the call of the **E_SWIV_H** cycle can be commented out at the relevant markers. If the rotary axes are to move to a certain position, an angle value can be transferred to parameters Par 2, Par 3.

Markers _M14 to _M15 (only relevant for ShopMill/ShopTurn)

Depending on the values of the retraction plane and the programmed swivel plane, it is possible that the linear axes now also travel the swivelled retraction plane while running up from the current position to the software limit switches after a block search. To avoid this problem, marker _M14 in the CUST_800.SPF is called after swivelling. The **E_SP_RP(30)** cycle preset there runs up to the retraction plane, whereby travel may be along the software limit switches. An appropriate retraction after block search can be set at marker _M15.

Markers _M20 to _M31

Markers _M20 to _M31 are distinguished by machine kinematics with two rotary axes or one rotary axis. A distinction is also made between automatic rotary axes (known to NCU) and manual (semi-automatic) rotary axes. There is only ever one valid marker for swivelling with the active swivel data record.

Marker _M35

Run through _M35 for block search and a swivel data set with manual rotary axes.

Marker _M46

Retraction before swivelling after a block search can be set at marker _M46. Variable _E_VER is 1 if it is a milling technology program.

Notes

Markers _M57 to _M59

Markers _M57 to _M59 are used for swivelling in JOG mode and active 5-axis transformation.

Note on "Tool tracking"

“Tool tracking” requires that a 5-axis transformation is set up, which is equivalent to the corresponding swivel data set. The programming section for "Tool tracking" is integrated in the markers _M20, _M21, _M22 and _M30. The first 5-axis transformation is called with TRAORI.

Note on tool change + swivel

In general, the swivel (CYCLE800) and tool change functions for a machine are independent of each other. Thus, the swivelled working plane can be retained in a technological sequence with multiple tools, e.g. centering, drilling, tapping.

If the rotary axes of the active swivel data set are involved in the mechanical sequence of the tool change or have to be retracted, this must be taken into account in the tool change program. After the tool change, the rotary axis positions are approached as prior to the tool change. If linear axes (geometry axes) are also involved in the tool change, the rotations in the NC (swivel frame) must not be deleted. Rather, the linear axes can be positioned as machine axes using the G153 or SUPA commands.

Note on swivelling without active tool compensation

If swivelling the rotary axes without active tool cutting edge (D0) is not possible, then you can adapt this in cycle CUST_800.SPF:

```
_M40:
IF ((NOT $P_TOOL) AND _TC1)
LOOP
MSG ("no tool cutting edge active")
M0
STOPRE
ENDLOOP
ENDIF
GOTOF_MEND
```

Notes

M103

Sinumerik
Operate

5-axis transformation TRAORI

Module Description:

This Module shows the application range with the 5-axis transformation **TRAORI**. With help of practical programming examples and explanation through graphical illustrations you get to learn how to use and program 5-axis transformation with TRAORI. In addition this manual shows important settings for configuring the 5-axis transformation.

Module Objective:

Objective of the module:

This module is aimed at experts working in the mould & die and aerospace industry who need to get acquainted with the key basic principles of multi-axis machining. Building on this, it also provides machine users with practical tips, so that they can organize their work efficiently and provides programmers with an insight into 5-axis transformation function **TRAORI** on the control and the CAD/CAM system.

Content:

Theory

Programming 5-axis with TRAORI

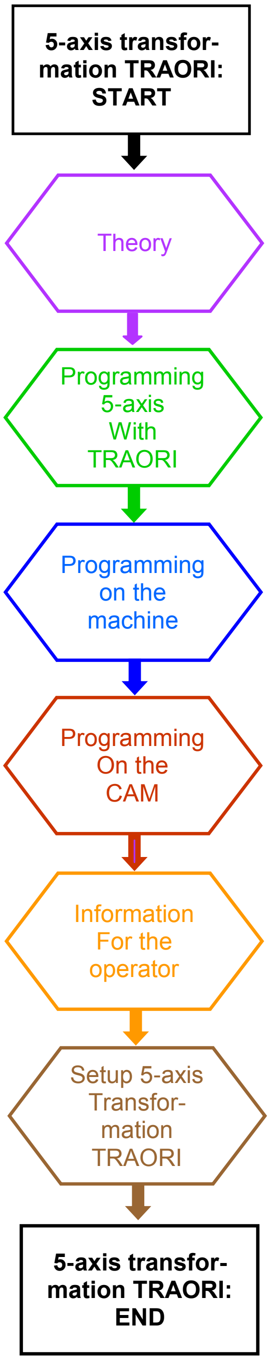
Programming on the machine

Programming on the CAM

Information for the operator

Setup 5-axis transformation TRAORI

840D sl SINUMERIK Operate


5-axis transfor-
mation TRAORI:
START
Theory
Programming
5-axis
With
TRAORI
Programming
on the
machine
Programming
On the
CAM
Information
For the
operator
Setup 5-axis
Transfor-
mation
TRAORI
5-axis transfor-
mation TRAORI:
END

Notes

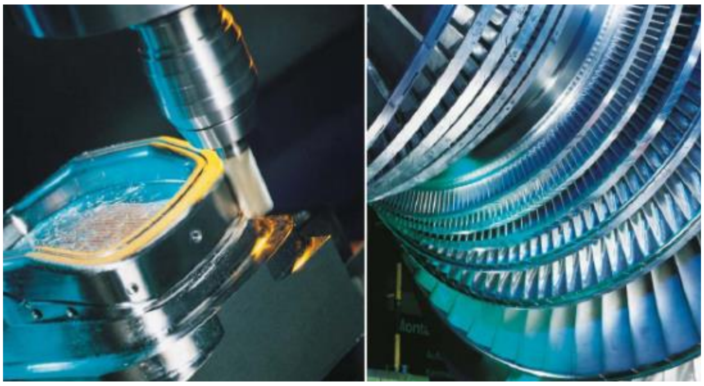
Theory

Introduction

With multi-axis machining, the main objectives are to achieve perfect surface quality, precision, and speed without any need for re-machining. Within this context, workflow is typically characterized by the CAD-CAM-CNC process chain.

840D sl Operate is equipped with powerful, advanced functions which, when intelligently used, make the whole process of 5 -axis machining and programming considerably easier while at the same time improving the results of production.

Various application areas of 5-axis machining and their dedicated functions that have been specifically developed for each area, are introduced and considered in this module.



Picture 1.1: Application range of 5-axis machining

Requirements of the 5-axis machining

Design standards in all application areas are becoming increasingly more demanding.

Expectations in terms of ergonomics, the air drag coefficient (CW value) or simply aesthetic appeal are creating a need for more complex surface geometries to be achieved in less time and with greater precision.

The design primarily comes from CAD systems, the machining programs from CAM stations.

Nevertheless, the skilled machine tool operator still has overall responsibility (in terms of technology) for the quality of the mold and the complete tool.

The 840D sl system, is perfectly suited to the demands of 5-axis machining as well as HSC applications:

- Simple to operate
- User-friendly programming at the machine
- Optimum performance throughout the CAD - CAM - CNC process chain
- Maximum control over quality at the machine
- Optimized 5-axis functions

Notes

Application range for 5-axis machining

Depending on the application, the requirements imposed on the control will vary and a whole range of different functions may be demanded.

Within this context, 5-axis machining can be broken down into three broad areas:

- Free-form surfaces (mould making)
- Turbine and driving gear components (impellers, blisks)
- Structural parts (aviation industry)

The 840D sl provides optimum support for each of these areas

Free-form surfaces, mold & die making industry

Driving gear and turbine components, e.g. impellers, blisks

Structural parts, aviation industry

Each of the above shown 5-axis machining applications requires a certain type of 5-axis machine, that is best suitable for the application and depends in most cases on size and weight of the workpiece and required rotary axis traversing range of the machine tool. 5-axis transformation functions are therefore a requirement for a efficient manufacturing process.

Notes

Kinematic transformation TRAORI

General explanation of the function

The kinematic transformation **TRAORI** is a **dynamic transformation** that allows simultaneous traversing of 4 or **5** and for some special applications (e.g. tape laying machines) up to 6 axes.
The most common type of transformation is 5-axis.

The use of the 5-axis transformation is to compensate movements of the tool tip during machining, which result from changes in orientation through rotary axes movement, by means of compensating movements of the geometry axes. The orientation movement is therefore decoupled from the movement on the workpiece contour. This allows an axially symmetrical tool (e.g. milling cutter) to be oriented in any desired relation to the workpiece in every point of the machining space.

Prerequisite is a machine, which can control tool movements in 5 axes simultaneously. The three familiar linear axes X Y Z plus an additional two rotary axes. The two rotary axes involved in this transformation have different kinematics solutions that are part of a **Kinematic chain.** The kinematic transformation requires therefore information about the machine design (kinematic) that is defined in a set of **trafo machine data**.

Tool orientation can be specified in two ways:

- **Machine-related orientation**
  The machine-related orientation is dependent on the machine kinematics.

- **Workpiece-related orientation**
  The workpiece-related orientation is "independent" of the machine kinematics and is programmed with:
  – Cartesian direction vectors
  – Orientation angles according to the Euler principle
  – Orientation angles according to the RPY principle
  – Lead and tilt angles with surface normal relative to the path tangent

In workpiece-related orientation the tool direction is described in the workpiece coordinate system. The orientation of a specific component of the tool to the workpiece can be programmed. In most cases, this is along the tool axis (Z for G17) to the tool tip; this is also called **TTP programming** (TTP = Tool Tip Point).

The movements of the tool are always programmed in the Cartesian coordinate system; any programmed or set **FRAMES** rotate and move this system relative to the basic system. The tool length compensation is included in the calculation for the machining. The kinematic transformation then converts this information into motion data for the real machine axes. For the programming of a workpiece-related orientation, the NC program is independent of the machine kinematics.

The programming of the path and path velocity is performed in the same way as for 3-axis programming. The tool orientation is also programmed in the individual motion blocks.

The real-time transformation calculates the resulting movements of all five axes. This eliminates the calculation of the feedrate for the rotary axes at a change in orientation (rotary axes are defined as synchronized axes in the FGROUP).

Notes

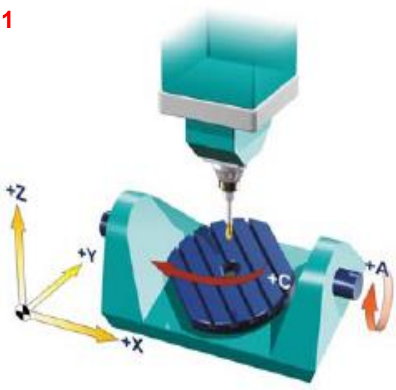
What moves and how?

The 5-axis transformation package covers the three possible basic machine configurations which differ in terms of tool and workpiece orientation:

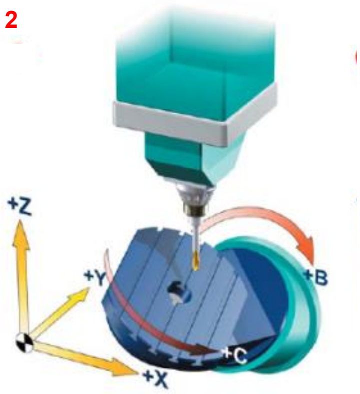
- Orientation of workpiece with 2 rotary axes in the table (type 40)
- Orientation of tool with 2 rotary axes in the head (type 24)
- Orientation of workpiece & tool with mixed kinematics (type 56)

The most common ones are shown here:

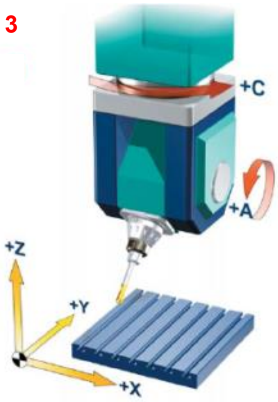
1



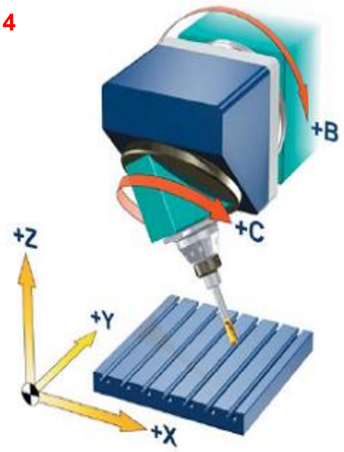
2



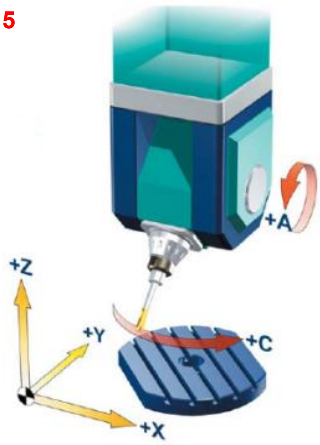
3



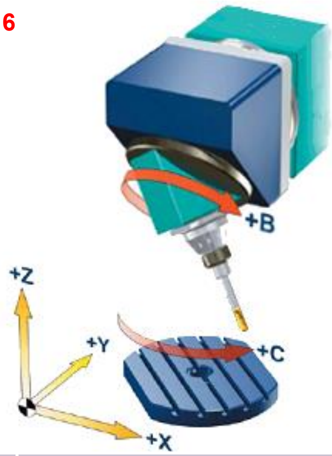
4



5



6



Type 40	2 Rotary axes in the table	1 rotary swivel table	2 nutated* rotary swivel table
Type 24	2 Rotary axes in the head	3 rotary fork head	4 nutated* rotary swivel head
Type 56	1 Rotary axis in the head 1 Rotary axis in the table	5 Fork head + rotary table	6 nutated* rotary head + table
Type 72	Kinematic from orientable tool carrier (TCARR)	Type description with P,T, M in $TC_TCARR23	

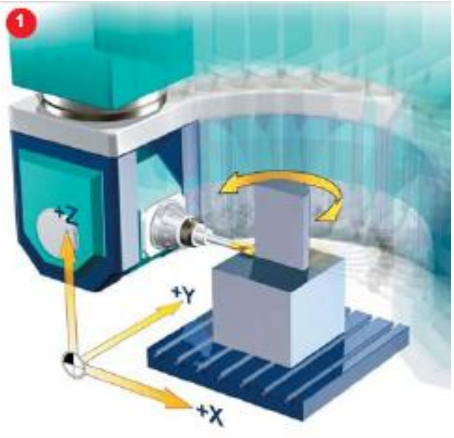
*Term: If the rotary axis is not perpendicular to a linear axis, it is referred to as a ”**nutated**” axis.

Notes

Effect of the kinematics on the machine movement

Depending on the machine kinematics, completely different machine movements may be required to machine the same surface and therefore different NC programs. In the following example a cylinder jacket surface is to be machined. It is quite obvious to notice that the machine kinematics in example **2** are much more suitable for the production of this workpiece.

Different kinematics



Sequence of movements for a head/head kinematic (Type T)

For one tool path around the circumference of the cylinder jacket, a point to point semicircle in X/Y has to be described in the program. At the same time the movement of the tool must rotate around the Z-axis during machining so that the cutter Is always perpendicular to the surface.

Sequence of movements for table/table kinematic (Type P)

For one tool path around the circumference of the cylinder jacket the table has to swivel the A-axis 90°. The C-axis turns from +90° to -90° and the tool moves along the Y-axis.

Note:

With active 5-axis transformation it is possible to program tool orientation independent of the machine kinematics. This results in a kinematic independent NC-program, that can be executed on any 5-axis machine (any kinematic type P,T or M).

Notes

Influence of tool length on machine axis movement

Different tool length

The longer the tool in the example, the larger the compensation for the traversing movements of the axis slides.

If the program has been created using a CAM-System without programming of 5-axis transformation, it will need to be recalculated whenever there is a change in the tool length.

For this reason the control system needs to know that a calculated NC program is compatible with any tool length.

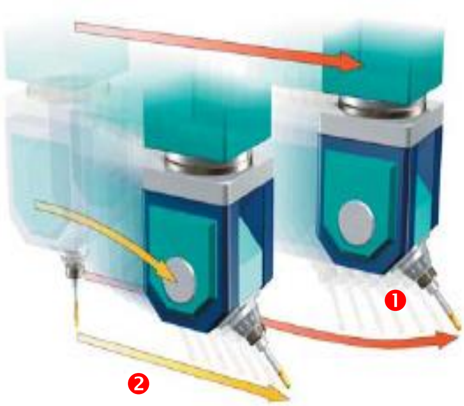
This active tool length compensation is integrated into the SINUMERIK system and does not need to be given any consideration during programming.

Note:

Depending on the tool length, it can happen that the traversing range of an axis is overrun and the axis runs on to the limit switch, even though all values in the CNC program are within the work area. A distinction is made here between the "**gross**" and "**net**" work area

Influence of a change in tool orientation on the movement of the linear axes

Change of tool orientation



Simple movements can be transformed into a complex curve if the tool orientation is changed simultaneously during traversal of the linear axes. To mill a straight line with no change of tool orientation, the tool holder describes a straight line. If the orientation changes at the same time, the tool tip describes a curve (❶).

However, In order to mill a straight line with change of tool orientation, the movement of the tool tip must be compensate for by the kinematics in such a way that the tool tip describes the desired straight line as the orientation changes (❷).

The example with the yellow linear movement (❷), works with active 5-axis transformation TRAORI .

Note:

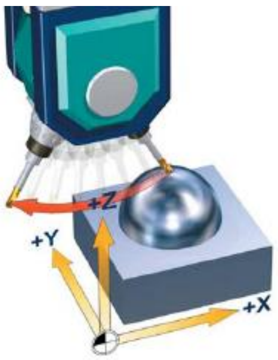
In order for all these requirements to be taken into consideration, a transformation is needed that will transform a kinematic independent NC program to a 5-axis machine of any kinematic type, taking into account both, the active tool length compensation and the change in tool orientation. All this is realized with the **TRAORI** function.

Notes

What does the TRAORI function do ?

- Compensation movements in X, Y and Z are being calculated with a change in tool orientation, where the tool tip remains fixed to a position. NC programs become more compact and handling of programs becomes easier.
- The current tool length and work offset is taken into consideration when calculating the traversing movements.
- Tool length and workpiece zero offsets can be altered on the control at any time and are taken into consideration during program run.
- Programmed feedrate relates to the tool tip point (TTP) and inverse time calculation for rotary axes is not necessary.

Without TRAORI



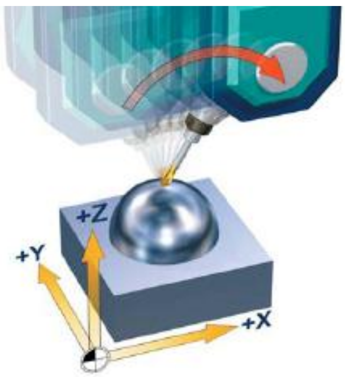
The control system disregards the tool length. It rotates around the pivot point of the rotary axis **1**.

The tool tip moves out of position and does not remain fixed. Contour violations occur.

Compensation movements for tool length from the tool tip to the swivel point have to be calculated by the CAM System.

Tool length offset compensation on the control is therefore not possible.

With TRAORI



The control system only changes the orientation of the tool, the position of the tool tip emains fixed.

The necessary compensating movements **2** for the linear in axes are calculated automatically.

The control compensates with active **TRAORI** automatically for changes in tool length and shift of workpiece location.

All this allows greater flexibility for machining of complicated 5-axis parts.

Note:

Whit active 5-axis transformation, the positional data (X, Y, Z) always relates to the tip point (TTP). Changing the positions of the rotary axes involved in the transformation causes compensating motion of the remaining linear axes, which means at the position of the tool tip remains unchanged.

Notes

Programming the 5-axis transformation with TRAORI

- ***Machine-dependent CNC programs (machine-related orientation)***

With machine-dependent CNC programs, the actual rotary axes (A, B, C) of the machine are programmed directly. This means that the CNC program does not contain any tool orientation in the form of a direction vector, but instead a machine-related rotary axis position.
The CNC program can therefore only run on the kinematics specified for the program.
This is called machine-dependent programming.

TRAORI is generally called in the CNC program output by the CAM system. The CNC program then only contains:

- The coordinates of the point to be approached in **X,Y,Z**
- The actual rotary axis positions **A, B, C** in relation to the machine kinematics (MCS).

Disadvantage

A kinematics-specific postprocessor is required for the 5-axis machining with TRAORI, in which an exact simulation of the machine kinematics is integrated in order to be able to calculate the kinematics-specific rotary axis position of the machine.

Advantage

Because of the programming of the real machine rotary axes, possible collisions can be detected in the G code simulation in the CAM even without VNCK (Virtual NC Kernel).

Machine-independent CNC programs (workpiece-related orientation)

With machine-independent CNC programs, CNC programs are required that can be executed on different machine kinematics.

Usually, CNC programs are created that are based on the workpiece, i.e. all tool positions are related to the workpiece coordinate system (WCS). In order that such a CNC program can be executed on the machine, the positions must be transformed to real axis movements, i.e. converted to the machine coordinate system (MCS). This is performed with the **TRAORI** function.

TRAORI is generally called in the CNC program output by the CAM system. The CNC program then only contains:

- The coordinates of the point to be approached in **X,Y,Z**
- The tool orientation to this point in the form of direction vectors **A3=, B3=, C3=**, or
- Tool orientation as Euler or RPY orientation angles **A2=, B2=, C2=** or
- **LEAD/TILT** angles with surface normal **A4= C4= B4=** or **A5= C5= B5=** relative to the path tangent

Advantage

The use of kinematics-specific postprocessors is not required for the 5-axis machining with TRAORI in this case.

Notes

Programming syntax:

TRAORI (<n>)

TRAORI (<n>, <X>, <Y>, <Z>, <A>, <B>, <C>)

TRAFOOF

NC command	Description
TRAORI	Activates the first programmed orientation transformation
TRAORI (<n>)	Activates the orientation transformation programmed with "**n**"
<n>	Transformation number (n = 1,2,3,4), TRAORI (1) activates the 1.Orientation transformation. The number can be omitted if the first orientation transformation is to be activated.
<X>, <Y>, <Z>:	Components of orientation normal vector to which the tool point refers (e.g. angle head attachments). The direction normal vector is defined by parameters 2, 3 and 4. If the number of the first orientation transformation is omitted, a blank space has to be inserted instead of the transformation number, e.g. TRAORI (, 0., 1., 2.) in order to enable the parameters to be identified correctly when specifying an orientation. The orientation data is absolute; it will not be modified by any active frame.
<X>, <Y>, <Z>:	The absolute value of the vector is insignificant; only the direction is relevant. Non programmed vector elements can be set to zero.
<A>, <B>, <C>:	Programmable offset for orientation axes. The offset for orientation axes is defined by parameters 5, 6 and 7 When orientation transformation is activated an additional offset can be programmed directly for the orientation axes. Parameters can be omitted if the correct sequence is used in programming. Example: TRAORI(, , , ,A,B,C) ; if only a single offset is to be entered. As an alternative to direct programming, the additional offset for orientation axes can also be used and acts additive to the currently active work offset for the participating rotary axes of the transformation. The transfer of the WO is configured in ***MD24590 $MC_TRAFO5_ROT_OFFSET_FROM_FR_1***
TRAFOOF	Switches all transformations off

Notes

Orientation normal vector for the tool tip reference

Example:

TRAORI (1,0,0,1) ; Activates the 1st orientation transformation, with an orientation normal vector of the tool in Z-direction

TRAORI (1,0,1,0) ; Activates the 1st orientation transformation, with an orientation normal vector of the tool in Y-direction

TRAORI (1,1,0,0) ; Activates the 1st orientation transformation, with an orientation normal vector of the tool in X-direction

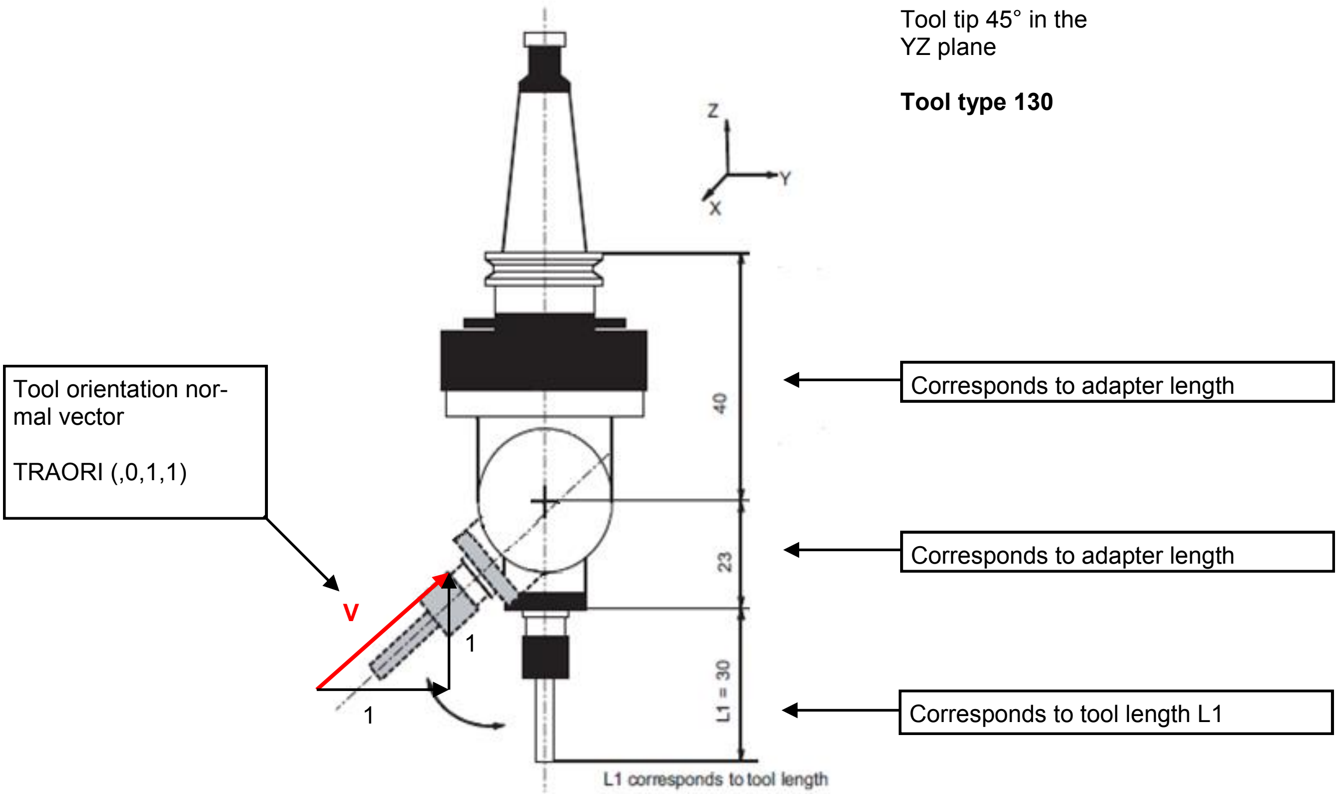
TRAORI (2,0,1,1) ; Activates the 2nd 5-axis transformation, with an orientation normal vector of the tool in YZ-direction (refers to an orientation normal vector of 45° in the YZ-plane).

Note:

The orientation normal vector of the tool tip does not lead to an orientation change of the tool itself, but only establishes the reference for the orientation normal vector to where the tool tip points.

Example:

Interchangeable swivel head with steep taper for incorporation in spindle.



Notes

Tool direction reference

Depending on the orientation direction selected for the tool, the active working plane (G17,G18, G19) must be set in the NC program in such a way that tool length offset works in the direction of tool orientation. Orientation transformation always points from the tool tip to the tool adapter.

Offset for orientation axes

When orientation transformation is activated an additional offset can be programmed for the orientation axes. When programming machine independent with specification of orientation vectors (A3=, B3=, C3=) or orientation angles (A2=, B2=, C2=) the specified offset modifies the actual machine rotary axis position.

Example AC table:

TRAORI (1,0,0,1,-10,20) ;1st Orientation transformation, tool orientation normal vector in Z-direction,
Offset of –10° for the 1st (A) and 20° for the 2nd rotary axis (C).
G54 G17 D1
ORIWKS
ORIAXES
G1 X0 Y0 Z0 A3=1 B3=1 C3=2 F10000
……

Result:

The programmed tool orientation vector results in following rotary axis positions in reference to the active WKS: A35.264 and C45 degrees.
Due to the programmed rotary axes offset the following rotary axes positions in reference to the MCS are being approached: A25.264 and C65° degrees.

Note:

In the general machine data ***MD10602 $MN_FRAME_GEOAX_CHANGE_MODE***, it is possible to reset the current active frame upon activation of 5-axis transformation. If it is not clear how to set this machine data then it is recommended to program the work offset after the TRAORI command (recommended setting =1).

Settings for MD10602
0= The current active frame is cancelled when geometry axes are switched over.
1= The current active frame remains active when geometry axes are switched over.

Attention! Only for Orientation programming A3=, C3= B3= and A2=, B2=, C2= a frame rotation is taken into consideration in the program main run.

The following machine data should be set for direct rotary axis programming:
MD24590 $MC_TRAFO5_ROT_OFFSET_FROM_FR_1 *(rotary axes offset for WO)*
1: A rotary axes offset in the storable work offset is automatically accepted with active 5-axis trafo.
0: A rotary axes offset in the storable work offset is not accepted with active 5-axis trafo.

Work offset – active [°]

				X	Y	Z	A	C
G54				260.000	200.000	150.000	0.256	0.012
Total WO				260.000	200.000	150.000	0.256	0.012

Notes

Milling with 3+2 or 5 axes ?

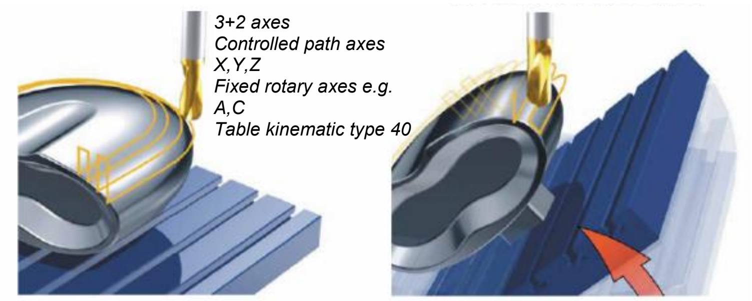
3+2-axis machining with change of tool orientation:

This type of machining uses TRAORI to orient the tool in relation to the machining surface. The workpiece coordinate system (WCS) is hereby not rotated. A change only takes place in tool orientation.

The tool is oriented from a start position through to an end position and the tool tip is traced during rotary axis positioning. The position of the rotary axes define the setting angle of the tool. Machining is executed with the three linear axes while the rotary axes remain static during the entire machining process. TRAORI takes the machine kinematics into account during swivelling, i.e. attention is paid to tool offsets and zero points.

Example 1:

On this machine the orientation of the tool or the position of the table can be set with the additional rotary axes A and C in the table.
In the picture on the left the cutting conditions deteriorate as the cutter moves up towards the top or to the side of the workpiece. In the picture on the left the cutter is working with optimum cutting conditions. In order to obtain optimum cutting conditions, the table is swivelled. To machine a freeform surface completely, the table can be swivelled several times in various directions.



3+2 axis machining in a inclined plane:

This type of machining uses frames
(e.g. ROT, TRANS), to shift and rotate the coordinate system (WCS) **1** in such a way, to align the work coordinate system with the machining plane.

With active “TRAORI” the tool tip maintains its workpiece reference and aligns the tool normal to the new machining plane **2**.

Machining is then executed in this plane with the three linear axes while the rotary axes remain static during the entire machining process.

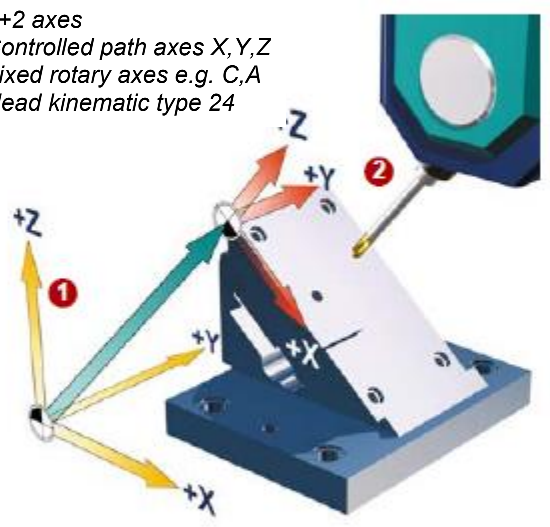
“TRAORI” takes the machine kinematics into account during swiveling, i.e. attention is paid to tool offsets and work offsets.

Notes

Example 2:

This example involves rotation of rotary axis C and A. The tool is oriented normal to the machining plane (XY plane G17). Drilling-/milling cycles can now be applied for inclined plane machining.

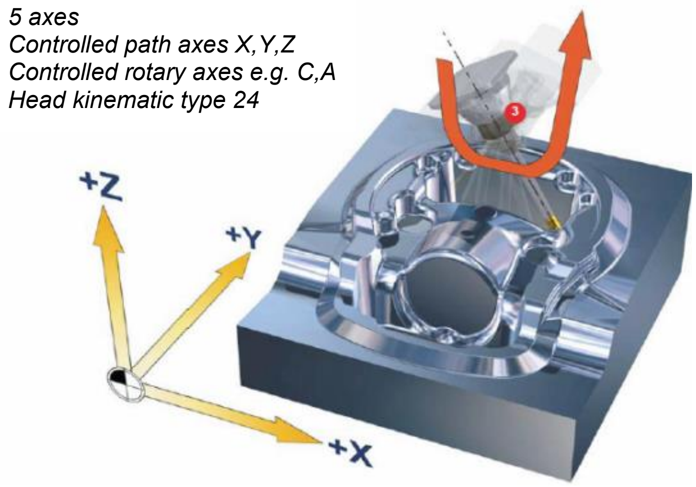
3+2 axes
Controlled path axes X,Y,Z
Fixed rotary axes e.g. C,A
Head kinematic type 24



Example 3:

5-axis simultaneous machining:

5 axes
Controlled path axes X,Y,Z
Controlled rotary axes e.g. C,A
Head kinematic type 24



For machining of deep cavities or frequent curvature, 5 controlled axes are required.

The orientation of the tool can be optimized along the entire path, synchronously to the linear movement of the tool.

In this way optimum cutting conditions are maintained throughout

This type of machining with TRAORI is a dynamic process. The rotary and linear axes can be moved simultaneously during machining. The tool can be continuously aligned with the surface while milling is in progress. The tool length is taken into

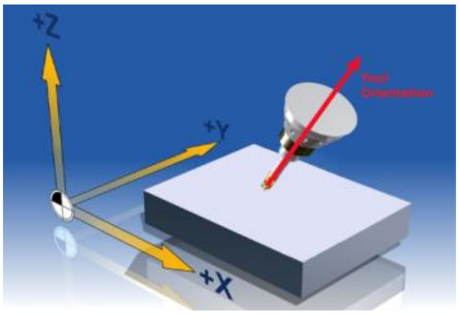
Notes

Programming 5-axis tool path with TRAORI

The basic questions

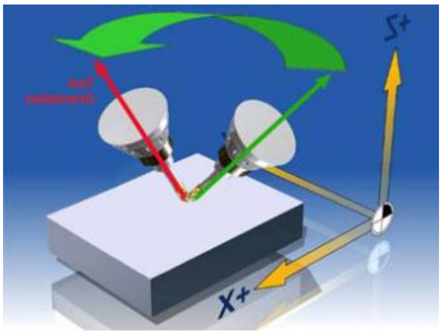
The 3 most commonly asked questions for 5-axis programming:

Programming of the tool orientation



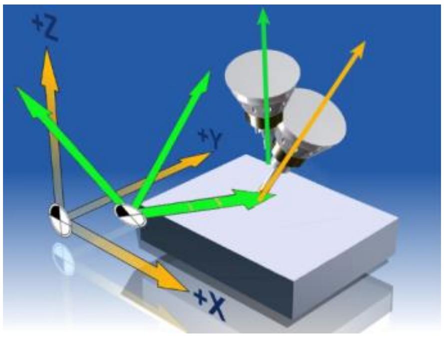
1. In which way is the tool orientation defined in the NC Program?

Interpolation of the orientation



2. In which way does the orientation (interpolation) from one NC-block to the next change ?

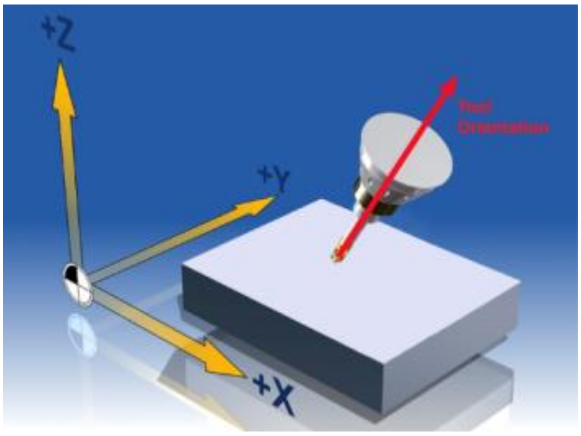
Orientation reference



3. To which coordinate system does the programmed tool orientation refer ?

Notes

Tool orientation



To accommodate machining operations involving a change in tool orientation in order to mill geometries located anywhere in space, the three linear axes X, Y and Z are required along with a combination of two additional rotary axes A, B or C. It must be possible to control the axes simultaneously.

This is necessary, for example on a head kinematic with two rotary axes C and A, if the tool needs to be aligned normal to the machining surface.

Using three linear axes and two rotary axes, theoretically any point in space can be approached with any tool orientation. This is the basic principle of 5-axis machining.

Generally speaking, 5-axis programs are created with the CAM system and the postprocessor is responsible for defining the type of orientation process used. The most commonly used types of tool orientation are described here in the following section.

Tool orientation independent of machine kinematics
(Workpiece related orientation)

The tool orientation (setting angle) is programmed in a workpiece related coordinate system (WCS) and is therefore independent of the machine kinematics. Programming takes place by means of:

- Cartesian vector A3=.. B3=.. C3=..
- Lead/Tilt angle LEAD=.. TILT=..
- RPY-angles A2=.. B2=.. C2=..
- Euler-angles A2=.. B2=.. C2=..

Tool orientation dependent on machine kinematics
(Machine-related orientation)

The tool orientation is programmed in a machine related coordinate system (MCS) and is therefore dependent of the machine kinematic. Programming takes place by means of:

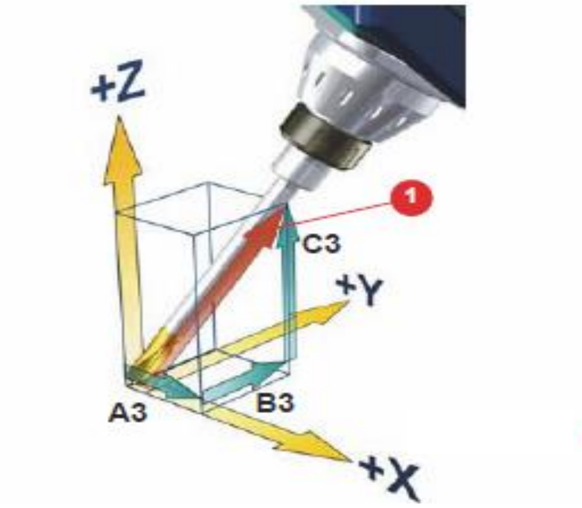
- Direct rotary axes positions A.. B.. C..

Notes

Programming of Cartesian vectors A3=.. B3=.. C3=..

General explanation:

The components of the directional vector 1 that define the tool orientation are programmed with A3, B3, C3. The vector points in direction of the tool holder from the current X, Y, Z origin. The length of the vector is meaningless. Vector components that are not programmed are set equal to zero.



NC command	Directional vector
G1 X.. Y.. Z.. A3=.. B3=.. C3=..	Tool orientation defined by a directional vector through the vector components defined by A3, B3, C3 (recommended).

In this example the tool points to the origin of the coordinate system (X0,Y0,Z0) and the tool vector represents the diagonal of a cube of 35.264° in the resulting XY-plane.

```
N10 TRAORI
N20 ORIWKS
N30 G54 D1
N40 G1 X0 Y0 Z0 A3=1 B3=1 C3=2 F10000
…….
```

Note:

The use of directional vectors is recommended as preferred programming method, in order to create programs **independent** of the machine kinematics. The vector accuracy should be set relatively high. Values of 5 decimal places for linear axes and 6 decimal places for directional vectors produce good results.

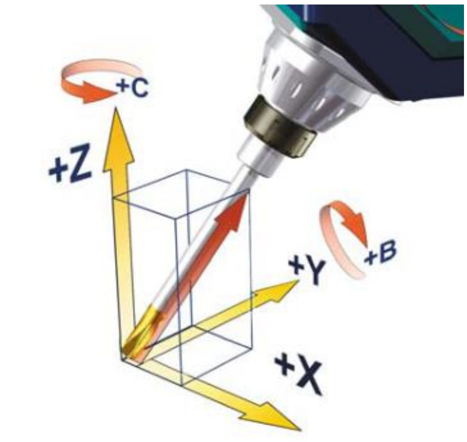
If C3=1 is programmed only, a alignment of the tool axis along the Z-axis takes place. This comes in handy if e.g. the tool needs to be retracted in Z direction with active FRAME rotation or alignment of the tool orientation in the active plane.

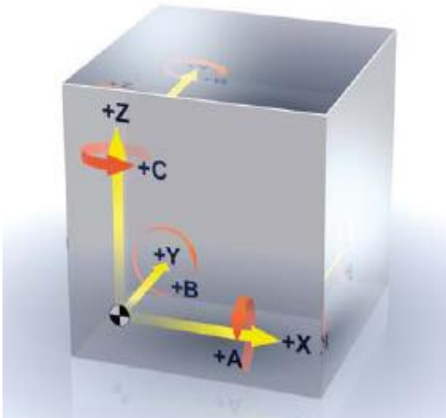
Notes

Programming of direct rotary axes positions (A.. B.. C..)

General explanation:

The same position with tool orientation as for directional vector programming can also be program- med with specification of direct machine rotary axes positions. The tool tip points to the X, Y, Z coordinate of the point to be approached on the workpiece. The tool orientation is described through the machine rotary axes position programmed with A, B, C in reference to the machine coordinate system (MCS). Programming of direct rotary axes makes the program dependent on the machine kinematics.





Definition of rotary axes as of DIN/ISO
(right hand rule)

In this example the tool points to the origin of the coordinate system (X0,Y0,Z0) and the tool orientation represents the diag-

NC command	Description
G1 X… Y… Z… A... B... C…	Direct programming of the rotary axes movements A, B, C. Rotary axes are traversed synchronous to the tool path.

onal of a cube of 35.264°.
N10 TRAORI
N20 G54 D1
N30 G1 X0 Y0 Z0 B35.264 C45 F10000
…….

Note:

If programming rotary axes direct, the generated NC program becomes dependent on the machine kinematics. Flexibility is lost, since the program only runs on the machine kinematic that the program was originally generated for.

As resolution the same number of decimals can be used for the rotary axes positions as for the linear axes. A higher reso- lution of decimal places is not required.

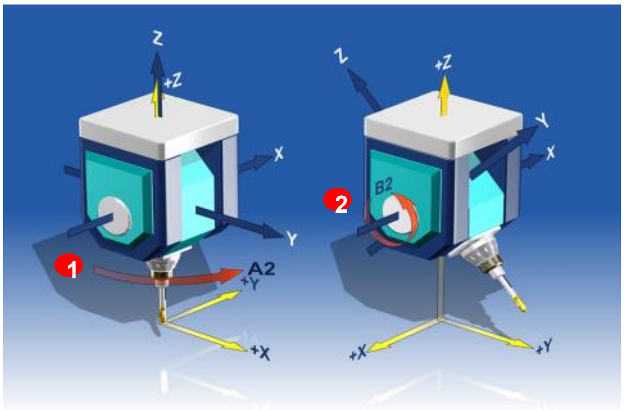
Notes

Programming with Euler-angles (A2=.. B2=.. C2=..)

General explanation:

With active G-code **ORIEULER** the values programmed during orientation programming with A2, B2, C2 are interpreted as Euler-angles

Starting with initial orientation the orientation vector results from turning a vector in the Z direction firstly with **A2** around the Z-axis 1, then with **B2** around the new X axis 2 and lastly with **C2** around the new Z-axis (order **Z, X', Z''**).



Programming example:

N10 TRAORI
N20 G54 D1 G0 X0 Y0 Z0
N30 G1 A2=0 B2=0 F1000
N40 G1 A2=90 B2=45
N50 …..

NC command	**Tool orientation with Euler angles**
ORIEULER	Orientation programming on the basis of Euler angles G-code in relation to MD21102
G1 X Y Z A2= B2= C2=	Programming on basis of Euler angles (in degrees), interpretation is defined by machine data MD21100.

Note:

With setting of ***MD21100 $MC_ORIENTATION_IS_EULER = 1*** (default)
orientation angles for the orientation axes are interpreted as Euler-angles.

In Kanal-Maschinendatum ***MD21102 $MC_ORI_DEF_WITH_G_CODE*** wird dann festgelegt ob die Festlegung durch MD21100 oder durch programmieren des G-Code Befehls erfolgen soll.

0 = Interpretation as in MD21100 $MC_ORIENTATION_IS_EULER
1 = Interpretation with G-Code ORIEULER or ORIRPY

Notes

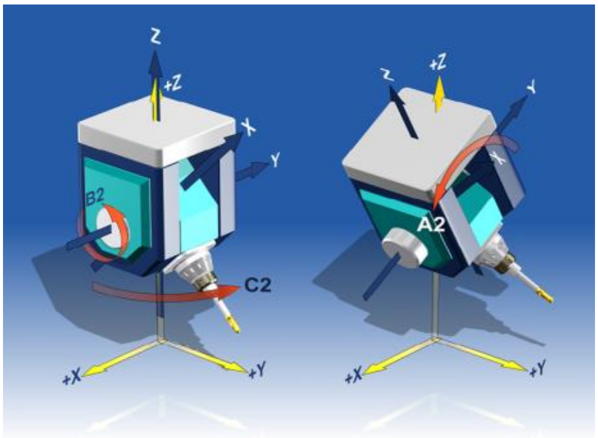
<u>Programming with RPY angles (A2=.. B2=.. C2=..)</u>

General explanation:

With active G-code **ORIRPY** the values programmed with A2, B2, C2 for orientation programming are interpreted as an RPY-angle (in degrees).

Starting with initial orientation the orientation vector results from turning a vector in the Z direction firstly with **C2** around the Z axis **1**, then with **B2** around the new Y axis **2** and lastly with **A2** around the new X axis **3** (order **Z, Y', X''**).

In contrast to Euler-angle programming, all three values here have an effect on the orientation vector.



Programming Example:

```
N10 TRAORI
N20 G54 D1 G0 X0 Y0 Z0
N30 G1 A2=0 B2=0 C2=0 F1000
N40 G1 A2=30 B2=45 C2=90
N50 …..
```

NC command	Description
ORIRPY	Orientation programming on the basis of RPY angles G-code (MD21102).
G1 X.. Y.. Z.. A2= B2= C2=	Programming of the orientation axes according to RPY as rotations in the order **Z, Y', X''**. The programmed order of the addresses A2=, B2=, C2= is not relevant in this case.

<u>Note:</u>

With setting of ***MD21100 $MC_ORIENTATION_IS_EULER = 0*** orientation angles for the orientation axes are interpreted as RPY angles.

With ***MD21102 $MC_ORI_DEF_WITH_G_CODE*** you can set whether the interpretation of the orientation angles are set through MD21100, or with programming the G-Code command.

0 = Interpretation as in MD21100 $MC_ORIENTATION_IS_EULER
1 = Interpretation with G-Code ORIEULER or ORIRPY

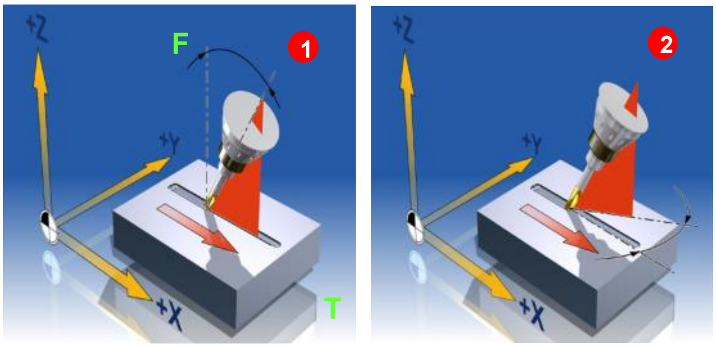
Notes

Programming LEAD & TILT relative to surface normal and path tangent

General explanation:

LEAD and TILT are programmed if the tool is to adopt a fixed setting angle in relation to the machining surface, e.g. so that machining is not performed at the cutter center at cutting surface speed of 0.

A surface is spanned in the plane of the surface normal vector FN and path tangent TB, which defines the meaning of **LEAD** and **TILT** at the end point. This means that the path reference applies only for the definition of the end orientation vector. The tool orientation is determined by:



- the path tangent TB
- the surface normal vector FN at the start of the block **A4, B4, C4** and at the end **A5, B5, C5**
- the lead angle **LEAD** 1
- the tilt angle **TILT** 2 at end of block

```
N100 TRAORI
N110 ORIWKS
N120 ORIPATH
N130 CUT3DF
N140 G54 D1
N150 G0 X50 Y40 LEAD=0 TILT=0 A4=0.000 B4=0.000 C4=1.000
N160 G1 Z-2 F500
N170 LEAD=35.264 TILT=45
N170 G1 G41 X140.000 A5=0.000 B5=0.000 C5=1.000
……
```

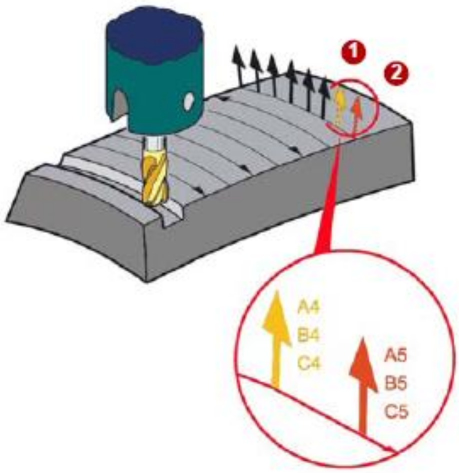
NC command	Description
LEAD	LEAD describes the angle relative to the surface normal vector in the plane spanned by the path tangent TB and surface normal vector FN. The maximum permissible LEAD angle can be set in cannel machine data: ***MD21090 $MC_MAX_LEAD_ANGLE*** (range 0-80 degree).
TILT	TILT then defines the rotation around the surface normal vector FN. The maximum permissible TILT angle can be set in the cannel machine data: ***MD21092 $MC_MAX_TILT_ANGLE*** (range –180 to +180 degree).

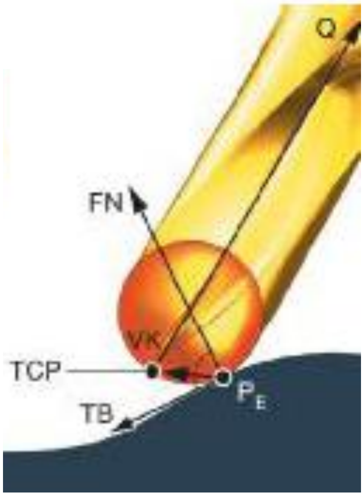
Notes

<u>Programming surface normal’s (A4= B4= C4= and A5= B5= C5=)</u>

General explanation:

This information is used for face milling of curved surfaces of any kind. For this type of 3D milling, you require line-by-line definition of 3D paths on the workpiece surface. The tool shape and dimensions are taken into account in the calculations with active 3D cutter compensation, which are normally performed in the CAM.





The surface normal vector (**FN**) is perpendicular to the machining surface (**TB**) and defines the path curvature. The specification of the surface normal at each point is necessary for the definition of LEAD and TILT angle and use of 3D cutter compensation with **CUT3DF**.

This has the advantage that when using a 3D tool radius compensation with **CUT3DF**, this can also be compensated by the control for tools that differ to the normal tools. A possible application would be an allowance programming for the face milling of free-form surfaces.

With activation of the **ORIPATH**, great circle interpolation is performed between the start to the end orientation if in addition the surface normal vectors are programmed. The following combinations of surface normal vector programming are possible:

- If only the start surface normal vector is programmed in a block (A4=, B4=, C4=) **1**, then the programmed surface normal vector will remain constant throughout the entire block.
- If only the end surface normal vector is programmed (A5=, B5=, C5=) **2**, then great circle interpolation is performed between the end surface normal of the previous block and the programmed end surface normal.
- If both start (A4=, B4=, C4=) **1** and end surface normal vectors (A5=, B5=, C5=) **2** are programmed, interpolation ac-

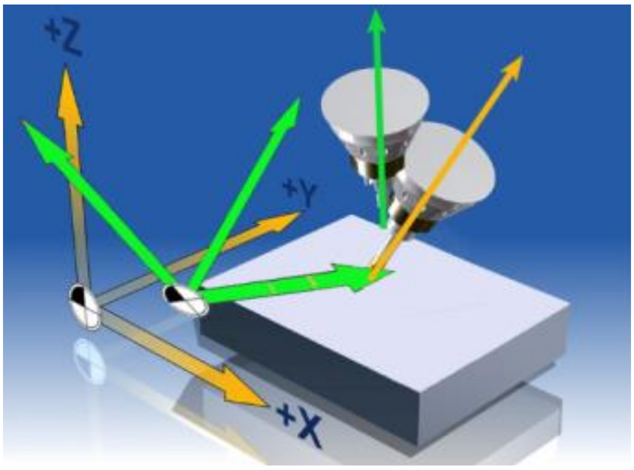
NC command	Description
A4= B4= C4=	Specification of the surface normal vector at start of block.
A5= B5= C5=	Specification of the surface normal vector at end of block

Notes

Orientation reference ORIMKS/ORIWKS

General explanation:

The reference of the tool orientation is set using the G-code commands ORIWKS/ORIMKS.



ORIWKS ***Tool orientation in reference to the workpiece coordinate system***

Is preferable used for orientation programming in the workpiece coordinate system with

- cartesian orientation
- vectors or orientation angles

Hereby the reference system for the orientation vector is the workpiece coordinate system (WCS), which can be rotated through a frame rotation.

Here the relative Position and orientation of the tool tip in relation to the workpiece are being programmed. The movement executed by the tool does therefore **not depend** on the machine kinematics. The actual movements of the machine then depends on the kinematic of the machine.

In the case of a 5-axis program, if it is not immediately obvious on which type of machine kinematics it is to run, always choose ORIWKS.

ORIMKS ***Tool orientation in reference to the machine coordinate system***

Is only used for orientation programming of direct rotary axes positions A, B, C in reference to the machine coordinate system (MCS).

The reference system for the orientation vector is the machine coordinate system (MCS). Rotation of the workpiece frame is not being considered. Rotary axes offsets in the work offset are considered if the following machine data is set

MD24590 $MC_TRAFO5_ROT_OFFSET_FROM_FR_1.

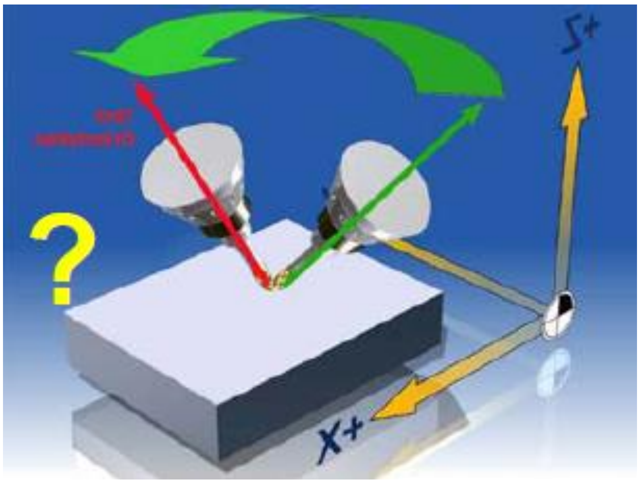
Direct rotary axes movements (A,B,C) are here being programmed, for example to avoid collision with fixtures or for simulation reasons on the CAM system (recommended for simulation on the CAM system without VNCK).

The directly programmed rotary axes movements determine the actual machine movements and are therefore **dependent** on the machine kinematic.

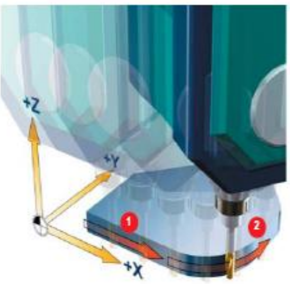
Notes

Interpolation of the orientation axes

General explanation:



A 5-axis machine can position the tool in any orientation to the workpiece, subject of course to the machine kinematics. In order to move from one orientation to another, intermediate positions must be interpolated. The path from the start to the end orientation is described in this way.



In the 3-axis range interpolation takes place between 2 points, with any number of paths between 2 orientations. The known interpolation types are:

- Straight line (G1) **1**
- Circle (G2, G3) **2**
- Polynomial interpolation (no picture)

  A-spline (Akima spline)

  B spline (Bezier Spline)

  C-spline (Cubic spline)

Various types of interpolation are used for 5-axis applications. In this example, which involves milling a pocket wall with an inclination of 45°, the tool moves from position **1** to position **2**. The A- and C-axes rotate during the movement simultaneously, so that the tool can be oriented along the edges of the pocket. This kind of interpolation is known as

- Vector interpolation or great circle interpolation **3**

The most common types of 5-axis orientation interpolations are explored on the following pages.

Notes

Rotary axes-/linear interpolation ORIAXES

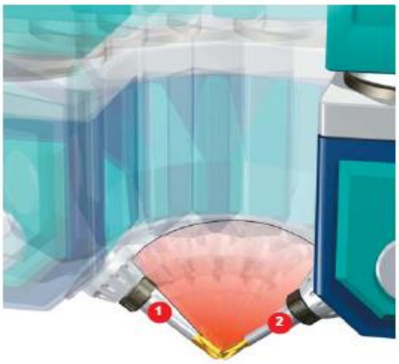
General explanation:

In linear interpolation from a start **1** to an end orientation **2** the necessary rotary axis movements are broken down into equi-distant sections.

This means that the orientation vector does not describe a defined surface. This interpolation is therefore not always ideal for circumferential milling.

CAM-systems attempt to compensate for this effect by using sufficiently small interpolation steps.
For optimum results, another type of interpolation (such as vector interpolation) should be used for these kinds of applica-tions.

The orientation interpolation with ORIAXES is recommended for programming on a CAM-system using 5-axis face milling strategies to machine free form surfaces in the mould and die sector.



Note:

When programming on a CAM system, the effect of the G-code command ORIAXES must be set accordingly in your CAM-system. By all means, with ORIAXES the CAM-system must calculate sufficiently small intermediate i nterpolation steps, between start and end orientation in order to interpolate an exact defined surface (tool path), otherwise contour gauging through linear interpolation of the rotary axes can occur.

For this exact reason the linear point tolerance on the CAM system must be set sufficiently small (recommended value LinTol <= 0.3mm). This off course increases also the number of data blocks in the NC-program.

Notes

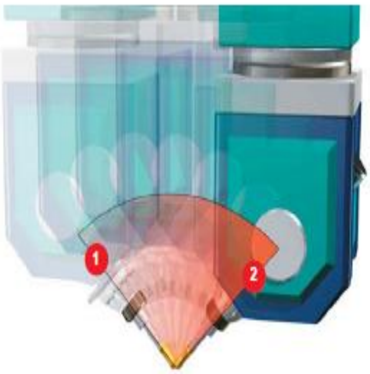
Vector-/Great circle interpolation ORIVECT

General explanation:

In this interpolation process the path from the start **1** to an end orientation **2** is interpolated in such a way that the orientation vector runs in the plane that is spanned by the starting vector and the final vector.

Each rotary axis approaches equidistant angles (see picture on next page). This type of orientation interpolation can be used for example to precisely machine inclined, plane walls in one block
(see example on next page).

The orientation interpolation with ORIVECT is recommended for programming on the CAM system
or manual programming on the control for side milling operations (5-axis circumferential milling) of ruled surfaces e.g. structural components.



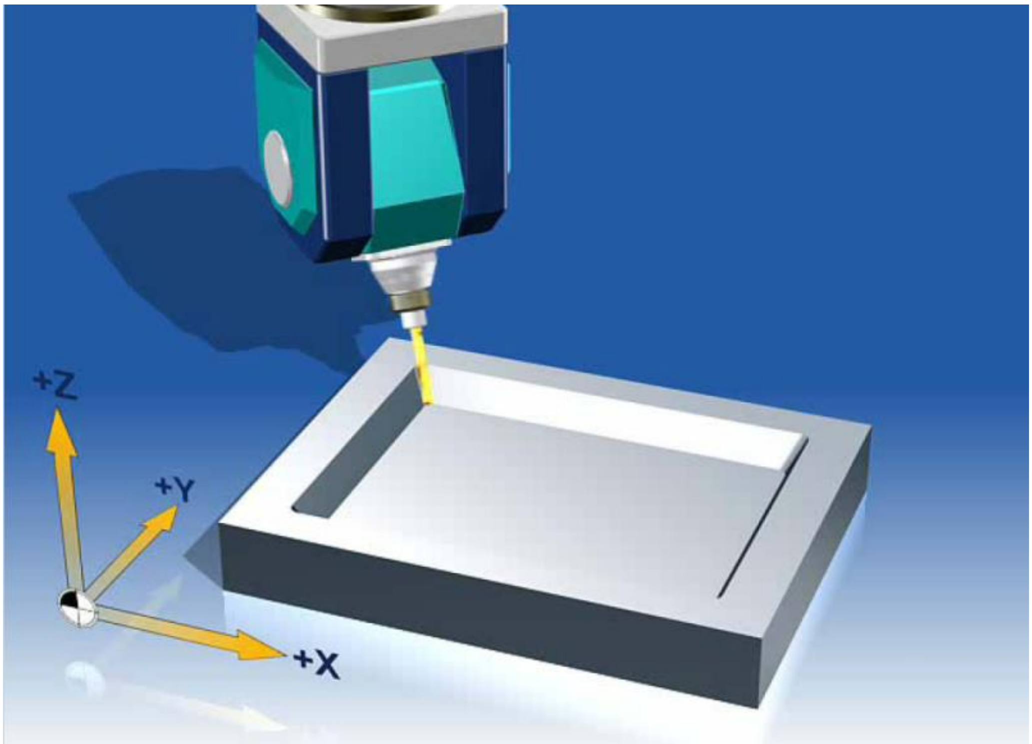
NC command	*Vector interpolation/Great circle interpolation*
ORIVECT	Interpolation of the orientation vector in one plane (great circle interpolation)

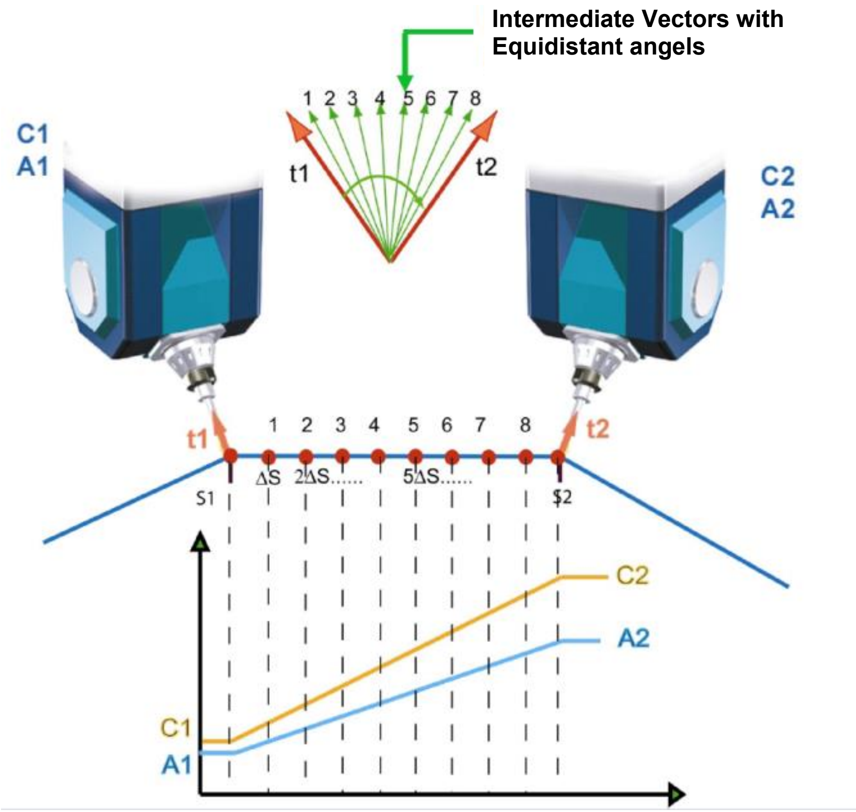
Note:

When programming on a CAM-system, the effect of the G-code command ORIVECT must be set accordingly in your CAM-system. By all means, with ORIVECT there is **no** need to calculate intermediate orientations, between start orientation **1** and end orientation **2** to interpolate an exact defined surface (tool path), the NC takes care of the calculation.

With **ORIVECT** the orientation axes are always traversed on the shortest possible path between two orientations.

Notes


+Z
+Y
+X

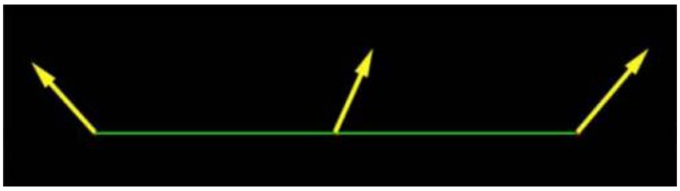

Intermediate Vectors with Equidistant angels
1 2 3 4 5 6 7 8
t1
t2
C1
A1
C2
A2
1 2 3 4 5 6 7 8
ΔS 2ΔS......
5ΔS......
S1
S2
C2
A2
C1
A1

Notes

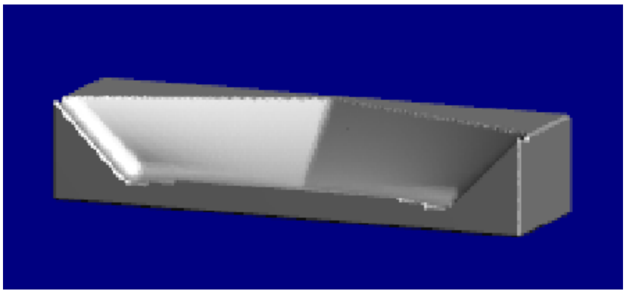
Comparison between interpolation with ORIAXES and ORIVECT

Programming example:

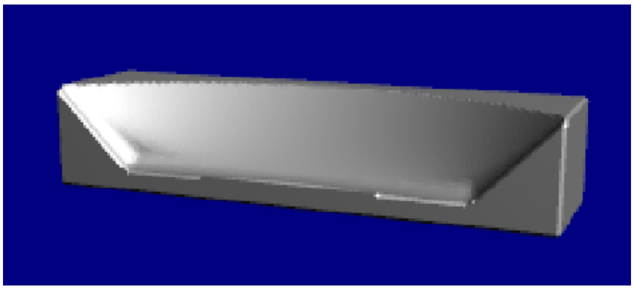
Inclined wall at 45 degrees with two positions/orientations at each corner and one intermediate position.
Tool orientation programmed with direct rotary axes positions



ORIVECT



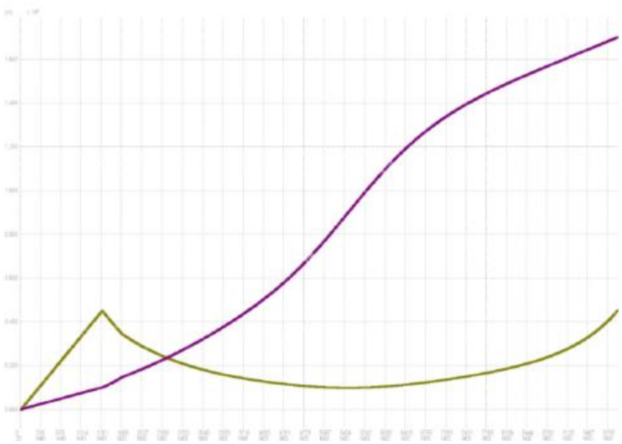
ORIAXES



N20 TRAORI
N30 ORIVECT
N40 G1 X0 Y0 Z0 C0 A45
N50 Y10 C90
N60 Y20 C180

N20 TRAORI
N30 ORIAXES
N40 G1 X0 Y0 Z0 C0 A45
N50 Y10 C90
N60 Y20 C180

Verfahrbewegungen der Rundachsen

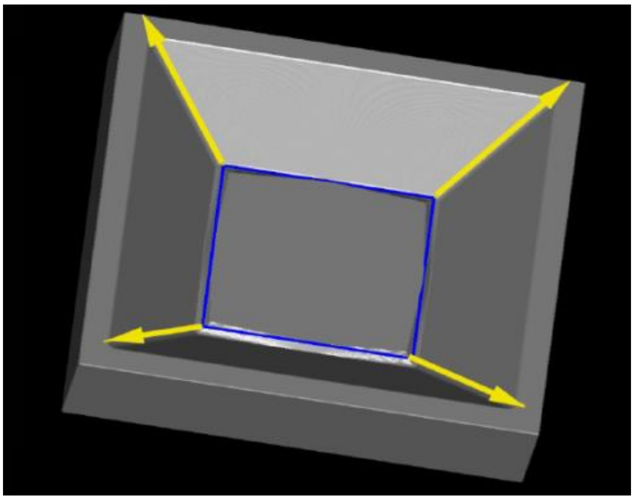


Notes

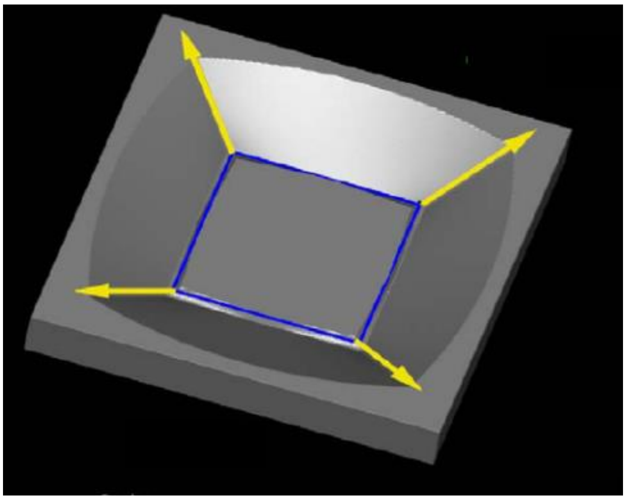
Programming example 1:

Rectangular pocket with inclined walls 45 degree. 4 positions/orientations at each corner of the rectangle. Tool orientations programmed with Cartesian vectors.

ORIVECT



ORIAXES



NC/WKS/M103_TRAORI/EX1_ORIVECT_1

…..

N112 TRAORI
N113 ORIWKS
N114 **ORIVECT**
N115 X0 Y0 A3=-1 B3=-1 C3=1
N116 Y10 A3=-1 B3=1 C3=1
N117 X10 A3=1 B3=1 C3=1
N118 Y0 A3=1 B3=-1 C3=1
N119 X0 Y0 A3=-1 B3=-1 C3=1
N159 TOROT
N160 G1 G91 Z100 F1000
N161 TOROTOF
N162 TRAFOOF
N163 M30

NC/WKS/M103_TRAORI/EX1_ORIAXES_1

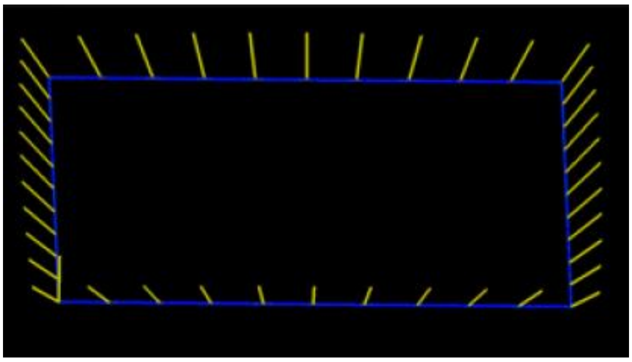
…..

N112 TRAORI
N113 ORIWKS
N114 **ORIAXES**
N115 X0 Y0 A3=-1 B3=-1 C3=1
N116 Y10 A3=-1 B3=1 C3=1
N117 X10 A3=1 B3=1 C3=1
N118 Y0 A3=1 B3=-1 C3=1
N119 X0 Y0 A3=-1 B3=-1 C3=1
N159 TOROT
N160 G1 G91 Z100 F1000
N161 TOROTOF
N162 TRAFOOF
N163 M30

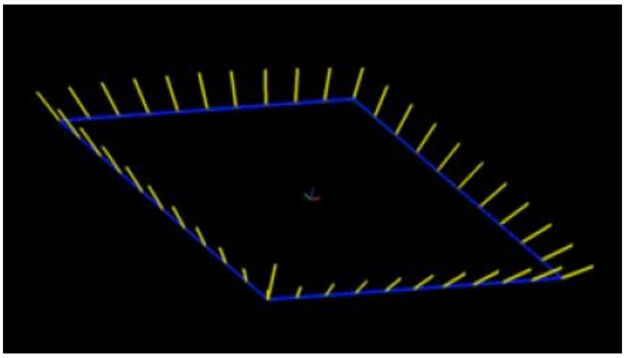
Notes

Programming example 2:

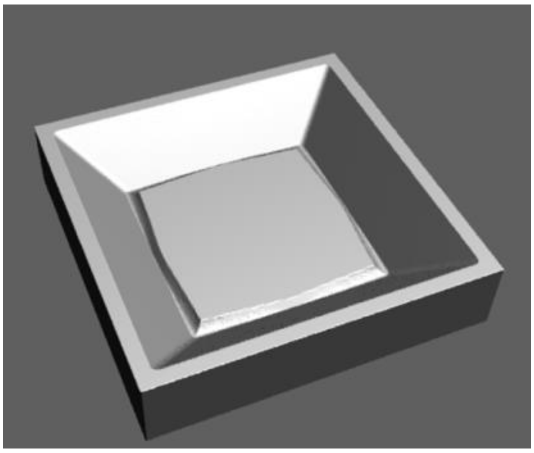
Rectangular pocket with inclined walls 45 degree. 4 positions/orientations at each corner and 9 intermediate positions/ orientations at each side of the rectangle.

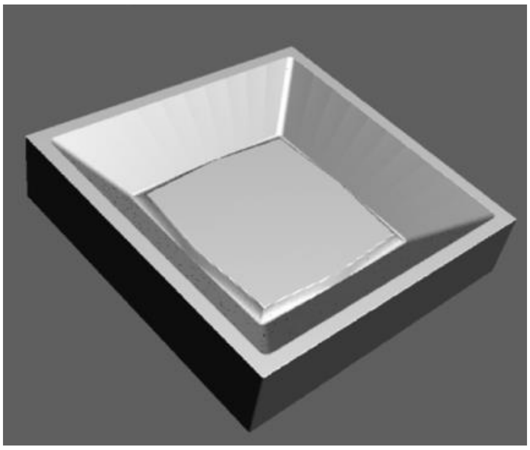


ORIVECT



ORIAXES





NC/WKS/M103_TRAORI/EX1_ORIVECT_2

…..
N112 TRAORI
N113 ORIWKS
N114 **ORIVECT**
N116 G0 X-10 Y-10 Z0
N117 G1 Z-10
N118 Y-10 A3=-1 B3=-1 C3=1
N119 Y-8 A3=-1 B3=-0.726542 C3=1
N120 Y-6 A3=-1 B3=-0.509524 C3=1

Continue on next page

NC/WKS/M103_TRAORI/EX1_ORIAXES_2

….
N112 TRAORI
N113 ORIWKS
N114 **ORIAXES**
N116 G0 X-10 Y-10 Z0
N117 G1 Z-10
N118 Y-10 A3=-1 B3=-1 C3=1
N119 Y-8 A3=-1 B3=-0.726542 C3=1
N120 Y-6 A3=-1 B3=-0.509524 C3=1

Continue on next page

Notes

```
N125 Y4 A3=-1 B3=0.324919 C3=1
N126 Y6 A3=-1 B3=0.509524 C3=
N127 Y8 A3=-1 B3=0.726542 C3=1
N128 Y10 A3=-1 B3=1 C3=1
N129 X-8 A3=-0.726542 B3=1 C3=1
N130 X-6 A3=-0.509524 B3=1 C3=1
N131 X-4 A3=-0.324919 B3=1 C3=1
N132 X-2 A3=-0.158384 B3=1 C3=1
N133 X0 A3=0 B3=1 C3=1
N134 X2 A3=0.1583844 B3=1 C3=1
N135 X4 A3=0.324919 B3=1 C3=1
N136 X6 A3=0.509524 B3=1 C3=1
N137 X8 A3=0.726542 B3=1 C3=1
N138 X10 A3=1 B3=1 C3=1
N139 Y8 A3=1 B3=0.726542 C3=1
N140 Y6 A3=1 B3=0.509524 C3=1
N141 Y4 A3=1 B3=0.324919 C3=1
N142 Y2 A3=1 B3=0.1583844 C3=1
N143 Y0 A3=1 B3=0 C3=1
N144 Y-2 A3=1 B3=-0.1583844 C3=1
N145 Y-4 A3=1 B3=-0.324919 C3=1
N146 Y-6 A3=1 B3=-0.509524 C3=1
N147 Y-8 A3=1 B3=-0.726542 C3=1
N148 Y-10 A3=1 B3=-1 C3=1
N149 X8 A3=0.726542 B3=-1 C3=1; P2
N150 X6 A3=0.509524 B3=-1 C3=1; P3
```


N125 Y4 A3=-1 B3=0.324919 C3=1
N126 Y6 A3=-1 B3=0.509524 C3=
N127 Y8 A3=-1 B3=0.726542 C3=1
N128 Y10 A3=-1 B3=1 C3=1
N129 X-8 A3=-0.726542 B3=1 C3=1
N130 X-6 A3=-0.509524 B3=1 C3=1
N131 X-4 A3=-0.324919 B3=1 C3=1
N132 X-2 A3=-0.158384 B3=1 C3=1
N133 X0 A3=0 B3=1 C3=1
N134 X2 A3=0.1583844 B3=1 C3=1
N135 X4 A3=0.324919 B3=1 C3=1
N136 X6 A3=0.509524 B3=1 C3=1
N137 X8 A3=0.726542 B3=1 C3=1
N138 X10 A3=1 B3=1 C3=1
N139 Y8 A3=1 B3=0.726542 C3=1
N140 Y6 A3=1 B3=0.509524 C3=1
N141 Y4 A3=1 B3=0.324919 C3=1
N142 Y2 A3=1 B3=0.1583844 C3=1
N143 Y0 A3=1 B3=0 C3=1
N144 Y-2 A3=1 B3=-0.1583844 C3=1
N145 Y-4 A3=1 B3=-0.324919 C3=1
N146 Y-6 A3=1 B3=-0.509524 C3=1
N147 Y-8 A3=1 B3=-0.726542 C3=1
N148 Y-10 A3=1 B3=-1 C3=1
N149 X8 A3=0.726542 B3=-1 C3=1; P2
N150 X6 A3=0.509524 B3=-1 C3=1; P3


```
N151 X4 A3=0.324919 B3=-1 C3=1; P4
N152 X2 A3=0.158384 B3=-1 C3=1; P5
N153 X0 A3=0 B3=-1 C3=1; P6
N154 X-2 A3=-0.1583844 B3=-1 C3=1; P7
N155 X-4 A3=-0.324919 B3=-1 C3=1; P8
N156 X-6 A3=-0.509524 B3=-1 C3=1; P9
N157 X-8 A3=-0.726542 B3=-1 C3=1; P10
N158 X-10 A3=-1 B3=-1 C3=1
N159 TOROT
N160 G1 G91 Z100 F1000
N161 TOROTOF
N162 TRAFOOF
N163 M30
```


N151 X4 A3=0.324919 B3=-1 C3=1; P4
N152 X2 A3=0.158384 B3=-1 C3=1; P5
N153 X0 A3=0 B3=-1 C3=1; P6
N154 X-2 A3=-0.1583844 B3=-1 C3=1; P7
N155 X-4 A3=-0.324919 B3=-1 C3=1; P8
N156 X-6 A3=-0.509524 B3=-1 C3=1; P9
N157 X-8 A3=-0.726542 B3=-1 C3=1; P10
N158 X-10 A3=-1 B3=-1 C3=1
N159 TOROT
N160 G1 G91 Z100 F1000
N161 TOROTOF
N162 TRAFOOF
N163 M30

Note:

Additional specification of intermediate orientations have no effect on the milling pattern when programming with **ORIVECT**. The same result is achieved when only four corner points are programmed.

Note:

For a good result of an exact straight wall with/without gauge marks, the linear point distance for interpolation with **ORIAXES** must be set to at least 0.3 mm. This is usually done in the CAM system with the setting of the linear tolerance (refer to CAM manufacturer).

Notes

Note:

The interpolation type for the orientation is specified in ***MD21104 $MC_ORI_IPO_WITH_G_CODE***
By means of this data it is determined whether the G-codes of group 51 can be used or not.

= 0: Interpolation type provided by G codes ORIWKS, ORIMKS. In this case the G-codes of group 51 are not allowed to be programmed. The following applies:
ORIWKS = Interpolation with ORIVECT
ORIMKS = Interpolation with ORIAXES.

= 1: Interpolation type provided by G codes of the G-code group 51 (e.g. ORIAXES, ORIVECT, ORIPATH, ...). All G codes of group 51 can be freely programmed (**recommended setting**).

Possible programming combinations of the G-code group 51

Interpolation type and reference	MD21104 $MC_ORI_IPO_WITH_G_CODE	
	= “0”	= “1”
Rotary axes interpolation in MCS	ORIMKS G1 X.. Y.. Z.. A.. C..	ORIMKS ORIAXES G1 X.. Y.. Z.. A.. C..
Rotary axes interpolation in WCS	Not possible	ORIWKS ORIAXES G1 X.. Y.. Z.. A.. C..
Great circle interpolation in MCS with vector orientation programming	ORIWKS G500 G1 X.. Y.. Z.. A3=.. B3=.. C3=.. (special case only because of G500 otherwise not possible)	ORIMKS ORIVECT G1 X.. Y.. Z.. A3=.. B3=.. C3=..
Great circle interpolation in WCS with vector orientation programming	ORIWKS G1 X.. Y.. Z.. A3=.. B3=.. C3=..	ORIWKS ORIVECT G1 X.. Y.. Z.. A3=.. B3=.. C3=..
Great circle interpolation in MCS with rotary axes programming	Not possible	ORIMKS ORIAXES G1 X.. Y.. Z.. A.. C..
Great circle interpolation in WCS with rotary axes programming	Not possible	ORIWKS ORIVECT G1 X.. Y.. Z.. A.. C..

Notes

Interpolation relative to path tangent ORIPATH

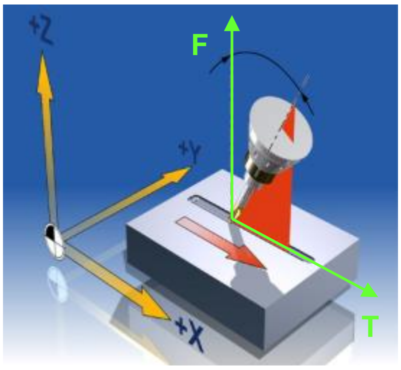
General explanation:

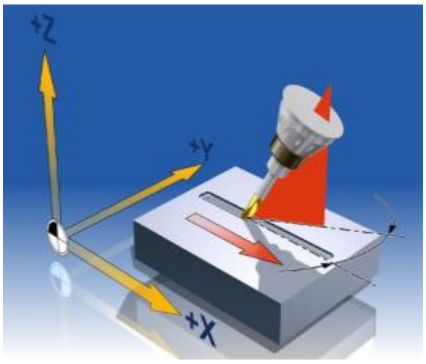
The orientation interpolation with **ORIPATH** is suitable for 5-axis face machining of free form surfaces in the mould and die sector. Prerequisite is a CAM-system capable of supporting the output of surface normal vectors.

The advantage lies in a adoptable setting angle of the tool programmed with **LEAD** and **TILT** relative to the path tangent and surface normal, so that machining is not performed at the cutter center at cutting surface speed of 0.

This leads to a improved tool interpolation path and ultimately to improvements of workpiece surfaces during machining.

If both start (**A4=,B4=,C4=**) and end (**A5=,B5=,C5=**) surface normal vectors are programmed, then a surface is spanned between these vectors and interpolation according to the great circle principle is performed between the two directions.





NC command	Interpolation relative to path tangent
ORIPATH	Tool orientation relative to the path tangent, by the meaning of **LEAD** and **TILT.** Through additional definition of surface normal vectors at start (A4=, B4=, C4=) and end orientation (A5=, B5=, C5=) great circle interpolation is performed between the two directions.

Note:

When programming on a CAM system, the effect of the G-code command **ORIPATH** must be set accordingly in the following machine data ***MD21094 $MC_ORIPATH_MODE = 0*** (default).
The setting in the machine data effect the interpolation behaviour when programming surface normal at the beginning or the end of the NC-block.

Notes

Interpolation behaviour in the vicinity of a pole

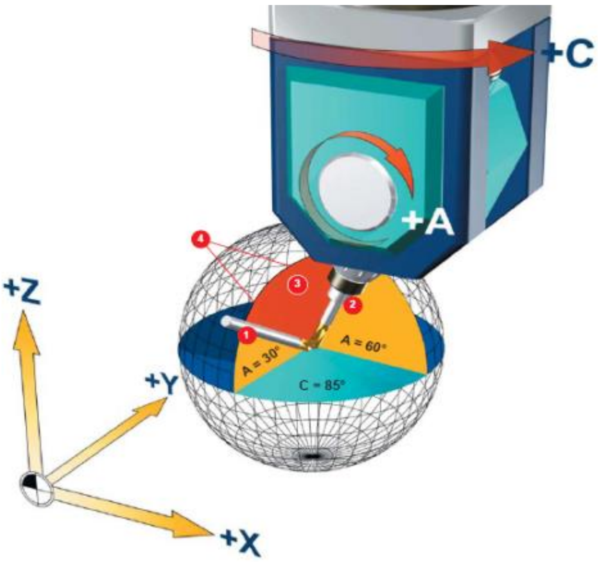
Great circle interpolation with ORIVECT without pole position

In this example of great circle interpolation the tool orientation changes from the start orientation of A=60°, C=0° to the end orientation of A=30°, C=85°. No value is zero, i.e. the orientation is inclined.

Reorientation in great circle interpolation takes place in a plane 3.
The line on the sphere is referred to as a great circle 4.

In the illustration the C axis rotates through 85°. The A-axis swivels from 60° to 30°.
Speed control for the reorientation is continuous. Every thing runs normal since the second rotary axes A does not reach the pole position “A=0” during great circle interpolation.

1 = start orientation, 2 = end orientation



Example Program RPY:

TRAORI
ORIVECT (4)
ORIWKS
G1 X0 Y0 Z0 A2=60 B2=0 C2=0 (1)
A2=30 B2=0 C2=85 (2)
.....

Notes

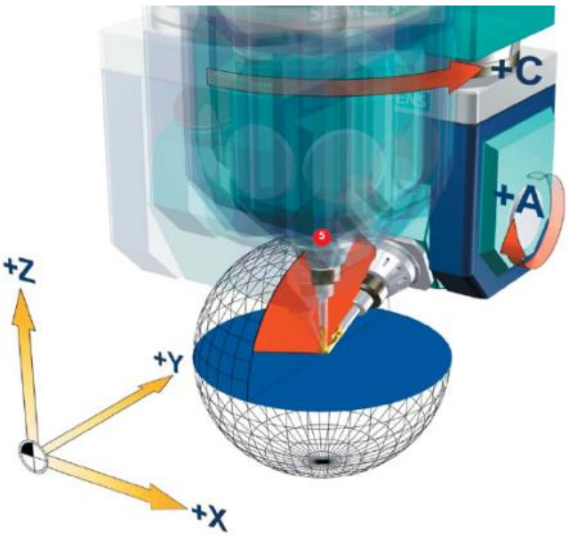
Behaviour ORIVECT at pole position

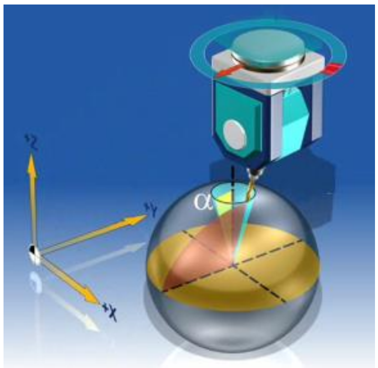
In this example the programmed start position **1** is A=60° and the end position **2** is A=-60°. The A-axis should not be allowed to swivel pass 0° due to mechanical limitation of the A-axis.

What happens next? The C-axis starts at 0°, the A-axis at 60°, position **1**. During the orientation interpolation C-axis remains at 0°, while the A-axis approaches the position **5** (pole position). Here the position of the C-axis is undefined, but just one interpolation phase later the C-axis must rotate abruptly to 180°. The A-axis moves away from point **5** back to start position **1** to A=60° and C remains at 180° (resulting end position **2**).

To maintain the orientation speed, i.e. in position **5**, the C-axis would have to accelerate infinitely, which of course is not possible. In this case we speak of a **pole position** (singular position). For conventional 5-axis machines the pole zero position is defined in such a way, that when the second rotary axis rotates (here C-axis), the tool orientation remains unchanged.

Example AC-head-kinematics: Here A=0 corresponds to the pole orientation **5**. To avoid such an extreme speed control, in the vicinity of the pole the control switches to **linear-/rotary axes interpolation** (ORIAXES) and changes orientation within the pole cone (α). The angle of the pole cone can be set in ***MD 24540 $MC_TRAFO5_POLE_LIMIT_1*** (default setting 2 degree).





Example NC program RPY:

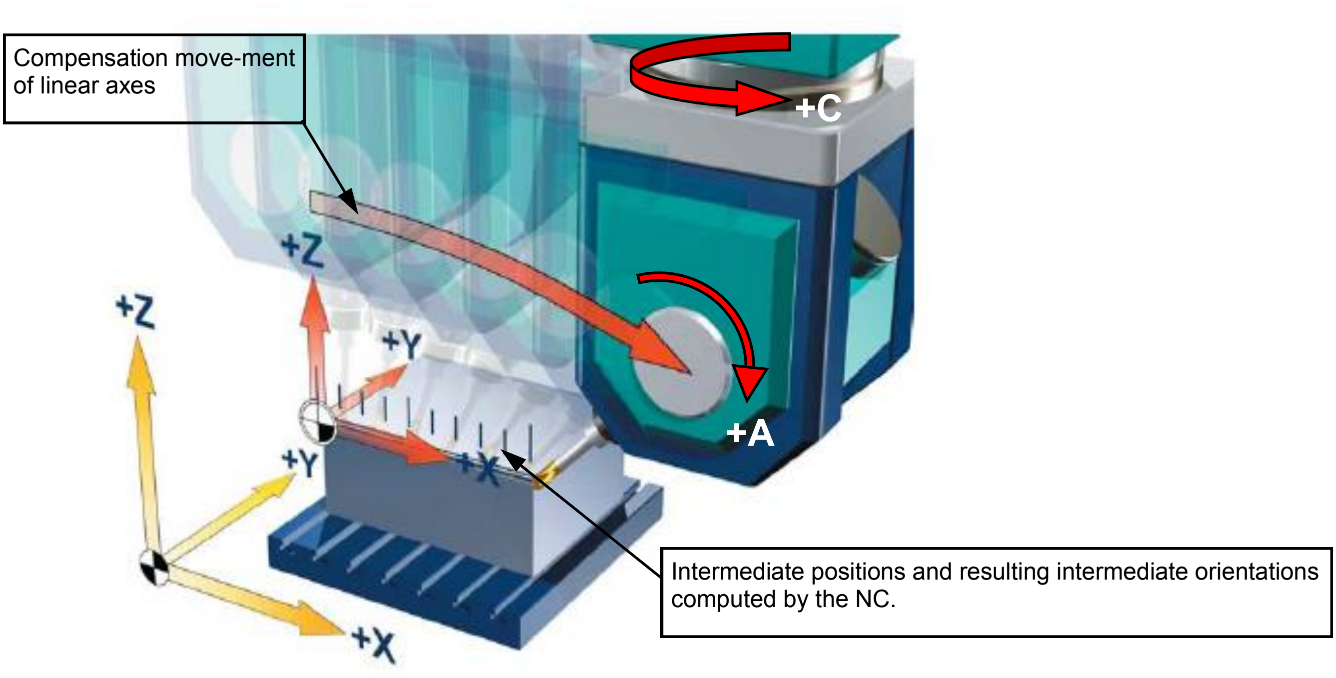
```
TRAORI
ORIVECT (4)
ORIWKS
G1 X0 Y0 Z0 A2=-60 B2=0 C2=0 (1)
G1 A2=60 B2=0 C2=0 (2)
.....
```

Note:

In channel machine data ***MD21108 POLE_ORI_MODE*** it is possible to define the behaviour in pole environment. For detailed information see on DoconCD.

Notes

Example NC-programm with Cartesian vectors A3=.. B3=.. C3=..




```
N10 WORKPIECE(,,,"BOX",112,0,-30,-80,0,0,100,60)
N11 T="BALL_MILL_D8"
N12 M6
N13 S10000 M3 F1000        ;Technology data (speed, feed etc.)
N14 TRAORI                 ;Call 5-axis transformation
N15 ORIWKS                 ;Orientation reference WCS
N16 ORIVECT                ;Great circle interpolation
N17 G54 D1                 ;Zero offset and cutting edge #
N18 G0 X0 Y0 C3=1          ;Approach starting point in X,Y tool orientation parallel to Z axis
N19 Z5                     ;Rapid to safety distance
N20 G1 Z0                  ;Infeed in Z
N21 X100 Y0 A3=1 C3=1      ;Linear move with orientation change to 45° in ZX-plane
N22 G0 Z100                ;Retract in Z
N23 C3=1                   ;Align tool parallel to Z axis
N24 TRAFOOF                ;Cancel 5-axis transformation
N25 M30                    ;End of program
```

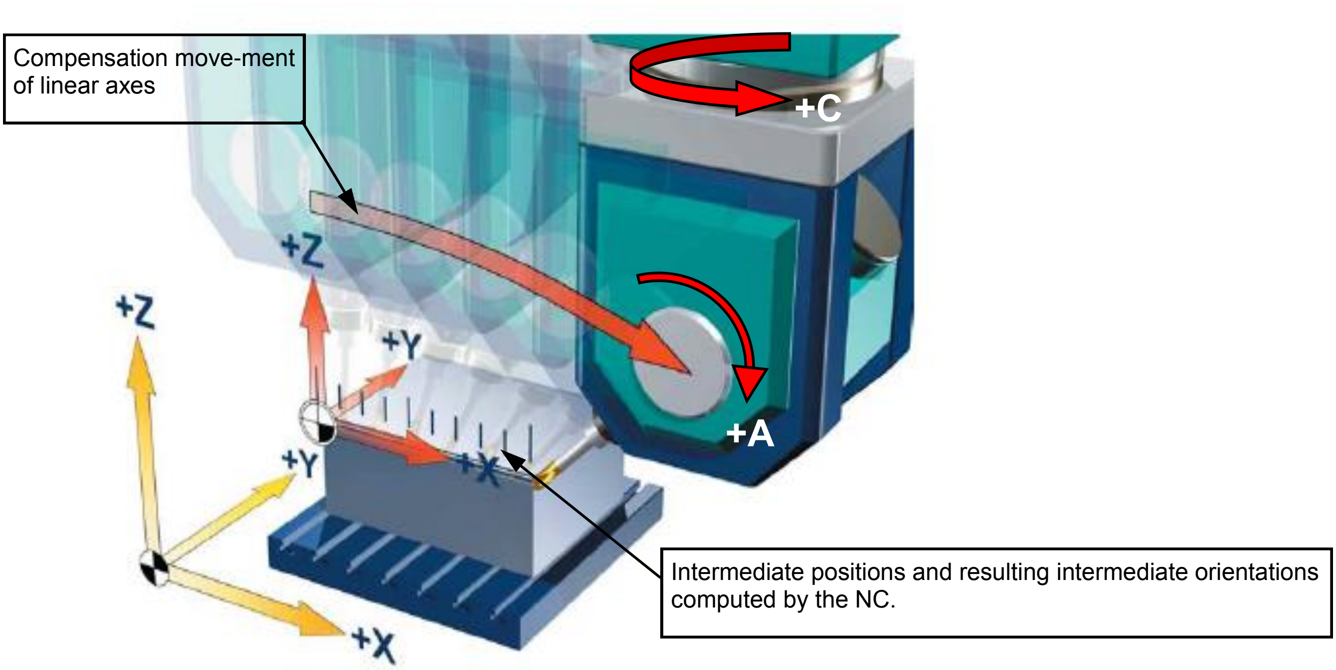
In this example a straight line is milled from X0 to X100. The tool orientation changes simultaneous from 90° to 45° in the ZX plane.

<u>Note:</u>

The tool path in the example is not compensated for the orientation change of the tool cutting edge. The orientation change takes place about the tool tip point (TTP), this results into contour gauging.
In general the CAM-system must calculate a 3 dimensional cutter radius compensation to modify the tool path for a change of the tool cutting edge orientation in space.

Notes

Example NC-programm with RPY orientation angles A2=.. B2=.. C2=..



NOTE ! For interpretation of orientation angles as of RPY MD21100 must be set to 0

```
N10 T="BALL_MILL_D8"
N11 M6
N12 S10000 M3 F1000          ;Technology data (speed, feed etc.)
N13 TRAORI                   ;Call 5-axis transformation
N14 ORIWKS                   ;Orientation reference WCS
N15 ORIVECT                  ;Great circle interpolation
N16 G54 D1                   ;Zero offset and cutting edge #
N17 G0 X0 Y0 A2=0 B2=0 C2=0  ;Approach starting point in X,Y tool orientation parallel to Z axis
N18 Z5                       ;Rapid to safety distance
N19 G1 Z0                    ;Approach starting point in Z
N20 X100 Y0 B2=45            ;Linear move with orientation change to 45° in ZX-plane
N21 G0 Z100                  ;Retract in Z
N22 A2=0 B2=0 C2=0           ;Tool orientation parallel to Z axis
N23 TRAFOOF                  ;Cancel 5-axis transformation
N24 M30                      ;End of program
```

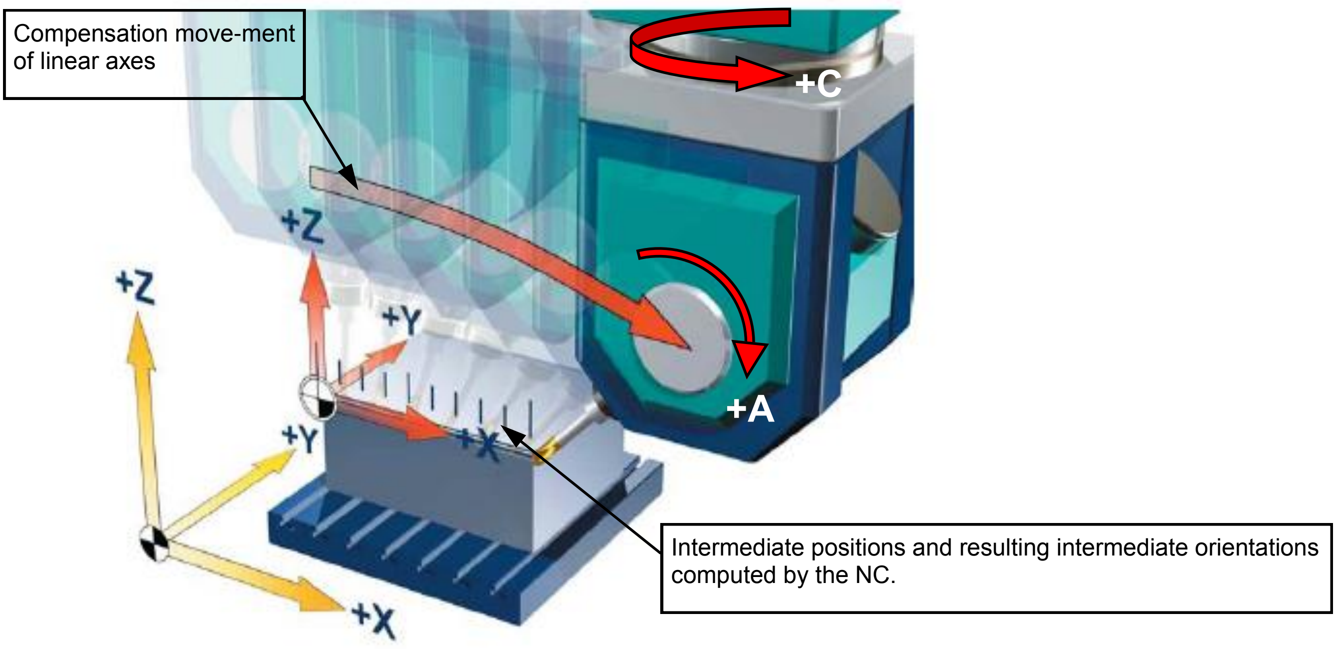
In this example a straight line is milled from X0 to X100. The tool orientation changes simultaneous from 90° to 45° in the ZX plane.

Note:

The tool path in the example is not compensated for the orientation change of the tool cutting edge. The orientation change takes place about the tool tip point (TTP), this results into contour gauging.
In general the CAM-system must calculate a 3 dimensional cutter radius compensation to modify the tool path for a change of the tool cutting edge orientation in space.

Notes

Example NC-programm with EULER orientation angles A2=.. B2=.. C2=..



NOTE ! For interpretation of orientation angles as of EULER MD21100 must be set to 1

```
N10 T="BALL_MILL_D8"
N11 M6
N12 S10000 M3 F1000          ;Technology data, tool, speed, etc
N13 TRAORI                   ;Call 5-axis transformation
N14 ORIWKS                   ;Orientation reference WCS
N15 ORIVECT                  ;Great circle interpolation
N16 G54 D1                   ;Zero offset and cutting edge #
N17 G0 X0 Y0 Z5 A2=0 B2=0    ;Approach starting point in X,Y tool orientation parallel to Z axis
N18 Z5                       ;Rapid to safety distance
N19 G1 Z0                    ;Approach starting point in Z
N20 X100 Y0 A2=90 B2=45      ;Linear move with orientation change to 45° in ZX-plane
N21 G0 Z100                  ;Retract in Z
N22 A2=0 B2=0                ;Tool orientation parallel to Z axis
N23 TRAFOOF                  ;Cancel 5-axis transformation
N24 M30
```

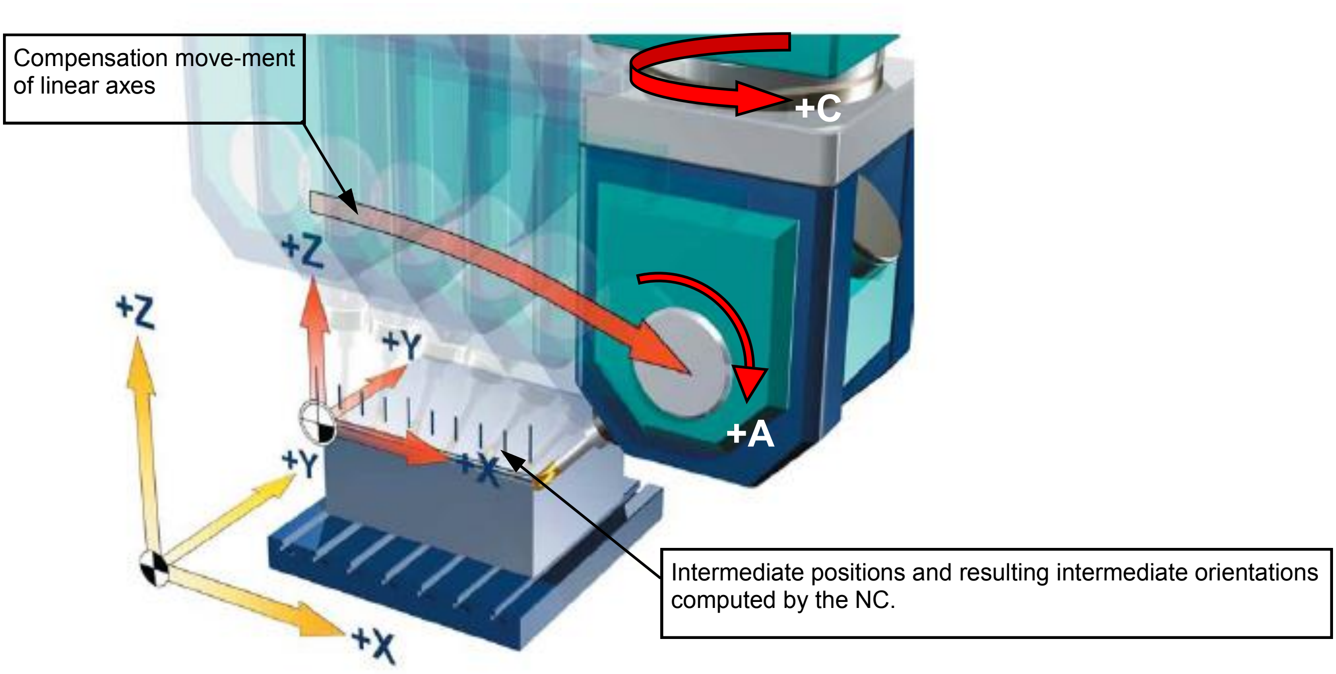
In this example a straight line is milled from X0 to X100. The tool orientation changes simultaneous from 90° to 45° in the ZX plane.

Note:

The tool path in the example is not compensated for the orientation change of the tool cutting edge. The orientation change takes place about the tool tip point (TTP), this results into contour gauging.
In general the CAM-system must calculate a 3 dimensional cutter radius compensation to modify the tool path for a change of the tool cutting edge orientation in space.

Notes

Example NC-programm with LEAD and TILT angles




```
N10 T="KUGEL_D8"
N11 M6
N12 S10000 M3 F1000         ;Technology data, tool, speed, etc
N13 G54 G0 A0 C90           ;Work offset, move rotary axes to starting point.
N14 TRAORI                  ;Call 5-axis transformation
N15 ORIWKS                  ;Orientation reference WCS
N16 ORIPATH                 ;Interpolation relative to path tangent
N17 CUT3DF                  ;3D cutter radius compensation (face milling)
N18 G0 X-4 Y0               ;Approach starting point in X,Y
N19 Z5                      ;Rapid to safety distance
N20 G1 Z0                   ;Infeed in Z
N21 G41 X0                  ;Approach move to activate cutter radius compensation
N22 LEAD=45 TILT=0          ;Set lead and tilt angle
N23 X100 A5=0 B5=0 C5=1     ;Linear move with orientation change to 45° in ZX-plane
N24 G40 X104 LEAD=0         ;Exit to deactivate cutter comp. tool parallel to surface normal
N25 G0 Z100                 ;Retract in Z
N26 TRAFOOF                 ;5-axis transformation off
N27 M30                     ;End of program
```

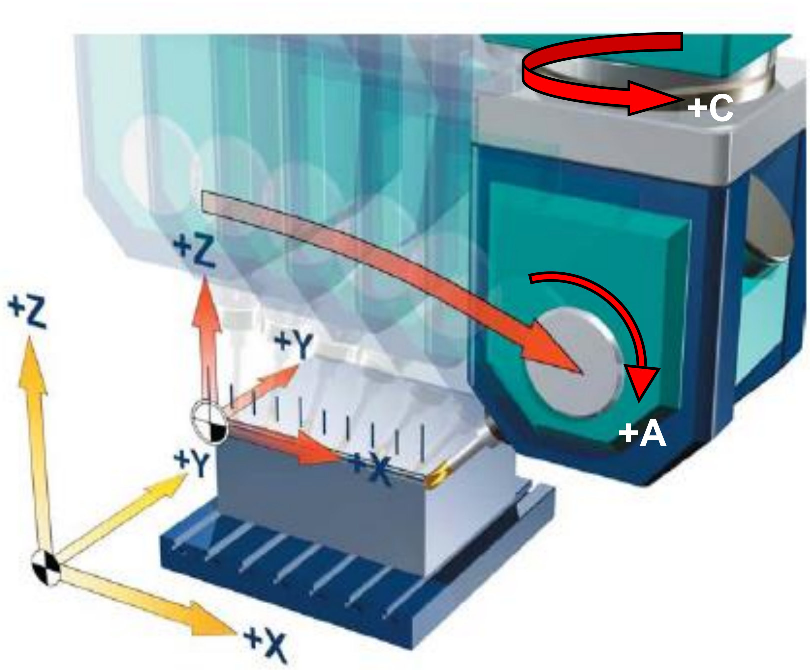
In this example a straight line is milled from X0 to X100. The tool orientation changes simultaneous from 90° to 45° in the ZX plane.

Note:

The tool path in the example is compensated for the orientation change of the tool cutting edge by the NC with CUT3DF. The orientation change takes place about the tool tip point (TTP), no contour gauging occurs. There is no need for the CAM -system to modify the tool path for a change of the tool cutting edge orientation in space.

Notes

Example NC-programm with direct rotary axes positions A.. B.. C..




```
N10 T="BALL_MILL_D8"
N11 M6
N12 S10000 M3 F1000      ;Technology data, tool, speed, etc.
N13 G54 G0 A0 C90        ;Work offset, move rotary axes to starting point.
N14 TRAORI               ;Call 5-axis transformation
N15 ORIMKS               ;Orientation reference MCS
N16 ORIAXES              ;Rotary axes interpolation
N17 G0 G54 X0 Y0 D1      ;Approach starting point in X,Y
N18 Z5                   ;Rapid to safety distance
N19 G1 Z0                ;Infeed in Z
N20 X100 Y0 A45          ;Linear move with orientation change to 45° in ZX-plane
N21 G0 Z100              ;Retract
N22 A0 C0                ;Zero return rotary axes
N23 TRAFOOF              ;Cancel 5-axis transformation
N24 M30                  ;End of program
```

In this example a straight line is milled from X0 to X100. The tool orientation changes simultaneous from 90° to 45° in the ZX plane.

.

Note:

The tool path in the example is not compensated for the orientation change of the tool cutting edge. The orientation change takes place about the tool tip point (TTP), this results into contour gauging.
In this case the CAM-system must calculate a 3 dimensional cutter radius compensation to modify the tool path for a change of the tool cutting edge orientation in space.

Notes

Programmable orientation smoothing ORISON

General explanation:

ORISON includes in general two features:

Smoothing of surfaces

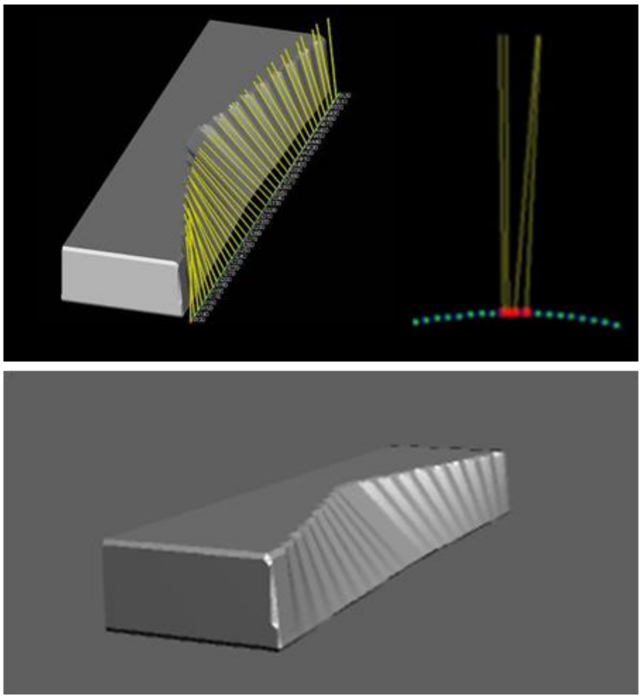
The function ORISON can be used to smoothen oscillations affecting orientations over several blocks in 5-axis programs. The aim is to achieve a smooth characteristic for both the orientation and the contour (see picture 5.11).

5-axis workpiece programs generated on a CAD/CAM-system with change in tool orientation usually contain minimal oscillations in tool alignment. Even if these deviations are only a couple tens of a degree, it can already lead to a compensation of the linear axes, which can cause a deceleration of the path motion or even a complete stop (see picture 5.12). The results are visible marks on the workpiece surface (see picture 5.10) and increase of machining time.

Improvement of machine dynamics

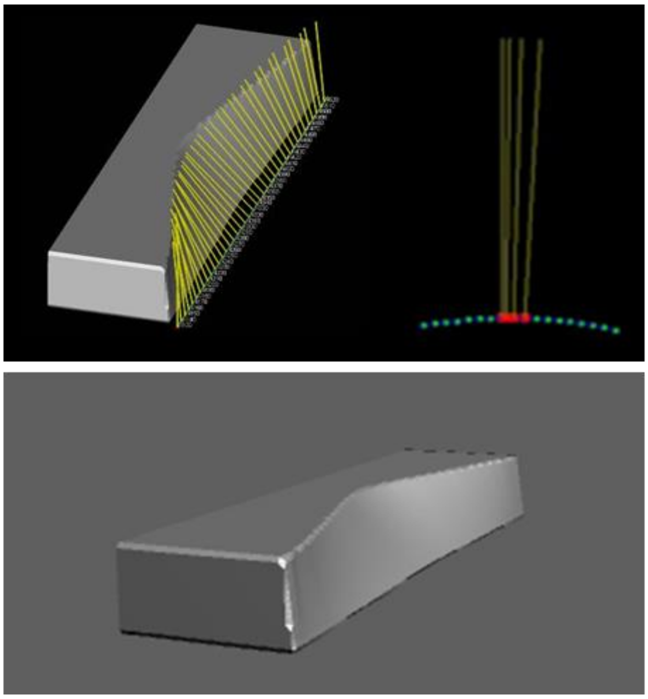
With ORISON it is possible to smoothen the tool orientation (vectors) independent of the contour (see picture 5.15), this allows the use of higher tolerances for the rotary axes which leads to an increase of the machining velocity (see picture 5.13), in other words reduction of machining time.

Without orientation smoothing
ORISOF



Picture 5.10: ORISOF

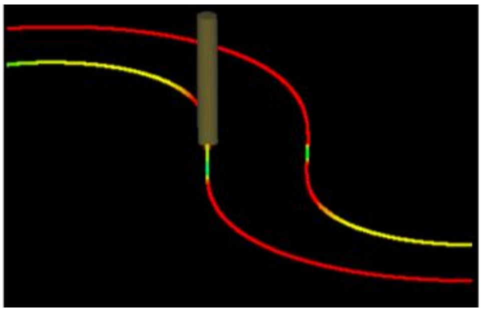
With orientation smoothing
ORISON

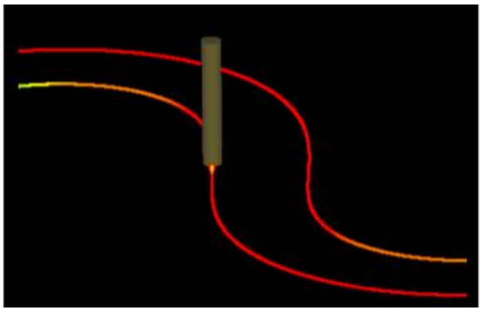


Picture 5.11: ORISON

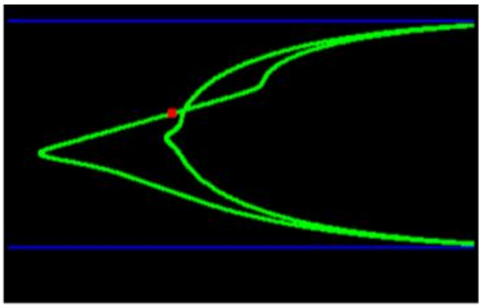
Notes

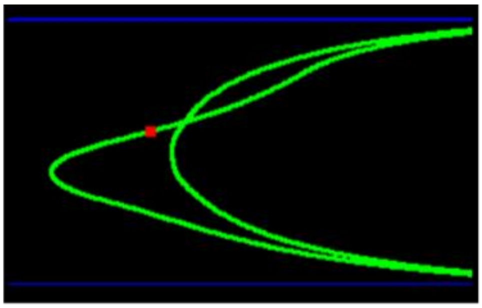
Characteristics of linear axis path


Picture 5.12: ORISOF


Picture 5.13: ORISON

Characteristics of rotary axis path


Picture 5.14: ORISOF


Picture 5.15: ORISON

Conditions:

The function ORISON can only be used in conjunction with 5-axis transformations TRAORI.
The function is not part of CYCLE832 or CUST_832 and therefore must be programmed separately in the workpiece program if vector smoothing is required.

Description of the NC commands:

NC command	Description
ORISON	ORISON = **Ori**entation **S**moothing **ON** Activates smoothing of the orientation vectors (modal).
ORISOF	ORISOF = **Ori**entation **S**moothing **OF** Deactivates smoothing of the orientation vectors.

Note:

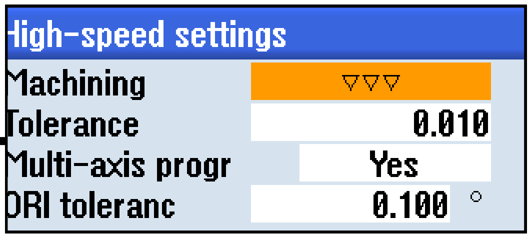
There are various ways on how the orientation smoothing tolerance for ORISON can be defined.

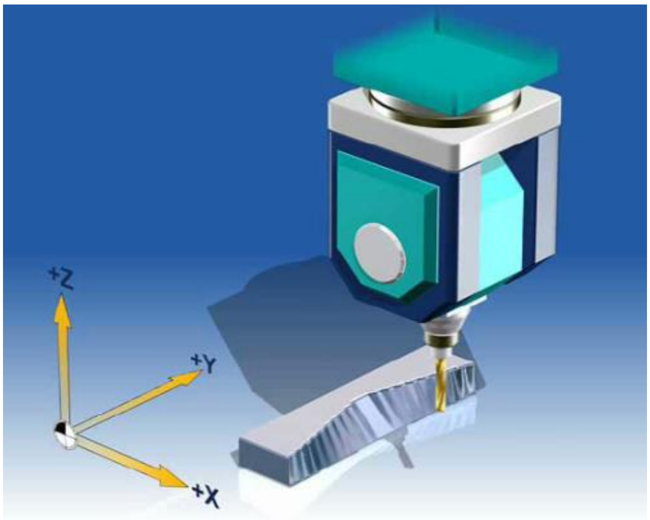
- Automatic calcualtion of orientation tolerance in CYCLE832 through specification of CAM-tolerance only (ORI tolerance = CAM tolerance * sqrt3 * 10)
- Orientation tolerance specified in CYCLE832 with parameter “ORI tolerance”.
- Orientation tolerance programmed after CYCLE832 in a separate NC block with OTOL=
- Orientation tolerance used from channel setting data ***SD42678 $SC_ORISON_TOL = 10*** (default) (only used if in ***MD20478 $MC_ORISON_MODE*** the thousands digit is set 1xxx.)

Notes

Example NC-Program:

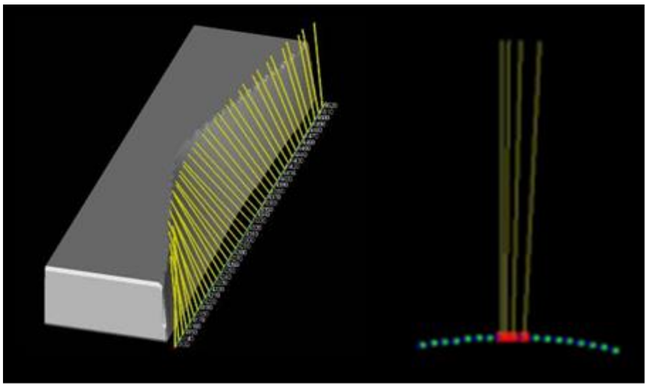
N30 TRAORI
N40 ORIAXES
N50 ORIWKS
N60 CYCLE832(0.01,_ORI_FINISH,0.1)
N70 ORISON
N110 G1 X0 A3=0 B3=0 C3=0.1 F1000
N130 X5 A3=0 B3=0.10 C3=0.99
N140 X10 A3=0 B3=0.10 C3=0.99
N150 X15 A3=0 B3=0.19 C3=0.98
N160 X20 A3=0 B3=0.19 C3=0.98
N170 X25 A3=0 B3=0.28 C3=0.95
N180 X30 A3=0 B3=0.28 C3=0.95
N190 X35 A3=0 B3=0.37 C3=0.92
N200 X40 A3=0 B3=0.37 C3=0.92
N210 X45 A3=0 B3=0.44 C3=0.89
N220 X50 A3=0 B3=0.44 C3=0.89
N230 X55 A3=0 B3=0.51 C3=0.85
N240 X60 A3=0 B3=0.51 C3=0.85
N250 X65 A3=0 B3=0.57 C3=0.81
N260 X70 A3=0 B3=0.57 C3=0.81
N270 X75 A3=0 B3=0.62 C3=0.77
N280 X80 A3=0 B3=0.62 C3=0.77
N290 X85 A3=0 B3=0.66 C3=0.74
N300 X90 A3=0 B3=0.66 C3=0.74
N310 X95 A3=0 B3=0.70 C3=0.70
N320 X100 A3=0 B3=0.70 C3=0.70
N330 X105 A3=0 B3=0.70 C3=0.71

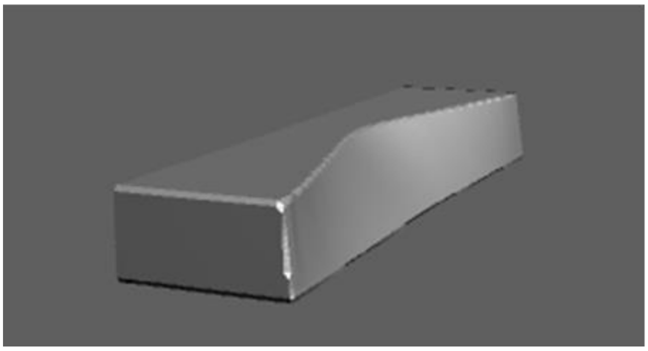




N340 X110 A3=0 B3=0.70 C3=0.71
N350 X115 A3=0 B3=0.66 C3=0.74
N360 X120 A3=0 B3=0.66 C3=0.74
N370 X125 A3=0 B3=0.62 C3=0.78
N380 X130 A3=0 B3=0.62 C3=0.78
N390 X135 A3=0 B3=0.57 C3=0.81
N400 X140 A3=0 B3=0.57 C3=0.81
N410 X145 A3=0 B3=0.51 C3=0.85
N420 X150 A3=0 B3=0.51 C3=0.85
N430 X155 A3=0 B3=0.44 C3=0.89
N440 X160 A3=0 B3=0.44 C3=0.89
N450 X165 A3=0 B3=0.37 C3=0.92
N460 X170 A3=0 B3=0.37 C3=0.92
N470 X175 A3=0 B3=0.28 C3=0.95
N480 X180 A3=0 B3=0.28 C3=0.95
N490 X185 A3=0 B3=0.19 C3=0.98
N500 X190 A3=0 B3=0.19 C3=0.98
N510 X195 A3=0 B3=0.10 C3=0.99
N520 X200 A3=0 B3=0.10 C3=0.99
N530 ORISOF
M30

Result:
Smoothing of surface and improvement of machine dynamics





Notes

Initial tool orientation setting ORIRESET

Initial tool orientation setting regardless of kinematics

With active 5-axis transformation **TRAORI** and machine independent programming of tool orientation, it can happen that depending on the current rotary axes position the machine starts the programmed machining operation at one of the two possible kinematic solutions. The solution which can be approached with shortest distance to the programmed orientation is always used. **ORIRESET** can be used to specify the initial position of up to 3 orientation axes with the optional parameters A, B, C, in this case the programmed rotary axes position determine which one of the two kinematic solutions is to be approached.

The order in which the programmed parameters are assigned to the rotary axes depends on the orientation axes order defined in the channel machine data for the transformation.

By programming ORIRESET (A, B, C), the orientation axes are moved in linear and synchronous motion from their current position to the specified initial setting position.

If an initial setting position is not programmed for an axis, a defined position from machine data ***$MC_TRAFO5_ROT_AX_OFFSET_1/2*** is used. Any active frames of rotary axes which may be present are ignored.

Note:

Only with active orientation transformation TRAORI(...) it is possible to program a initial setting of the tool orientation independent of the machine kinematics with ORIRESET(...) without alarm 14101.

Examples:

1. Example of machine kinematics AC (channel axis names A, C):

```
ORIRESET(90, 45)  ;A at 90 degrees, C at 45 degrees
ORIRESET(, 30)    ;A at $MC_TRAFO5_ROT_AX_OFFSET_1/2[0],
                  ;C at 30 degrees
ORIRESET( )       ;A at $MC_TRAFO5_ROT_AX_OFFSET_1/2[0],
                  ;C at $MC_TRAFO5_ROT_AX_OFFSET_1/2[1]
```

2. Example of machine kinematics ACB (channel axis names A, C, B):

```
ORIRESET(90, 45, 90)  ;A at 90 degrees, C at 45 degrees, B at 90 degrees
ORIRESET( )           ;A at $MC_TRAFO5_ROT_AX_OFFSET_1/2[0],
                      ;C at $MC_TRAFO5_ROT_AX_OFFSET_1/2[1],
                      ;B at $MC_TRAFO5_ROT_AX_OFFSET_1/2[2]
```

Example program

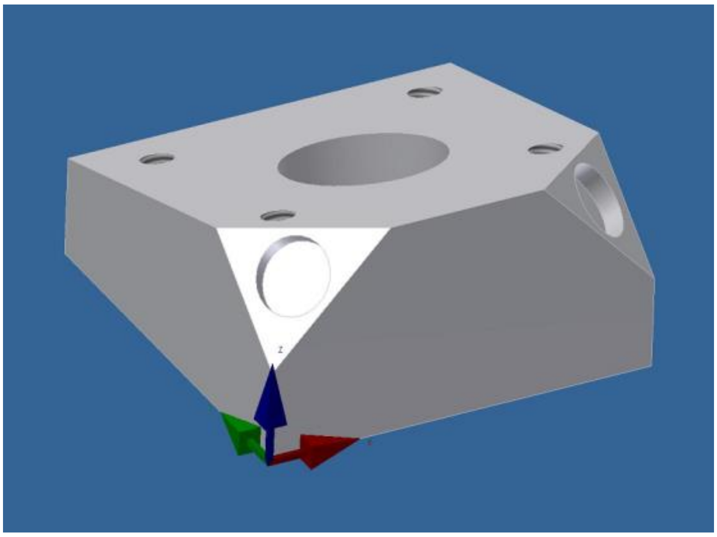
```
N111 TRAORI
N112 ORIWKS
N113 ORIAXES
N111 PTPWOC          ; Point To Point Without Compensation
N112 ORIRESET( -39.461,38.41)   ; Position rotary axes C and A to specified position
N113 CP              ; Continue Path
N114 G0 X-52.73538 Y-17.80536 Z31.9 A3=-.39485858 B3=.49800333 C3=.77206177
……..
```

Notes

Programming on the machine

3+2 axes positioning perpendicular to machining plane

Example 1: 3+2 axis machining with frame translation/rotation and TRAORI



Work Steps:

1. Face milling inclined plane (Plane 1)
2. Circular Pocket milling Ø40mm (Plane 1)
3. Spot face positions at -7 deg (Plane 1)
4. Spot face positions at +7 deg (Plane 1)
5. Drill positions Ø8.5mm at +7 deg (Plane 1)
6. Drill positions Ø8.5mm at -7 deg (Plane 1)
7. Tap positions M10x1.5 at -7 deg (Plane 1)
8. Tap positions M10x1.5 at +7 deg (Plane 1)
9. Face milling inclined plane (Plane 2)
10. Face milling inclined plane (Plane 3)
11. Circular pocket milling (Plane3)
12. Circular pocket milling (Plane 2)

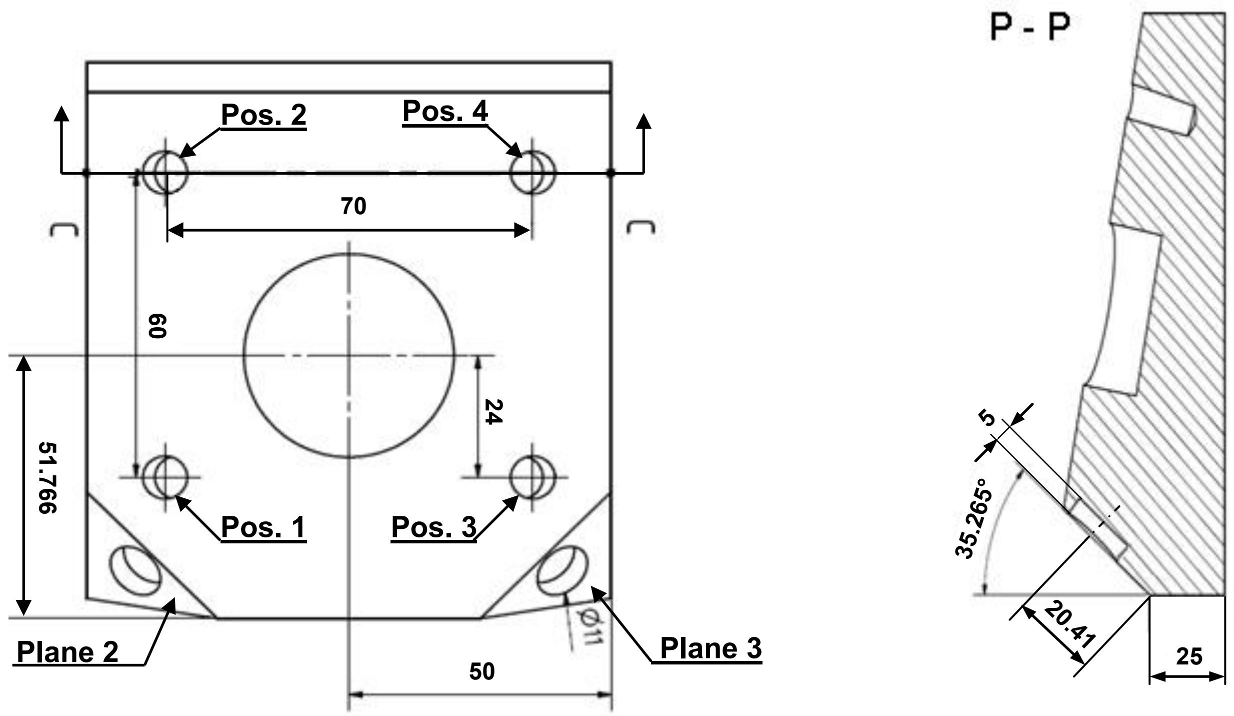
Tool List:

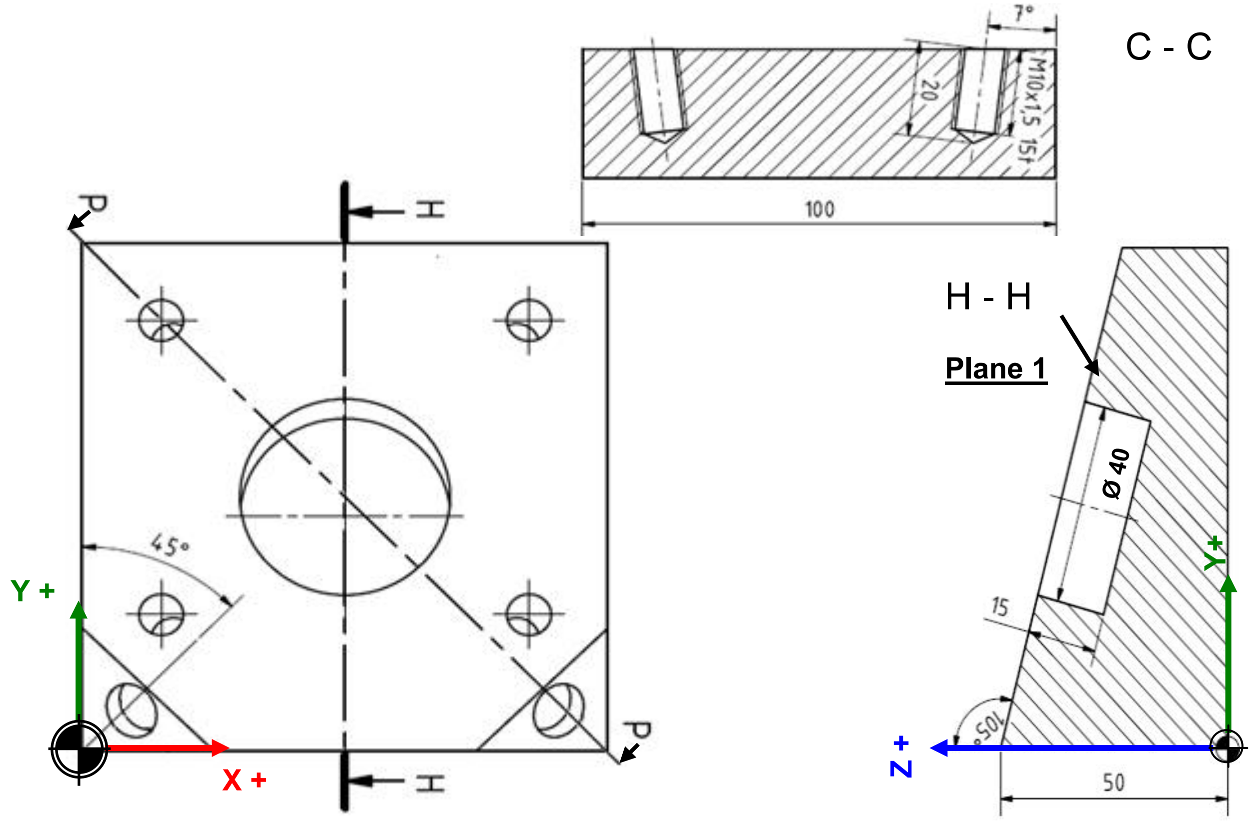
T10 (Indexable roughing endmill D=32)
T11 (Carbide endmill 2 Flutes D=16)
T12 (Carbide endmill 2 Flutes D=10)
T13 (Carbide Twist Drill 8.5 mm)
T14 (Tap M10x1.5)

Workpiece Blank:

Aluminium 100 x 100 x 50

Notes


Pos. 2
Pos. 4
70
60
24
51.766
Pos. 1
Pos. 3
Plane 2
Plane 3
Ø11
50
P - P
5
35.265°
20.41
25


C - C
7°
20
M10x1,5 15t
100
H
P
45°
Y +
X +
H - H
Plane 1
Ø 40
15
105°
Z +
Y +
50

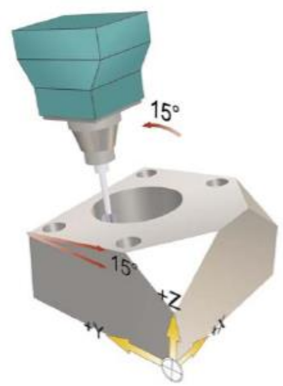
Notes

NC/WKS/M103_TRAORI/EXAMPLE_1_TRAORI.MPF

;* 3+2 axis positioning with TRAORI and FRAMES *****
N100 CYCLE800(2,"TABLE",200000,57,0,0,0,0,0,0,0,0,0,1,,1)
N101 CYCLE800()
N102 G54 G17
N103 WORKPIECE(,,,“BOX“, 112,0,50,-80,0,0,100,100)

N104 **T10 D1** ; Indexable roughing endmill D=30
N105 M6
N106 S8000 M3
N107 ORIWKS ;(Reference for transformation is WCS)
N108 ORIVECT ;(Vector interpolation)
N109 CUT2DF ;(2D cutter comp. in active frame)
N110 G0 G54 X0 Y0 Z100

N111 **MSG (“Set FRAME for Plane 1“)**
N112 **TRANS** Z50 ;(Absolute Shift of WCS to work piece top)
N113 **AROT** X-15 ;(Additive rotation of WCS about X plane1)
N114 **TRAORI** ;(Attention TTP is being tracked !)
N115 G0 X-20 Y10 M8 ;(Prepositioning tool in XY)
N116 G0 Z38.5 ;(Prepositioning tool in Z)
N117 G0 A3=0 B3=0 C3=1 ;(Set tool normal to working plane)



N118 **MSG (“Face milling inclined plane 1“)**
N119 CYCLE71(35.8,25.8,0,0,0,0,115,103.5,0,5,20,10,0,2000,11,5)
N120 G0 Z100
N121 **TRAFOOF** ;(Cancel 5-axis transformation)
N122 TRANS ;(Cancel all active programmable frames)

N123 **T11 D1** ;(SC-Endmill D=16 FL=2)
N124 M6
N125 S8000 M3
N126 G54 G0 X0 Y0 Z100

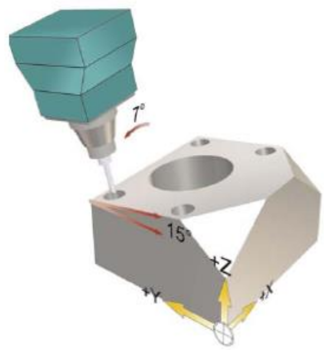
N127 **MSG (“Recalculate FRAME for Plane 1 after ATC“)**
N128 **TRANS** Z50 ;(Absolute Shift of WCS to work piece top)
N129 **AROT** X-15 ;(Additive rotation of WCS about X plane1)
N130 **TRAORI** ;(Attention TTP is being tracked !)
N131 G0 X50 Y51.76 M8
N132 Z30

N133 **MSG(“Pocket milling D40 plane 1“)**
N134 POCKET4(30,0,2,-15,20,50,51.76,5,0,0,1000,1000,0,21,10,,,8,2.5)
N135 G0 Z100
N136 **TRAFOOF** ;(Cancel 5-axis transformation)
N137 TRANS ;(Cancel all active programmable frames)
N138 G0 A0 C0 ;(Zero return rotary axes)

Notes

N140 **T12 D1** ;(SC-Endmill D=10 FL=2)
N141 M6
N142 S4000 M3 F500
N143 G54 G0 X0 Y0 Z100 D1

N144 **MSG (“Set FRAME for Plane of Position 1 and 2“)**
N145 **TRANS** Z50 ;(Absolute Shift of WCS to work piece top)
N146 **AROT** X-15 ;(Additive rotation of WCS about X-axis plane1)
N147 **ATRANS** X50 Y51.76 ;(Additive shift of WCS to pocket centre)
N148 **ATRANS** X-35 Y-24 ;(Additive shift of WCS to pos.1)
N149 **AROT** Y-7 ;(Additive rotation of WCS about Y-axis pos.1)
N150 **TRAORI** ;(Attention TTP is being tracked !)
N151 G0 X0 Y0 M8 ;(Prepositioning tool above pos.1)
N152 G0 Z20
N153 G0 A3=0 B3=0 C3=1 ;(Set tool normal to working plane)



N154 **MSG (“Spot face positions 1, 2 “)**
N155 MCALL CYCLE82(20,0,5,-3,,1)
N156 G0 X0 Y0
N157 G0 X0 Y60
N158 MCALL ;(Cancel modal cycle call)
N159 G0 Z100
N160 **TRAFOOF** ;(Cancel 5-axis transformation)
N161 TRANS ;(Cancel all active programmable frames)
N162 G0 A0 C0 ;(Zero return rotary axes)
N163 **MSG (“Set FRAME for Plane of Position 3 and 4“)**

N164 **TRANS** Z50 ;(Absolute Shift of WCS to work piece top)
N165 **AROT** X-15 ;(Additive rotation of WCS about X-axis plane1)
N166 **ATRANS** X50 Y51.76 ;(Additive shift of WCS to pocket centre)
N167 **ATRANS** X35 Y-24 ;(Additive shift of WCS to pos.3)
N168 **AROT** Y7 ;(Additive rotation of WCS about Y-axis)
N169 **TRAORI** ;(Attention TTP is being tracked !)
N170 G0 X0 Y0 ;(Prepositioning tool over pos. 3)
N171 G0 Z20
N172 G0 A3=0 B3=0 C3=1 ;(Set tool normal to working plane)

N173 **MSG (“Spot face positions 3, 4 “)**
N174 MCALL CYCLE82(20,0,5,-3,,1)
N175 G0 X0 Y0
N176 G0 X0 Y60
N177 MCALL ;(Cancel modal cycle call)
N178 G0 Z100
N179 **TRAFOOF** ;(Cancel 5-axis transformation)
N180 TRANS ;(Cancel all active programmable frames)

*WCS = Work coordinate system
*TTP = Tool tip point
*ATC = Automatic tool change

Notes

```
N181 T13 D1                    ;(Twist drill D=8.5")
N182 M6
N183 S4000 M3 F500

N184 MSG ("Recalculate FRAME for Plane pos. 1 and 2 after ATC)
N185 TRANS Z50                 ;(Absolute Shift of WCS to work piece top)
N186 AROT X-15                 ;(Additive rotation of WCS about X-axis plane1)
N187 ATRANS X50 Y51.76         ;(Additive shift of WCS to pocket centre)
N188 ATRANS X35 Y-24           ;(Additive shift of WCS to pos. 3)
N189 AROT Y7                   ;(Additive rotation of WCS about Y-axis)
N190 TRAORI                    ;(Attention TTP is being tracked !)
N191 G0 X0 Y0 M8               ;(Prepositioning tool above Pos. 3)
N192 G0 Z20

N193 MSG ("Deep hole drilling positions 3, 4 ")
N194 MCALL CYCLE83(20,0,5,-20,,-5,,,0,,1,1,3,5,,,1)
N195 G0 X0 Y0
N196 G0 X0 Y60
N197 MCALL                     ;(Cancel modal cycle call)
N198 G0 Z100
N199 TRAFOOF                   ;(Cancel 5-axis transformation)
N200 TRANS                     ;(Cancel all active programmable frames)
N201 G0 A0 C0                  ;(*** Zero return rotary axes ***)
```

```
N202 MSG ("Set FRAME for Plane of Position 1 and 2")
N203 TRANS Z50                 ;(Absolute Shift of WCS to work piece top)
N204 AROT X-15                 ;(Additive rotation of WCS about X-axis plane1)
N205 ATRANS X50 Y51.76         ;(Additive shift of WCS to pocket centre)
N206 ATRANS X-35 Y-24          ;(Additive shift WCS in XY Pos. 1***)
N207 AROT Y-7                  ;(Additive rotation of WCS about Y-axis pos.1)
N208 TRAORI                    ;(Attention TTP is being tracked !)
N209 G0 X0 Y0                  ;(Prepositioning tool above Pos. 1)
N210 G0 Z20
N211 G0 A3=0 B3=0 C3=1         ;(Set tool normal to working plane)

N212 MSG ("Deep hole drilling positions 1, 2 ")
N213 MCALL CYCLE83(20,0,5,-20,,-5,,,0,,1,1,3,5,,,1)
N214 G0 X0 Y0
N215 G0 X0 Y60
N216 MCALL                     ;(Cancel modal cycle call)
N217 G0 Z100
N218 TRAFOOF                   ;(Cancel 5-axis transformation)
N219 TRANS                     ;(Cancel all active programmable frames)
```

Notes

```
N220 T14 D1                  ;(Tap M10x1.5)
N221 M6
N222 S800 M3
N223 G54 X0 Y0 Z100

N224 MSG ("Recalculate FRAME for Plane of Position 1 and 2")
N225 TRANS Z50               ;(Absolute Shift of WCS to work piece top)
N226 AROT X-15               ;(Additive rotation of WCS about X-axis plane1)
N227 ATRANS X50 Y51.76       ;(Additive shift of WCS to pocket centre)
N228 ATRANS X-35 Y-24        ;(Additive shift of WCS to pos. 3)
N229 AROT Y-7                ;(Additive rotation of WCS about Y-axis pos.1)
N230 TRAORI                  ;(Attention TTP is being tracked !)
N231 G0 X0 Y0 M8             ;(Prepositioning tool above Pos. 1)
N232 G0 Z20

N233 MSG ("Tapping M10 positions 1, 2 ")
N234 MCALL CYCLE84(20,0,5,-15,,1,3,,1.5,0,800,800,3,1,0,0,,)
N235 G0 X0 Y0
N236 G0 X0 Y60
N237 MCALL                   ;(Cancel modal cycle call)
N238 G0 Z100
N239 TRAFOOF                 ;(Cancel 5-axis transformation)
N240 TRANS                   ;(Cancel all active programmable frames)
N241 G0 A0 C0                ;(Zero return rotary axes)
```

```
N242 MSG ("Set FRAME for Plane of Position 3 and 4")
N243 TRANS Z50               ;(Absolute Shift of WCS to work piece top)
N244 AROT X-15               ;(Additive rotation of WCS about X-axis plane1)
N245 ATRANS X50 Y51.76       ;(Additive shift of WCS to pocket centre)
N246 ATRANS X35 Y-24         ;(Additive shift of WCS to pos. 3)
N247 AROT Y7                 ;(Additive rotation of WCS about Y-axis pos.1)
N248 TRAORI                  ;(Attention TTP is being tracked !)
N249 G0 X0 Y0                ;(Prepositioning tool above Pos. 1)
N250 G0 Z20
N251 G0 A3=0 B3=0 C3=1       ;(Set tool normal to working plane)

N252 MSG ("Tapping M10 positions 3, 4 ")
N253 MCALL CYCLE84(20,0,5,-15,,1,3,,1.5,0,800,800,3,1,0,0,,)
N254 G0 X0 Y0
N255 G0 X0 Y60
N256 MCALL                   ;(Cancel modal cycle call)
N257 G0 Z100
N258 TRAFOOF                 ;(Cancel 5-axis transformation)
N259 TRANS                   ;(Cancel all active programmable frames)
N260 G0 A0 C0                ;(Zero return rotary axes)
```

Notes

```
N261 T10 D1              ;(Indexable roughing endmill D=30)
N262 M6
N263 S8000 M3
N264 G54 G0 X0 Y0 Z100

N265 MSG ("Set FRAME for Plane 2")
N266 TRANS X0 Z25        ;(Absolute Shift of WCS to left edge)
N267 AROT Z-45           ;(Additive rotation of WCS about Z-axis)
N268 AROT X54.735        ;(Additive rotation of WCS about X-axis plane 2)
N269 TRAORI              ;(Attention TTP is being tracked !)
N270 G0 X0 Y0 M8         ;(Prepositioning tool above corner)
N271 G0 Z50
N272 G0 A3=0 B3=0 C3=1   ;(Set tool normal to working plane)

N273 MSG ("Face milling corner in plane 2")
N274 CYCLE71(50,20,0,0,-20,0,65,35,0,5,20,5,0,2000,11,10)
N275 G0 Z100
N276 TRAFOOF             ;(Cancel 5-axis transformation)
N277 TRANS               ;(Cancel all active programmable frames)
N278 G0 A0 C0            ;(Zero return rotary axes)
```

```
N279 MSG ("Set FRAME for Plane 3")
N280 TRANS X100 Z25      ;(Absolute Shift of WCS to right edge)
N281 AROT Z45            ;(Additive rotation of WCS about Z-axis)
N282 AROT X54.735        ;(Additive rotation of WCS about X-axis)
N283 TRAORI              ;(Attention TTP is being tracked !)
N284 G0 X0 Y0            ;(Prepositioning tool above corner)
N285 G0 Z50
N286 G0 A3=0 B3=0 C3=1   ;(Set tool normal to working plane)

N287 MSG ("Face milling corner in plane 3")
N288 CYCLE71(50,20,0,0,-30,0,65,35,0,5,20,5,0,2000,11,10)
N289 G0 Z100
N290 TRAFOOF             ;(Cancel 5-axis transformation)
N291 TRANS               ;(Cancel all active programmable frames)

N292 T12 D1              ;(Carbide endmill 2 flutes D=10)
N293 M6
N294 S8000 M3
N295 G54 G0 X100 Y0 Z100

N296 MSG ("Recalculate FRAME for Plane 3 after ATC")
N297 TRANS X100 Z25      ;(Absolute Shift of WCS to right edge)
N298 AROT Z45            ;(Additive rotation of WCS about Z-axis)
N299 AROT X54.735        ;(Additive rotation of WCS about X-axis)
N300 TRAORI              ;(Attention TTP is being tracked !)
N301 G0 G54 X0 Y20.41 M8 ;(Prepositioning tool above corner)
N302 G0 Z50
```

Notes

```
N303 MSG ("Pocket milling D=11 in plane 3 ")
N304 POCKET4(50,0,2,-5,9,0,20.41,2.5,0,0,1000,1000,0,21,5,,,4,2.5)
N305 G0 Z100
N306 TRAFOOF                ;(Cancel 5-axis transformation)
N307 TRANS                  ;(Cancel all active programmable frames)
N308 G0 A0 C0               ;(Zero return rotary axes)

N309 MSG ("Set FRAME for Plane 2")
N310 TRANS X0 Z25           ;(Absolute Shift of WCS to left edge)
N311 AROT Z-45              ;(Additive rotation of WCS about Z-axis)
N312 AROT X54.735           ;(Additive rotation of WCS about X-axis)
N313 TRAORI                 ;(Attention TTP is being tracked !)
N314 G0 X0 Y20.41           ;(Prepositioning tool above corner)
N315 G0 Z50
N316 G0 A3=0 B3=0 C3=1      ;(Set tool normal to working plane)

N317 MSG ("Pocket milling D=11 in plane 2 ")
N318 POCKET4(50,0,2,-5,9,0,20.41,2.5,0,0,1000,1000,0,21,5,,,4,2.5)
N319 G0 Z100
N320 TRAFOOF                ;(Cancel 5-axis transformation)
N321 TRANS                  ;(Cancel all active programmable frames)
N322 G0 A0 C0               ;(Zero return rotary axes)
N323 G0 SUPA Z0 D0          ;(Retract tool to Max Z position in MCS)
N324 G0 SUPA X0 Y0 D1       ;(Retract Tool to Max XY position in MCS)
N325 M30
```

Note:

If their is a coordinate rotation (ROT or AROT) active right after a tool change, followed by a XY move, it is possible that a software limit alarm occurs before the first movement of the machining axes. Therefore it is recommended to position tool closer to the workpiece, before swivelling to avoid this problem.

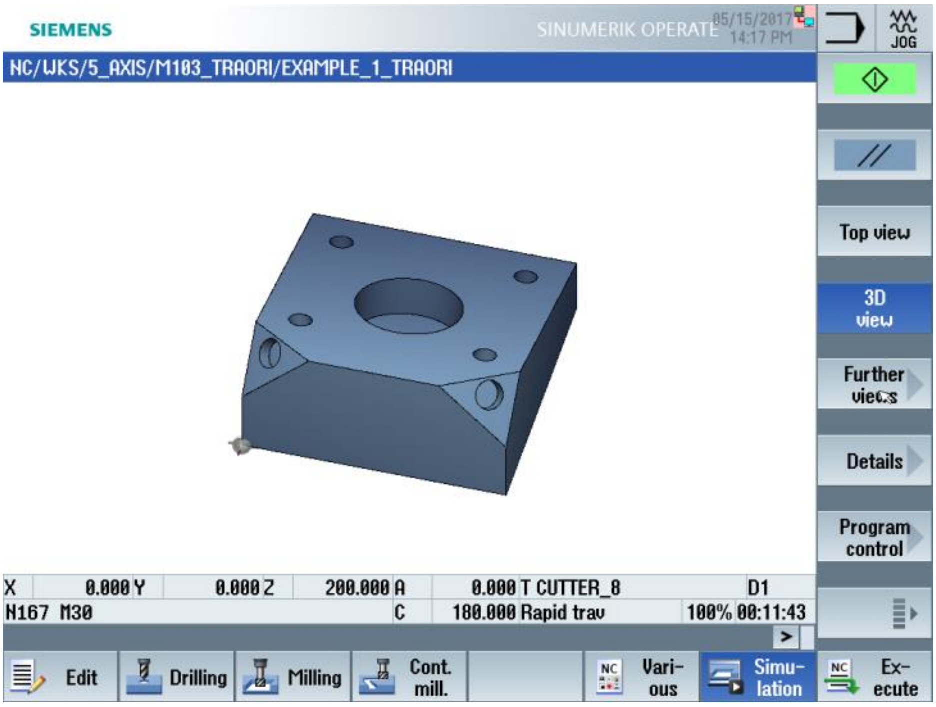
With active TRAORI the current position of the tool tip maintains it's relative position to the workpiece during rotary axes movement (TTP).

Note:

With the commands **ATRANS** and **AROT**, It is possible to shift and rotate a FRAME additively (incrementally) multiple times in row, in order to define a new work plane. With specification of the following Cartesian vectors it is possible to align the tool normal to the rotated work plane as defined below.

G0 A3=1 B3=0 C3=0 → Tool vector orientation normal to X-Axis
G0 A3=0 B3=1 C3=0 → Tool vector orientation normal to Y-Axis
G0 A3=0 B3=0 C3=1 → Tool vector orientation normal to Z-Axis

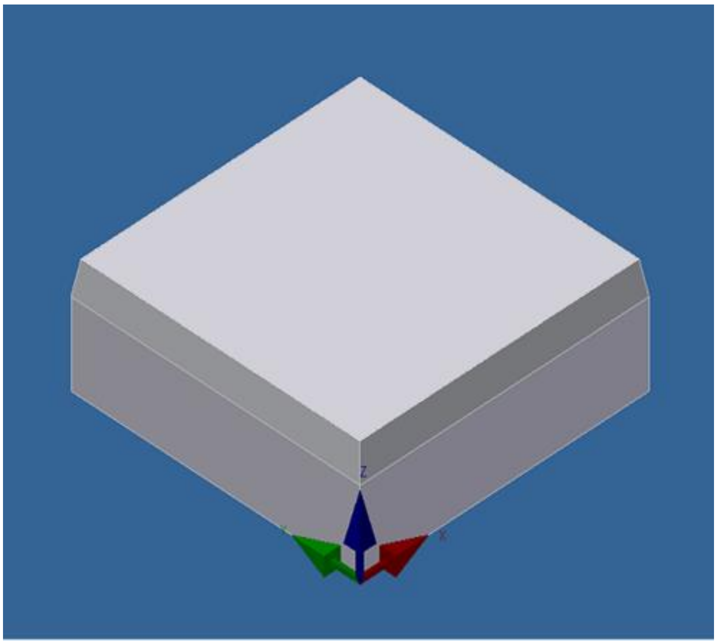
Notes

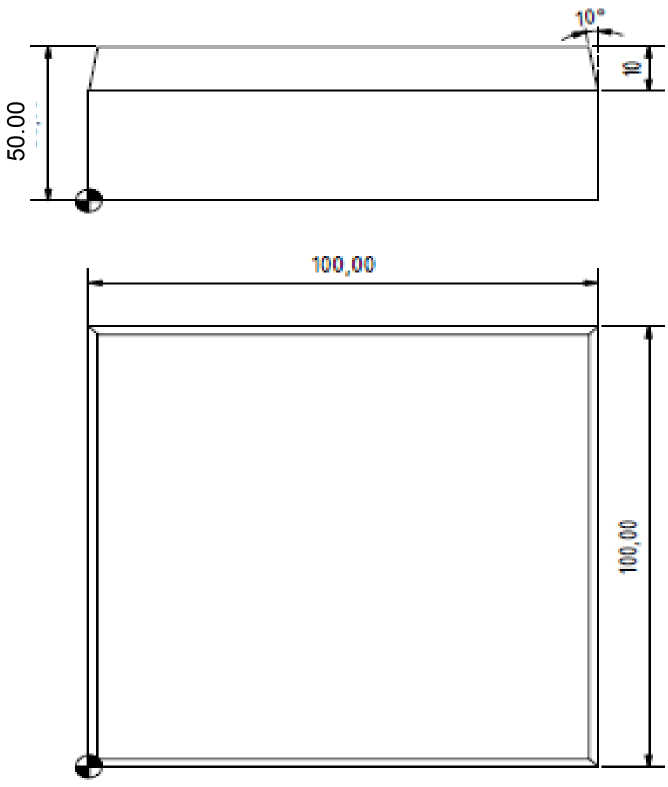

SIEMENS
SINUMERIK OPERATE
05/15/2017
14:17 PM
JOG
NC/WKS/5_AXIS/M103_TRAORI/EXAMPLE_1_TRAORI
Top view
3D view
Further views
Details
Program control
X 0.000 Y 0.000 Z 200.000 A 0.000 T CUTTER_8 D1
N167 M30 C 180.000 Rapid trav 100% 00:11:43
Edit
Drilling
Milling
Cont. mill.
Vari- ous
Simu- lation
Ex- ecute


SIEMENS
SINUMERIK OPERATE
05/15/2017
14:17 PM
JOG
NC/WKS/5_AXIS/M103_TRAORI/EXAMPLE_1_TRAORI
-100
-80
-60
-40
-20
0
40 20 0 -20 -40 -60 -80 -100 -120 -140 -160
Top view
3D view
Further views
Details
Program control
X 0.000 Y 0.000 Z 200.000 A 0.000 T CUTTER_8 D1
N167 M30 C 180.000 Rapid trav 100% 00:11:43
Edit
Drilling
Milling
Cont. mill.
Vari- ous
Simu- lation
Ex- ecute

Notes

5-axis circumferential milling with ORIVECT





Calculation of missing vector component

tan10 = OS / AS
tan10 = OS / 10

OS = tan10 *10 = 1.7632698

Notes

Task description:

Circumferential milling (side milling) of inclined walls under 10 degrees. Programming of great circle interpolation ORIVECT, tool orientation vectors A3=, B3=, C3= and 3D cutter compensation CUT3DC.

Tools: Endmill D=10 **Workpiece Blank:** Aluminium 100 x 100 x 50

Example 2: Circumferential milling with Cartesian vectors A3=, B3=, C3=

NC/WKS/M103_TRAORI/EXAMPLE_2_ORIVECT_VECTOR.MPF

```
;***VECTOR PROGRAMMING WITH ORIVECT***
N10 CYCLE800(2,"TABLE",200000,57,0,0,0,0,0,0,0,0,0,1,,1)
N20 CYCLE800()
N30 G54
N40 WORKPIECE(,,"","BOX",112,0,50,-80,0,0,100,100)
N50 T="CUTTER_10"
N60 M6
N70 S6000 M3
N80 TRANS Z50
N90 TRAORI
N100 ORIRESET(10,90) ;Position rotary axes in reference to MCS (A10,C90)
N110 ORIWKS
```

```
N120 ORIVECT
N130 CUT3DC          ;CUTTER COMP 3D CIRCUMFERENTIAL
N140 G54 G0 X-20 Y-20 D1
N150 G0 Z10
N160 G1 Z-10 F1000 M8
N170 G41 X0 A3=1.7632698 B3=0 C3=10
N180 G1 Y100
N190 G1 A3=0 B3=-1.7632698 C3=10
N200 G1 X100
N210 G1 A3=-1.7632698 B3=0 C3=10
N220 G1 Y0
N230 G1 A3=0 B3=1.7632698 C3=10
N240 G1 X0
N250 G40 X-20 Y-20 A3=0 B3=0 C3=1
N260 G0 Z100
N270 TRAFOOF
N280 M30
```

Notes

Example 2: Circumferential milling with orientation angle RPY A2=,B2=,C2=

NC/WKS/M103_TRAORI/EXAMPLE_2_ORIVECT_RPY_PROG.MPF

```
;***ORIVECT WITH RPY-angles***
N10 CYCLE800(2,"TABLE",200000,57,0,0,0,0,0,0,0,0,0,1,,1)
N20 CYCLE800()
N30 G54
N40 WORKPIECE(,,"","BOX",112,0,50,-80,0,0,100,100)
N50 T="CUTTER_10"
N60 M6
N70 S6000 M3
N80 TRANS Z50
N90 TRAORI
N100 ORIRESET(10,90) ;Position rotary axes in reference to MCS (A10,C90)
N110 ORIWKS          ;Tool orientation in reference to active frame
N120 ORIVECT         ;Great circle interpolation
N130 CUT3DC          ;CUTTER COMP 3D CIRCUMFERENTIAL
N140 G54 G0 X-20 Y-20 D1
N150 Z10
N160 G1 Z-10 F1000 M8
N170 G41 X0 A2=0 B2=10 C2=0
N180 G1 Y100
N190 G1 A2=10 B2=0 C2=0
```

```
N200 G1 X100
N210 G1 A2=0 B2=-10 C2=0
N220 G1 Y0
N230 G1 A2=-10 B2=0 C2=0
N240 G1 X-10
N250 G40 X-20 Y-20 A2=0 B2=0 C2=0
N260 G0 Z100
N270 TRAFOOF
N280 M30
```

Note:

For detailed information about 3D cutter compensation (CUT3DC) refer to module M105

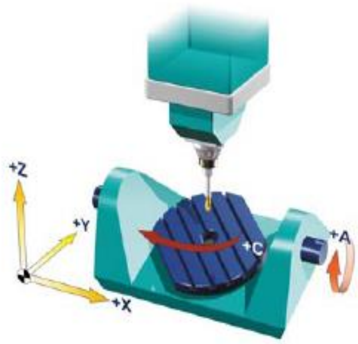
For interpretation of orientation angles as of RPY the following channel machine data must be set:
MD 21100 $MC_ORIENTATION_IS_EULER = 0

Notes

Example 2: Circumferential milling with direct rotary axes positions A,C

NC/WKS/M103_TRAORI/EXAMPLE_2_ORIVECT_DIRECT.MPF

```
;*** DIRECT ROTARY AXIS WITH ORIVECT AC-KINEMATIC***
N10 CYCLE800(2,"TABLE",200000,57,0,0,0,0,0,0,0,0,0,1,,1)
N20 CYCLE800()
N30 G54
N40 WORKPIECE(,,,"BOX",64,0,50,-80,0,0,100,100)
N50 T="CUTTER_10"
N60 M6
N70 S6000 M3
N80 TRANS Z50
N90 TRAORI
N100 ORIMKS        ;Tool orientation in reference to machine coordinate system
N120 ORIVECT       ;Great circle interpolation (also possible with ORIAXES)
N130 CUT3DC        ;CUTTER COMP 3D CIRCUMFERENTIAL
N140 G54 G0 X-20 Y-20 A0 C0 D1
N150 Z10
```




```
N160 G1 Z-10 F1000 M8
N170 G41 X0 A10 C90
N180 G1 Y100
N190 G1 C0 A10
N200 G1 X100
N210 G1 C270 A10
N220 G1 Y0
N230 G1 A10 C180
N240 G1 X-10
N250 G40 X-20 Y-20 A0 C0
N260 G0 Z100
N270 TRAFOOF
N280 M30
```

Note:

The rotary axes are here programmed in reference to the MCS. A rotary axes offset in the storable work offset for the rotary axis A and C (e.g. in G54) is only taken into consideration, if the following channel machine data is set ***MD24590 $MC_TRAFO5_ROT_OFFSET_FROM_FR_1 = 1***.
Attention! A possible rotation of the Frame in this case will be ignored.

Resetting the initial orientation with **ORIRESET** is here not necessary, since the sign of the first rotary axes positions define already the initial orientation.

The program in our example is **dependent on the machine kinematics** “AC swivel rotary table” (see picture above).

Notes

Example 2: Circumferential milling with Cartesian vectors A3=,B3=,C3= and programmable tool offset correction TOFFL and TOFFR

NC/WKS/M103_TRAORI/EXAMPLE_2_ORIVECT_TOFFL.MPF

```
;***VECTOR PROGRAMMING WITH MULTIPLE CUTS AND SUBSEQUENT FINISHING***
N100 CYCLE800(2,"TABLE",200000,57,0,0,0,0,0,0,0,0,0,1,,1)
N101 CYCLE800()
N102 G54
N103 WORKPIECE(,,"","BOX",112,0,50,-80,0,0,100,100)
N104 T="CUTTER_10"
N105 M6
N106 S6000 M3
N107 TRANS Z50
N108 TRAORI
N109 ORIWKS
N110 ORIVECT
N111 CUT3DC
N112 TOFFL= 7.5 ; Programmable tool correction offset (length)
N113 TOFFR=0.2 ; Programmable tool correction offset (radius)
N114 G54 G0 X-20 Y-20 Z10
N115 _START:
N116 G1 Z-10 F1000 M8
N117 G41 X1.7632698  A3=1.7632698  B3=0 C3=10
N118 G1 Y98.2367302
N119 G1 A3=0 B3=-1.7632698   C3=10
N120 G1 X98.2367302
```

```
N121 G1 A3=-1.7632698  B3=0 C3=10
N122 G1 Y1.7632698
N123 G1 A3=0 B3=1.7632698  C3=10
N124 G1 X-10
N125 G1 G40 X-20 Y-20 A3=0 B3=0 C3=1
N126 _END:
N127 TOFFL=5
N128 REPEAT _START _END
N129 TOFFL=2.5
N130 REPEAT _START _END
N131 TOFFL=0
N132 REPEAT _START _END
N133 TOFFR=0
N134 REPEAT _START _END
N135 G0 Z100
N136 TRAFOOF
N137 M30
```

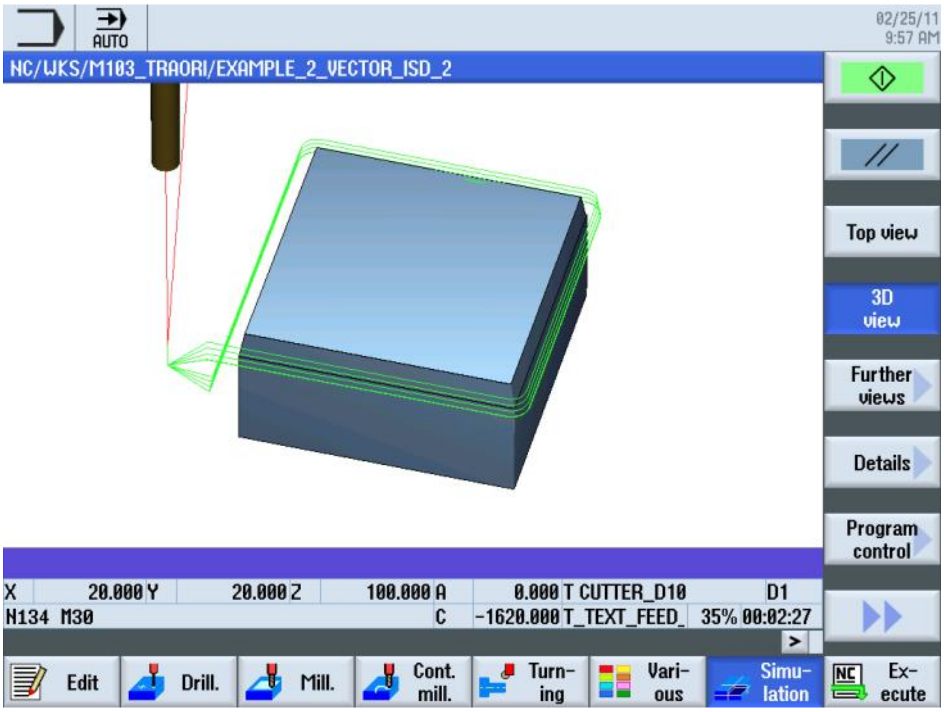
Note:

A detailed explanation of the G-code commands TOFFL= and TOFFR= can be found in Section 5.2 „Information for the programmer“.

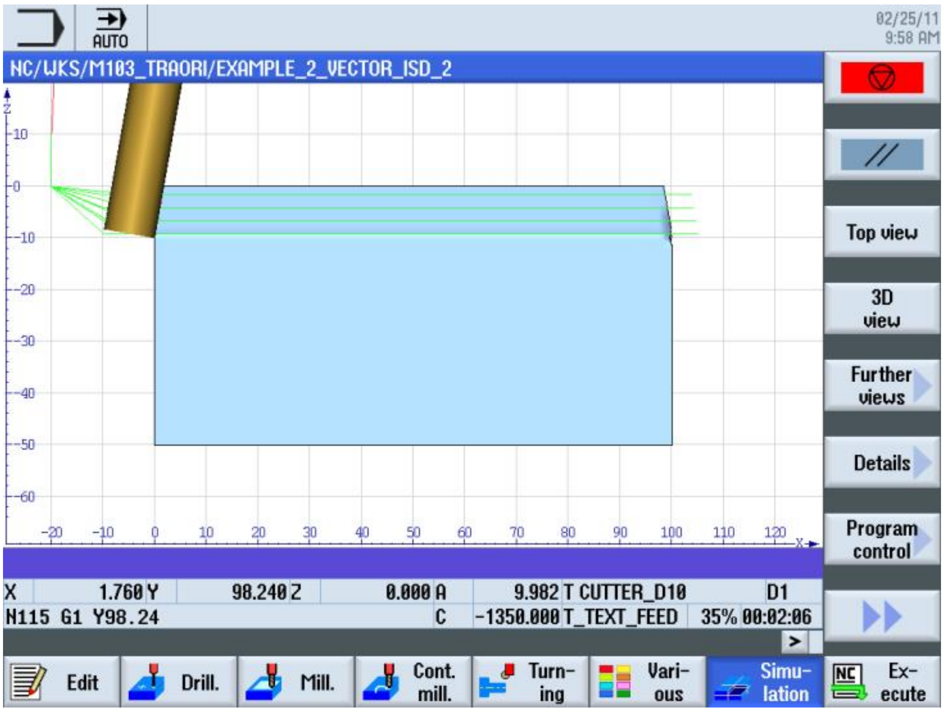
Notes

Simulation in operating area PROGRAM

3D view



Further views, From side



Notes

Programming on the CAM

Process chain for producing 5-axis workpieces

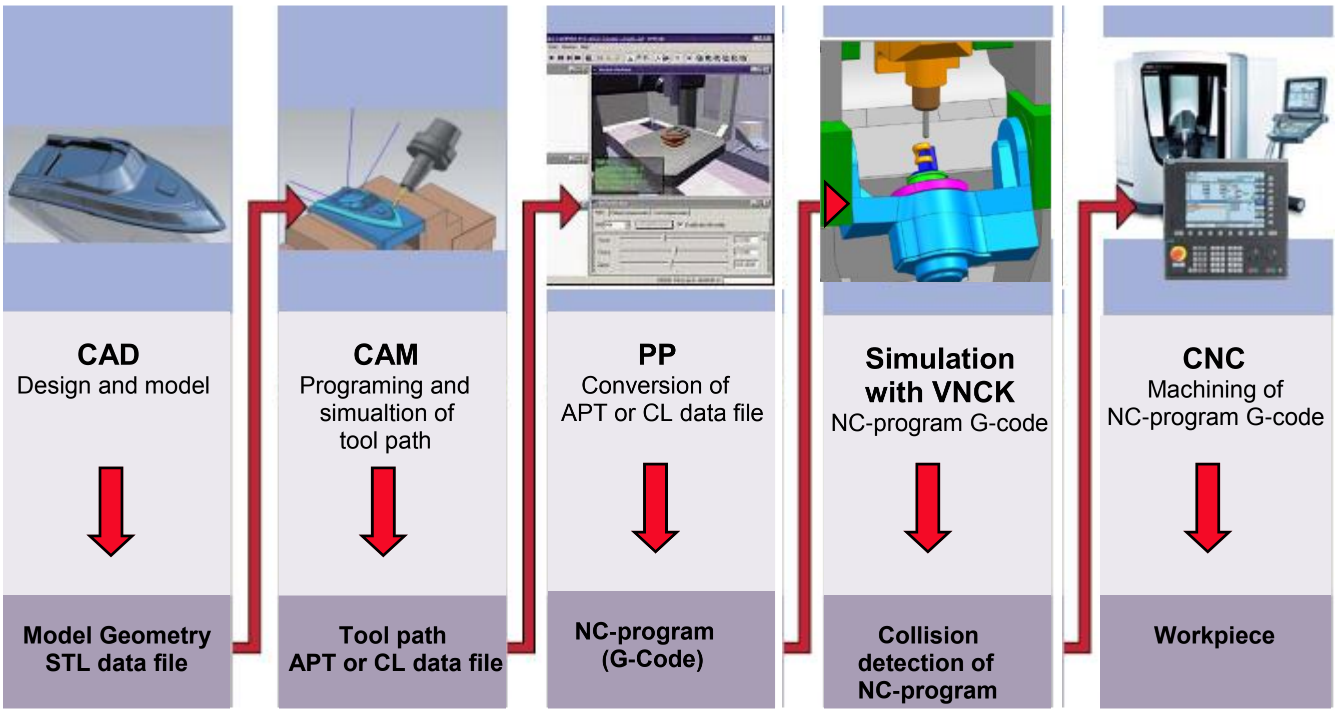
Especially within the context of mold making machining, the entire CAD/CAM/CNC process chain plays a major role in terms of ensuring optimum results on the machine.

The CAD system generates the geometry of the desired workpiece. Based on this geometry file, the CAM system generates the corresponding machining strategy with the associated technology information.

The data format output from the CAM system is generally an APT or CL data file. This is converted into an executable NC code in the postprocessor.

The upstream postprocessor is of the utmost importance when it comes to using the capabilities and performance of SINUMERIK controls to the full.

The postprocessor should ensure that the higher-order functions of SINUMERIK controls (as described in this module) are activated in the best possible way. An overview of all higher-order SINUMERIK functions is to be used as described in the modules M102, M103, M104 and M105. The most important functions are once more briefly introduced in the following chapters.



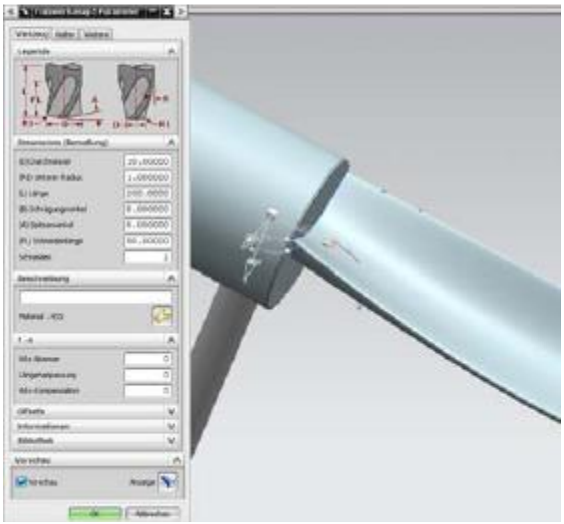
Notes

CAM systems

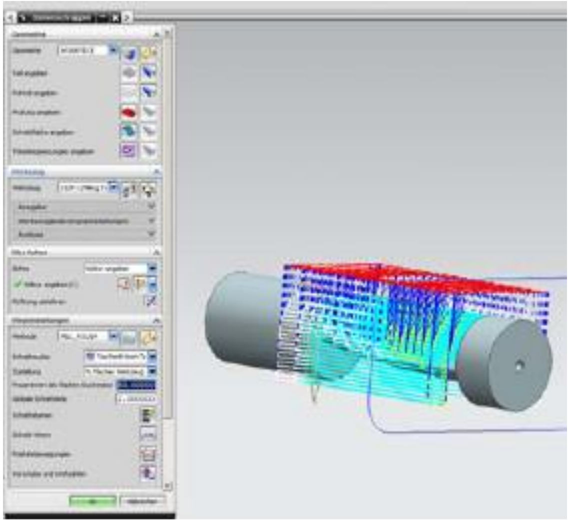
Within the context of the process chain, it is the CAM system that is responsible for the key task of generating the tool path data. The quality of this data plays a decisive role in determining the results of workpiece production.

This section outlines the procedure for generating the tool path. Given the diverse range of systems available on the market, only a brief summary can be provided.

Tool definition



Machining strategies



Procedure when working with a CAM system:

- **CAD data**
  Read CAD data into the CAM system. When reading the data in, a check should be performed to ensure that the surface geometries are free of defects, i.e. that there are no steps or jumps. Flaws in the data such as these will manifest themselves on the surface of the finished workpiece.

- **Chucking situation**
  Define the chucking situation and the geometry that can be freely machined in a chucking device. Define geometries such as a zero point.

- **Specify tools**
  Define the necessary tools on the basis of the machining task at hand and enter the technology data. As a general rule, CAM systems are able to read the data in from tool databases. Amongst other things, the tools determine which subsequent machining strategies can be used, e.g. whether plunge cutting is possible with the tool.

- **Specify machining strategies**
  Define the machining process for the various workpiece geometries using the appropriate strategies. First of all, roughing strategies are applied, e.g. roughing in the Z plane or equidistant from the surface geometry. CAM systems offer various options from 2 1/2-axis to 5-axis machining, such as 5-axis face milling with constant tool setting angle normal to surface, or 5-axis circumferential milling of free form surfaces with or without 3D tool radius compensation.
  .

Notes

The machining strategies are built up step-by-step in this procedure and supported, for example, by automatic residual material detection.

Important parameters
When working with CAD/CAM systems, certain tolerances and levels of accuracy that will have an impact on subsequent machining must be observed.

Tolerance
The CAM system uses the CAD surface (spline) to generate a contour consisting of linear tra– versing blocks (straight line elements). The extent to which the linear contour deviates from the real contour from the CAD system is known as the chord error or chord tolerance. This tolerance depends on the strategy used and is greater in the case of roughing strategies than with finishing strategies. When the NC programs are executed on the machine, the tolerance is specified by the CAM system and should be passed over into **CYCLE832 ratio 1:1**, so that optimum results can be achieved in terms of surface quality and contour accuracy.

Accuracy
When outputting the NC blocks from the CAM system, you can specify the number of decimal places. The required level of accuracy is dependent on the type of interpolation. In the case of linear axes (X, Y, Z), at least 3 decimal places should be used for 3-axis programs.
If the blocks are to be output as rotary axis positions, with 5-axis programs 5 decimal places should be used in the linear and rotary axes for optimum surface quality. If they are to be output in the form of a direction vector, we recommend 5 decimal places in the linear axes and at least 6 decimal places for the direction vectors.

- **Calculation and simulation of tool path (CL or APT file)**
  When calculating the tool path, different levels of tool path accuracy can be used by setting the appropriate chord tolerances. The simulation on the CAM is only suitable to verify the calculated tool path, a complete simulation of all machine-specific and control-specific data and machine kinematics is not considered in the simulation. Potential collisions cannot be detected accurately and can therefore not be avoided.

- **Output of the NC code through the postprocessor**
  The postprocessor converts the CAM generated tool path data (CL or APT file) into the CNC readable NC program (G-Code), taking into account the control-specific syntax and the control's special functions. Control-specific functions (e.g. CYCLE800, CYCLE832, TRAORI, ...) should be integrated into the postprocessor as much as possible in order to utilize the control system/machine to its full potential. For this purpose, CAM systems make use of universal postprocessors or special post processors that have been optimized for the SINUMERIK system. Manufacturer-specific functions such as separate coolant strategies must be implemented in the postprocessor in consultation with the machine manufacturer.

- **Machine simulation of the NC program (G-code) with VNCK virtual machine**
  Various software suppliers or CAM manufacturers offer a virtual machine simulation software tool. After generating the NC-program (G-code program) through the postprocessor it is possible to simulate the actual machine traversing movements under consideration of all machine-specific and control-specific data and machine kinematics. Here, potential collisions can be detected and avoided, for example the machine's maximum axis traversing ranges can be taken into account.

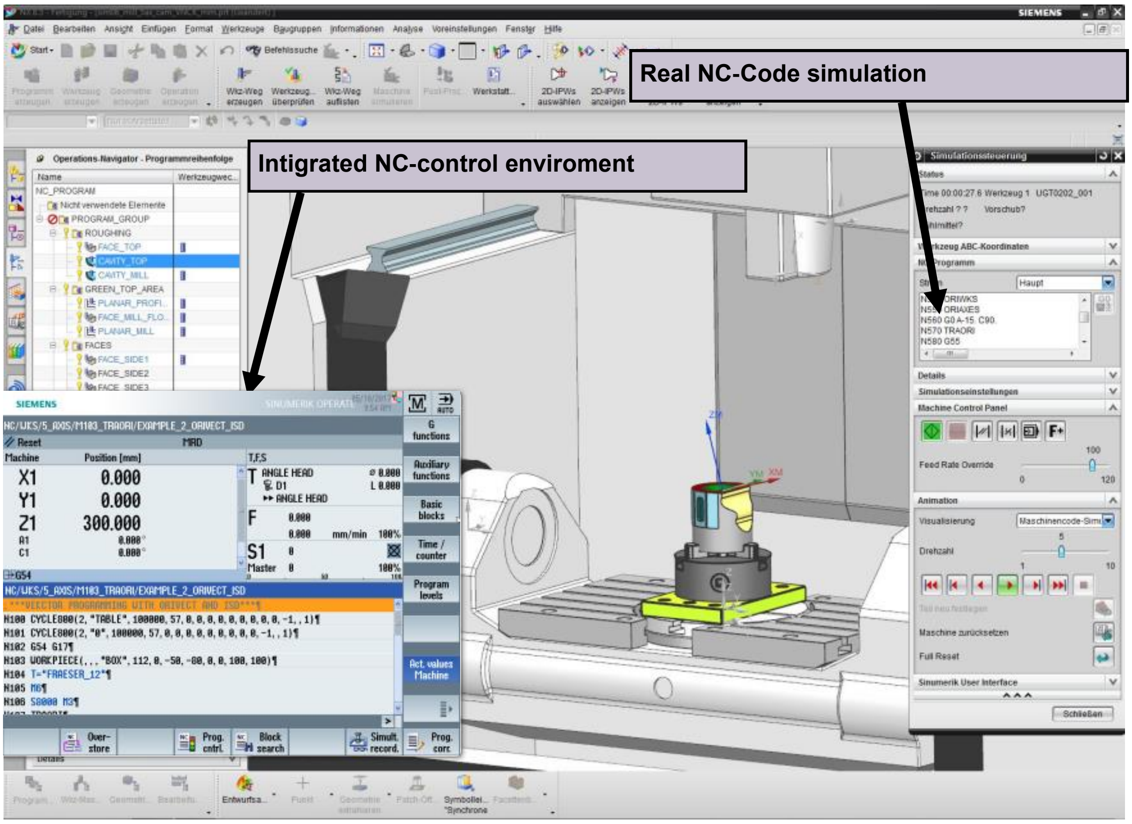
Notes

To ensure that the data obtained is as realistic as possible, virtual models of the machine, tools, workpiece clamping and control are created and simulated. The accuracy of the simulation can only be as good as the accuracy of the imported data. The simulation can only be realistic if the virtual NC kernel (VNCK) of the control provided by SIEMENS is integrated into the simulation software.

Using these basic modules and other components, such as the CAD data for the real machine, the machine manufacturer or CAM system manufacturer can create a virtual machine that resembles the real machine as closely as possible.

Using the virtual machine in conjunction with the SIEMENS VNCK offers many advantages:

- Programming errors are detected immediately.
- Program simulation with calculation of the actual time so that production effort can be estimated more easily.
- Collision checking with actual tools, chucking devices, and machine geometries
- While the current production process is underway, the workpiece can be programmed, optimized and then implemented on the machine immediately.
- Shorter setup times.
- Can be used for training and instruction. New machines can be programmed without any risk.



Notes

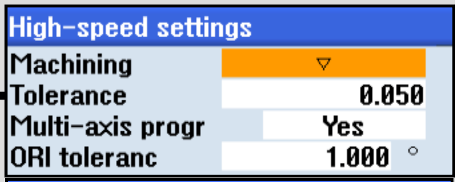
Program structure for 5-axis machining with CYCLE832

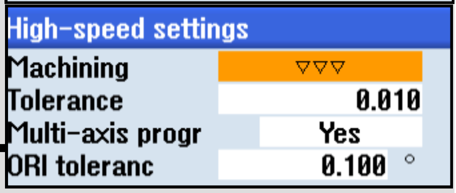
For machining purposes, a main program is generated (**1**) that includes all technology data. The main program calls one or more subprograms (**2**) and (**3**), that contain the workpiece' s geometry data. The tool change defines how the content is divided into subprograms.

Main pro-

Mainprogram.MPF (1)

```
N100  T1 D1
N101  M6
N102  S10000 M3
N103  G54
N104  CYCLE832(0.05,_ORI_ROUGH,1)  (4)
N106  EXTCALL "CAM_ROUGH"  (5)
N107  T2 D1
N108  M6
N109  S15000 M3
N110  G54
N111  CYCLE832(0.01,_ORI_FINISH,0.1)  (4)
N112  EXTCALL "CAM_FINISH"  (5)
N113  M30
```





Subprogram

CAM_ROUGH.SPF (2)

```
N100   TRAORI
N101   ORISON  (6)
N102   G0 X0 Y0 Z10
N103   G1 Z0 F500
N104   G1 X-1.45345 Y0.67878 F10000
N105   G1 X-1.18141 Y0.84245
.......
N5046 G1 X-4.11845 Y-11.44212
N5047 G0 Z10
.......
N5051 G1 Z-2.13247 A3=0.34202 B3=0 C3=-0.93969 F800
.......
N6582 G1 X7.60978 Y3.55541 A3=0.34202 B3=0 C3=-0.9396
N6583 G0 Z50 A3=0.34202 B3=0 C3=0.93969
N6584 TRAFOOF
N6585 M17
```

Subprogram

CAM_FINISH.SPF (3)

```
N100   TRAORI
N101   ORISON  (6)
N102   G0 X0 Y0 Z10 A3=0 B3=0 C3=0
N103   G1 Z0 F500
......
```

Notes

Main program:

The main program includes the two key functions for milling, CYCLE832 (**4**) and EXTCALL (**5**).

CYCLE832 “High Speed Settings”:

CYCLE832 (**4**) has been specifically developed for the program structure shown, where technology data and geometry data are separated. It brings together all the key commands and activates control functions for 3 and 5-axis machining in the high speed machining range (HSC). A selection can be made between 4 different types of machining that activate different dynamic parameters.

For the roughing program "CAM_Rough" using T1, the parameters in CYCLE832 were geared towards achieving a high velocity.

For the finishing program "CAM_Finish" using T2 the parameters in CYCLE832 were geared towards achieving a high level of accuracy and surface quality.

With additional selection of “**Multi-axis program**” a additional input field for specification of the orientation smoothing tolerance is opened. The value can also be programmed with the command OTOL=
in the program after CYCLE832.

A detailed explanation of CYCLE832 “High Speed Settings” can be found in module M104

EXTCALL:

CAM programs are generally extremely large, which is why they are stored in an external memory. The EXTCALL command (**5**) is used to call the subprograms from various locations, including external memories. All programs should be located in the same directory. If this is not the case, the paths must also be specified during the call.

ORISON / OTOL:

The NC command ORISON (**6**) is a vector smoothing function that has been specially developed for the 5-axis area. This function can be used to smooth fluctuating orientation across several blocks.
The aim is to achieve a smooth characteristic for the orientation and a more harmonious movement of the axes. Since this command is not part of CYCLE832, it is recommended to program ORISON after switching on the 5-axis transformation TRAORI.

The OTOL command can be used to define the orientation tolerance for vector smoothing with ORISON. The value is entered in degrees. The recommended value for finishing is 0.5 degrees.
The programmed value with OTOL=... over writes the specified value of the parameter
“**ORI tolerance**” in CYCLE832.

There are various ways on how the orientation smoothing tolerance for ORISON can be defined.

- Automatic calcualtion of orientation tolerance in CYCLE832 through specification of CAM-tolerance only (ORI tolerance = CAM tolerance * sqrt3 * 10)
- Orientation tolerance specified in CYCLE832 with parameter “ORI tolerance”.
- Orientation tolerance programmed after CYCLE832 in a separate NC block with OTOL=
- Orientation tolerance used from channel setting data ***SD42678 $SC_ORISON_TOL = 5*** (default)
  (only used if in ***MD20478 $MC_ORISON_MODE*** the thousands digit is set 1xxx.)

Attention ! If the orientation tolerance is selected too high for roughing operations, collisions with neighboring workpiece geometries can occur.

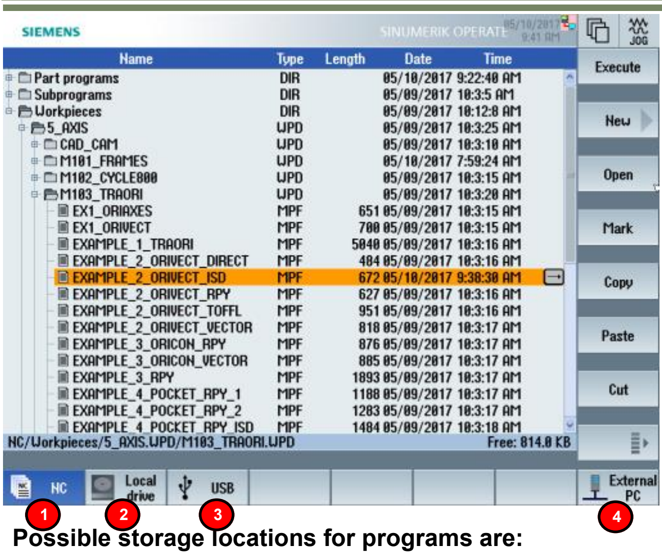
Notes

<u>***Program storage/data transfer***</u>

Program manager

The program manager offers you an optimum overview of the directories and programs, and very easy-to-use file handling. It supports plain text names of up to 24 characters for directories and files. For the SINUMERIK 828D and on the NC, subdirectories can also be managed on external storage media such as CF cards and USB flash drives.

All storage media including the network drives are displayed in the program manager. The part programs can be edited in all media. You can create, paste, copy, delete and cut programs via the horizontal softkey bar.



Possible storage locations for programs are:

1. NC
2. Local drive (CompactFlash card or hard drive PCU50)
3. USB drives
4. Network drives

<u>***Note:***</u>

In the program manager, you can use standard Windows short-cuts such as CTRL+C, CTRL+X, and CTRL+V.

Notes

External storage media - data transfer

NC programs are stored in the control, if required, downloaded into the NCK working memory (RAM), and executed on the machine.

Mold-making programs are often too large for the NC memory or they cannot be processed. Therefore these are swapped to an external memory and processed successively. In the main program, an EXTCALL command is programmed, which calls up the swapped-out program according to the network path on the server, the USB port, hard drive, etc.

Procedure when calling the geometry program using EXTCALL:

Program the geometry program call, e.g. SAMPLE in the main program. The call differs depending on the control and where the data is saved.

- The subprogram is located on the hard drive (NC) EXTCALL "SAMPLE"
- The subprogram is located in the directory on the CompactFlash card EXTCALL "CF_CARD:/PROGRAMS/SAMPLE.SPF"
- The subprogram is located on a local hard drive EXTCALL "LOCAL_DRIVE:/PROGRAMS/SAMPLE.SPF"
- The subprogram is located on a USB flash drive EXTCALL "USB:/MOLD_DIE/CAM_SCHRUPP.SPF"

- Network connected with Ethernet and path is specified in the channel setting data ***SD 42700 $SC_EXT_PROG_PATH*** e.g. on a server "//R4711/workpieces/subprograms". The default setting is optional. The directory can first be specified when making the call with EXTCALL.
  EXTCALL “SAMPLE.SPF”

Processing of USB flash drive/CF card

On the controller there is a USB port on the front, and on the SINUMERIK 828D there is a USB port on the front side and on the rear side. On the SINUMERIK 828D, there is a CompactFlash card slot on the front. On the SINUMERIK 840D sl, it is located on the rear side.

- Storage media can be inserted or removed during operation, i.e. the machine does not have to be restarted in order for the storage medium to be recognized
- Loading, editing and executing of part programs from the storage medium.
- When executing part programs from a storage medium there is no loss of speed (DNC operation), in which case executing from a CF card is recommended.
- No special software is necessary for reading from or writing to the storage medium on the PC.

Note:

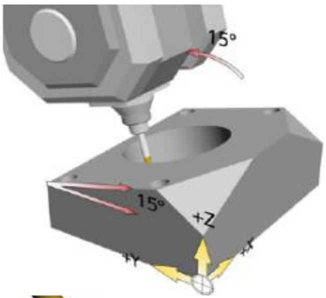
Direct execution from a USB flash drive is not recommended. Disconnecting during operation will stop machining and, under certain circumstances, cause damage to the workpiece.

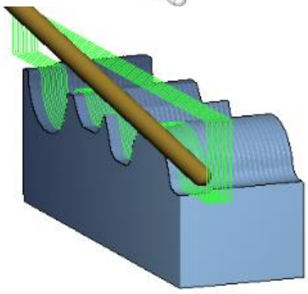
Notes

Milling with 3+2 or 5 axes ?

3+2 axis machining

The linear axes move during machining while the rotary axes only position and remain static during the machining process (3+1 or 3+2 axis machining is possible).





- **2D range:**

  Standard drilling, taping and contour milling operations in inclined planes with frame rotation.

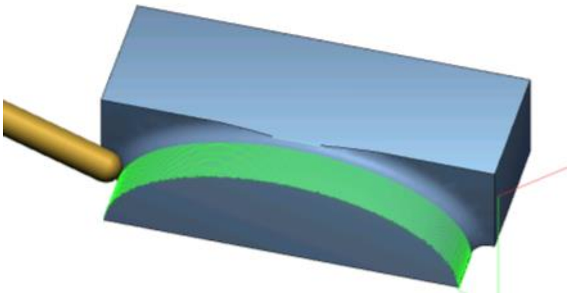
  Radius compensations with rotated workpiece frame can be applied in the NC-program with CUT2DF. (see example 1 page 45 - 53).

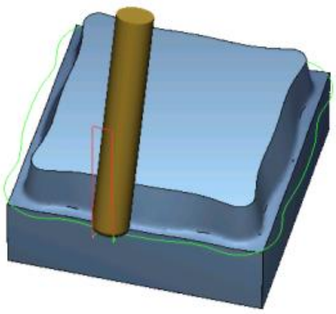
- **3D range:**

  Machining of free form surfaces with change in tool setting angle e.g. to avoid cutting in centre of tool (Surface speed = 0).

5-axis simultaneous machining

Milling of freeform surfaces in the mould and die sector with continuous change of tool orientation angle. Up to 5-axis can move simultaneous in order to trace a tool path, depending on the geometry of the surface and therefore to maintain opti-mum cut-ting con-ditions dur-ing ma-chining.





- **5-axis face milling:**

  The tool is engaged only with the face side of a symmetrical spherical ball mill or end mill with corner radius (bull nose cutter).

  Radius compensations for various tool types can be applied in the NC-program with CUT3DF. (see example 1 and 2 page 70 - 73).

- **5-axis circumferential milling:**

  The tool is engaged only with the side of a symmetrical tool (e.g. spherical ball mill, end mill, end mill with corner radius (bull nose cutter).

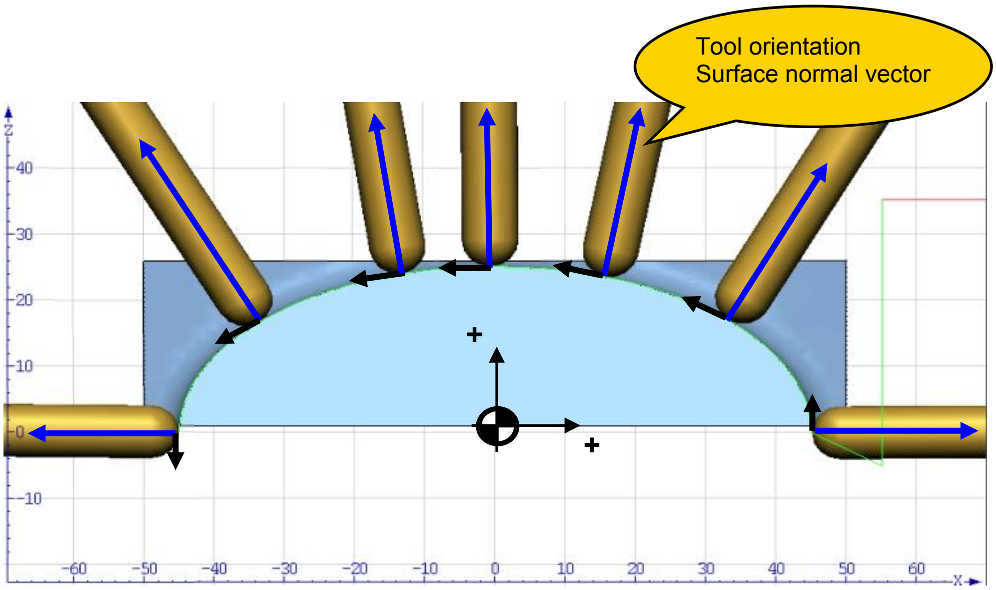
  Radius compensations for various tool types can be applied in the NC-program with CUT3DCC or CUT3DCCD (see example 3 page 74 - 75).

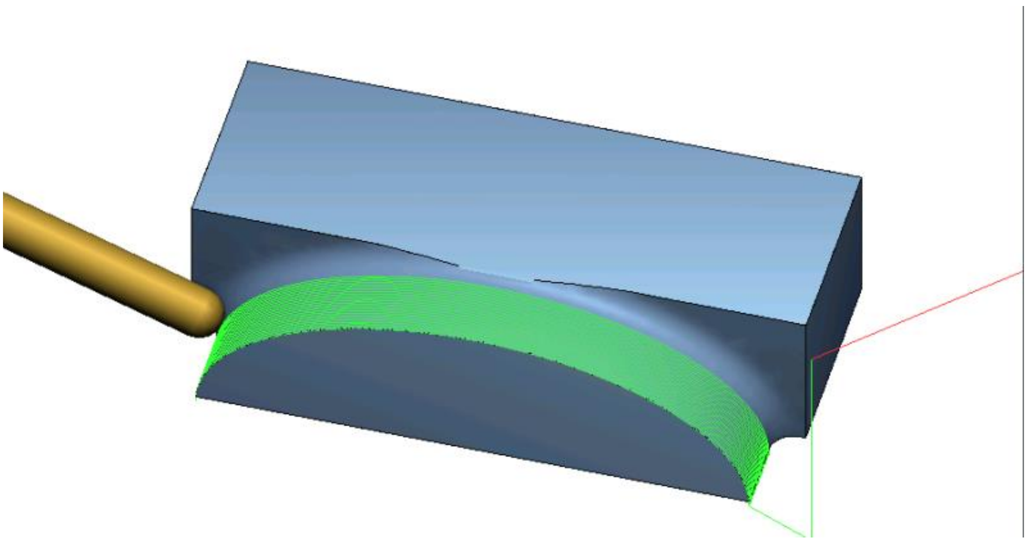
Notes

5-axis face milling

Example 1:

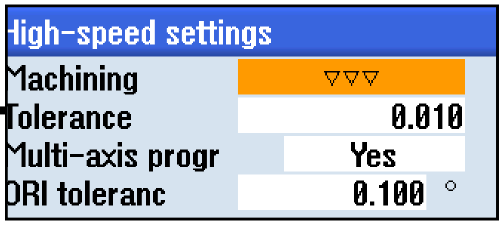
Programming of a ellipse (finishing with angle step FI=0.5°)
Programming tool path with the tool path tangent to the surface normal
Cartesian vector tool orientation A3=, B3=, C3=
Orientation Interpolation ORIAXES/ORIWKS





Notes

NC/WCS/CAD_CAM/ELLIPSE_1_A3

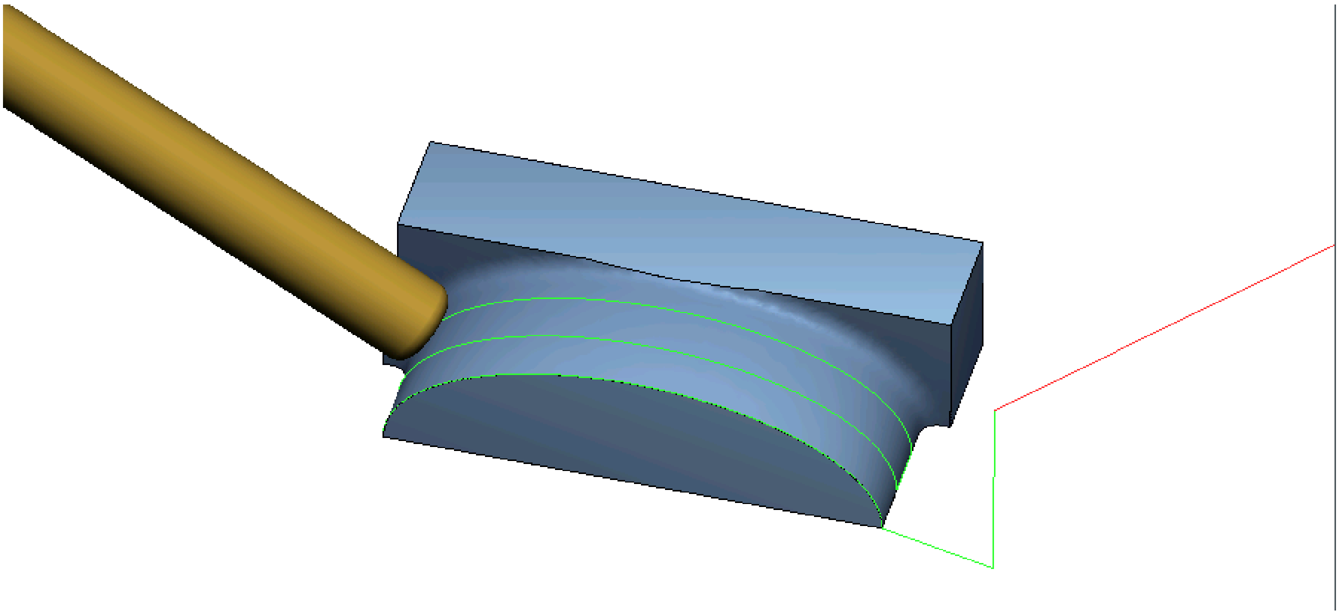


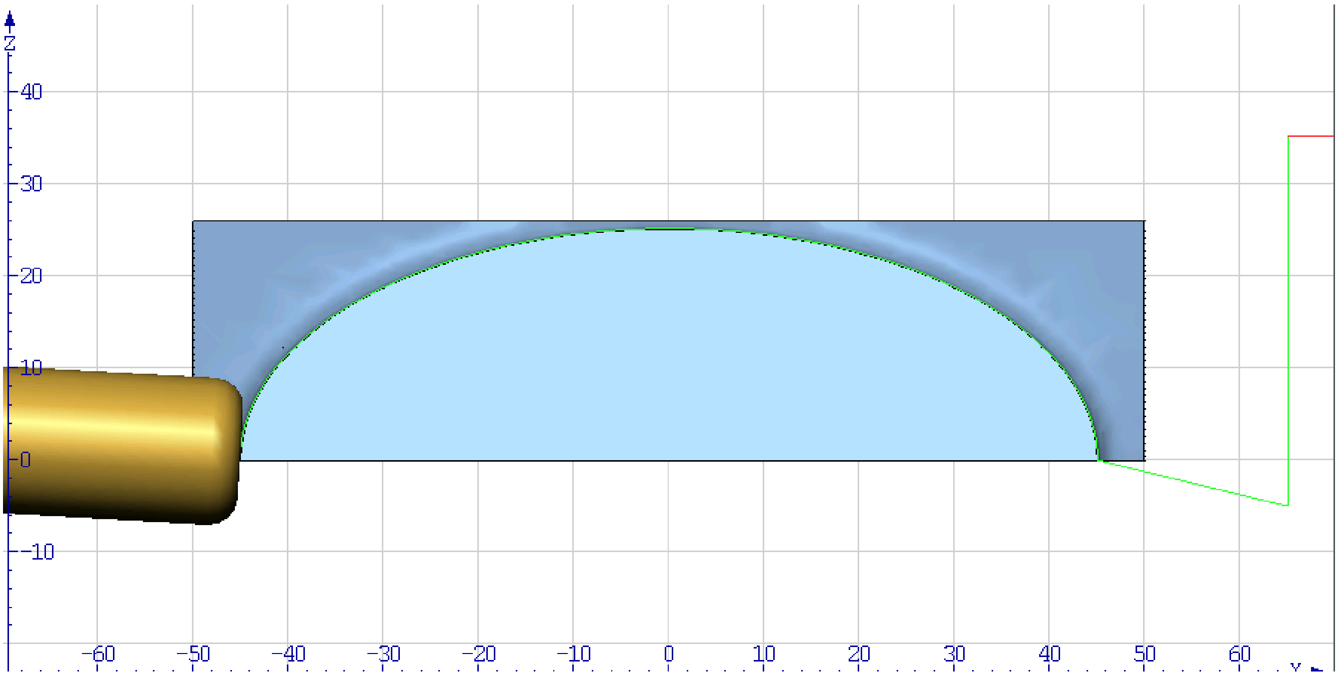

```
;***CARTESIAN VECTOR PROGRAMMING A3=,B3=,C3= WITH ORIAXES***
N100 CYCLE800(2,"TABLE",200000,57,0,0,0,0,0,0,0,0,0,1,,1)
N101 CYCLE800(2,"0", 200000, 57, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, , 1)
N102 G54 G17
N103 WORKPIECE(,,,"BOX",64,25,0,-50,-50,0,100,50)
N104 T="BALLMILL_D8"
N105 M6
N106 S8000 M3 F1000
N107 CYCLE832(0.01,_ORI_FINISH,0.1)
N108 TRAORI; Activate 5-axis transformation
N109 ORIWKS; Reference for tool orientation WCS
N110 ORIAXES; Rotary axes interpolation
N111 ORISON;  Orientation smoothing on
N112 G0 Z35.241 A3=0 B3=0 C3=1 ;Preposition in Z
N113 G0 X55.078 Y0 ;Preposition in X and Y
N114 G1 Z-5 F1000 ;Feed to start position in Z
N115 G1 X55.078 Z-5 A3=25.241 B3=0 C3=0 ;FI= 0
N115 G1 X45.078 Z0 A3=25.241 B3=0 C3=0 ;FI= 0
N116 G1 X45.076 Z0.22 A3=25.2400389 B3=0 C3=0.3933748 ;FI= .5
N117 G1 X45.071 Z0.441 A3=25.2371557 B3=0 C3=0.7867196 ;FI= 1
N118 G1 X45.063 Z0.661 A3=25.2323505 B3=0 C3=1.1800045 ;FI= 1.5
N119 G1 X45.051 Z0.881 A3=25.2256239 B3=0 C3=1.5731995 ;FI= 2
N120 G1 X45.035 Z1.101 A3=25.2169762 B3=0 C3=1.9662747 ;FI= 2.5
N121 G1 X45.016 Z1.321 A3=25.2064081 B3=0 C3=2.3592002 ;FI= 3
```

```
N122 G1 X44.994 Z1.541 A3=25.1939204 B3=0 C3=2.751946 ;FI= 3.5
N123 G1 X44.968 Z1.761 A3=25.1795142 B3=0 C3=3.1444823 ;FI= 4
N124 G1 X44.939 Z1.98 A3=25.1631904 B3=0 C3=3.5367791 ;FI= 4.5
N125 G1 X44.906 Z2.2 A3=25.1449504 B3=0 C3=3.9288065 ;FI= 5
N126 G1 X44.87 Z2.419 A3=25.1247954 B3=0 C3=4.3205348 ;FI= 5.5
N127 G1 X44.831 Z2.638 A3=25.1027272 B3=0 C3=4.711934 ;FI= 6
N128 G1 X44.788 Z2.857 A3=25.0787472 B3=0 C3=5.1029744 ;FI= 6.5
N129 G1 X44.742 Z3.076 A3=25.0528574 B3=0 C3=5.4936262 ;FI= 7
……
N828 G1 X44.906 Z2.2 A3=25.1449504 B3=0 C3=3.9288065 ;FI= 5
N829 G1 X44.939 Z1.98 A3=25.1631904 B3=0 C3=3.5367791 ;FI= 4.5
N830 G1 X44.968 Z1.761 A3=25.1795142 B3=0 C3=3.1444823 ;FI= 4
N831 G1 X44.994 Z1.541 A3=25.1939204 B3=0 C3=2.751946 ;FI= 3.5
N832 G1 X45.016 Z1.321 A3=25.2064081 B3=0 C3=2.3592002 ;FI= 3
N833 G1 X45.035 Z1.101 A3=25.2169762 B3=0 C3=1.9662747 ;FI= 2.5
N834 G1 X45.051 Z0.881 A3=25.2256239 B3=0 C3=1.5731995 ;FI= 2
N835 G1 X45.063 Z0.661 A3=25.2323505 B3=0 C3=1.1800045 ;FI= 1.5
N836 G1 X45.071 Z0.441 A3=25.2371557 B3=0 C3=0.7867196 ;FI= 1
N837 G1 X45.076 Z0.22 A3=25.2400389 B3=0 C3=0.3933748 ;FI= .5
N838 G1 X45.078 Z0 A3=25.241 B3=0 C3=0 ;FI= 0
N841 G1 X55.078 Z0 A3=25.241 B3=0 C3=0 ;FI= 0
N842 G1 X55.078 Z0 A3=0 B3=0 C3=1 ;FI= 0
N840 TRAFOOF; Switch off 5-axis transformation
N841 ORISOF;  Switch off orientation smoothing
N842 M30
```

Notes

Programming tool path with the tool path tangent to the surface normal is more suitable in this example for a toroid endmill (endmill with corner radius). The programmed tool path is in this case is the same, only the stepover can be much larger (e.g. 50% of cutter Ø or bigger).



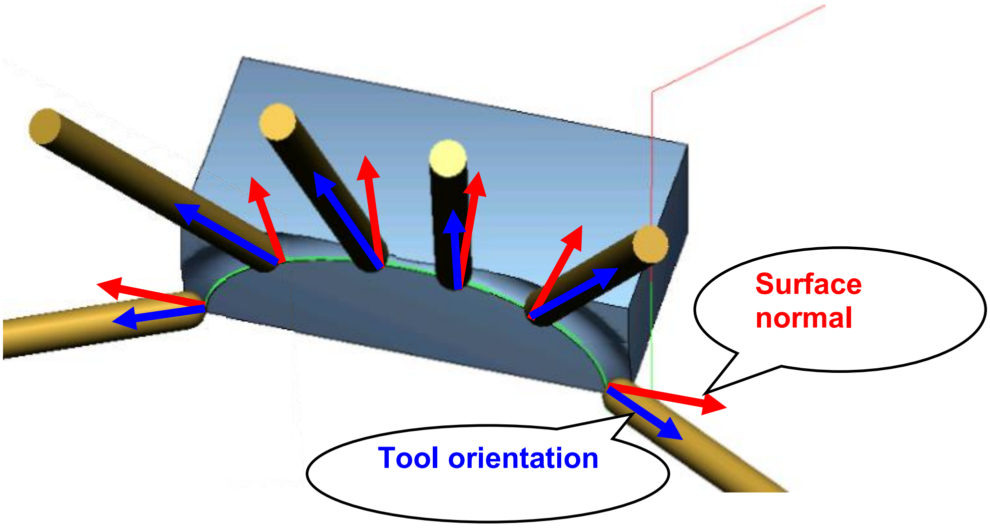


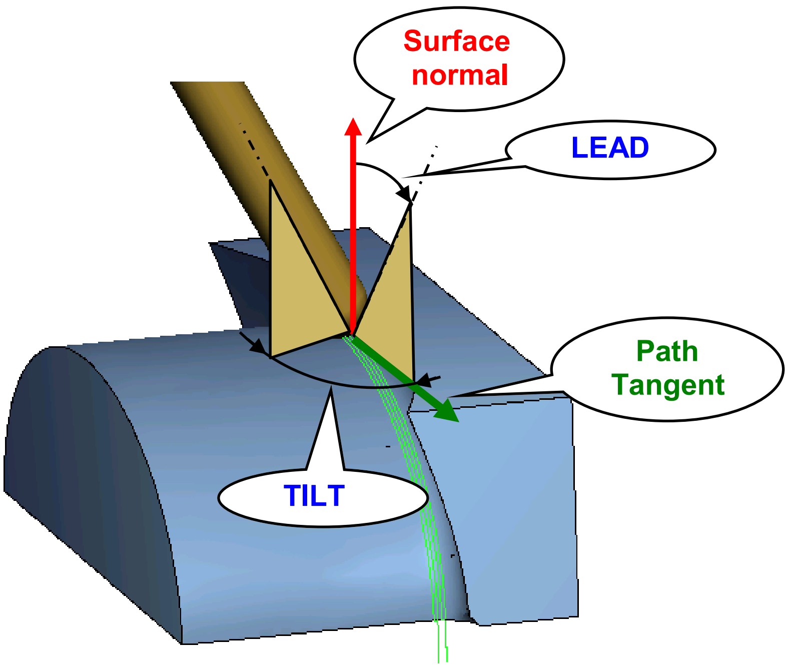
Notes

5-axis face milling

Example 2:

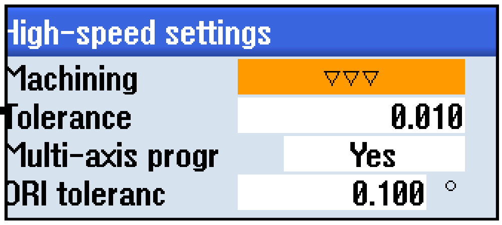
Programming of a ellipse (finishing with angle step FI=0.5°)
Tool path with constant tool setting angle to surface normal
Tool orientation with LEAD and TILT
Surface normal with A5=.. B5=.. C5=..
Orientation Interpolation ORIPATH/ORIWKS





Notes

NC/WKS/CAD_CAM/ELLIPSE_1_A3




```
;***CARTESIAN VECTOR PROGRAMMING A3=,B3=,C3= WITH ORIAXES***
N100 CYCLE800(2,"TABLE",200000,57,0,0,0,0,0,0,0,0,0,1,,1)
N101 CYCLE800(2,"0", 200000, 57, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, , 1)
N102 G54 G17
N103 WORKPIECE(,,,"BOX",64,25,0,-50,-50,0,100,50)
N104 T="BALLMILL_D8"
N105 M6
N106 S8000 M3 F1000
N107 CYCLE832(0.01,_ORI_FINISH,0.1)
N108 TRAORI; Activate 5-axis transformation
N109 ORIWKS; Reference for tool orientation WKS
N110 ORIAXES; Rotary axes interpolation
N111 ORISON;  Orientation smoothing on
N112 G0 Z35.241 A3=0 B3=0 C3=1 ;Preposition in Z
N113 G0 X55.078 Y0 ;Preposition in X and Y
N114 G1 Z-5 F1000 ;Feed to start position in Z
N115 G1 X55.078 Z-5 A3=25.241 B3=0 C3=0 ;FI= 0
N115 G1 X45.078 Z0 A3=25.241 B3=0 C3=0 ;FI= 0
N116 G1 X45.076 Z0.22 A3=25.2400389 B3=0 C3=0.3933748 ;FI= .5
N117 G1 X45.071 Z0.441 A3=25.2371557 B3=0 C3=0.7867196 ;FI= 1
N118 G1 X45.063 Z0.661 A3=25.2323505 B3=0 C3=1.1800045 ;FI= 1.5
N119 G1 X45.051 Z0.881 A3=25.2256239 B3=0 C3=1.5731995 ;FI= 2
N120 G1 X45.035 Z1.101 A3=25.2169762 B3=0 C3=1.9662747 ;FI= 2.5
N121 G1 X45.016 Z1.321 A3=25.2064081 B3=0 C3=2.3592002 ;FI= 3
N122 G1 X44.994 Z1.541 A3=25.1939204 B3=0 C3=2.751946 ;FI= 3.5
N123 G1 X44.968 Z1.761 A3=25.1795142 B3=0 C3=3.1444823 ;FI= 4
N124 G1 X44.939 Z1.98 A3=25.1631904 B3=0 C3=3.5367791 ;FI= 4.5
```

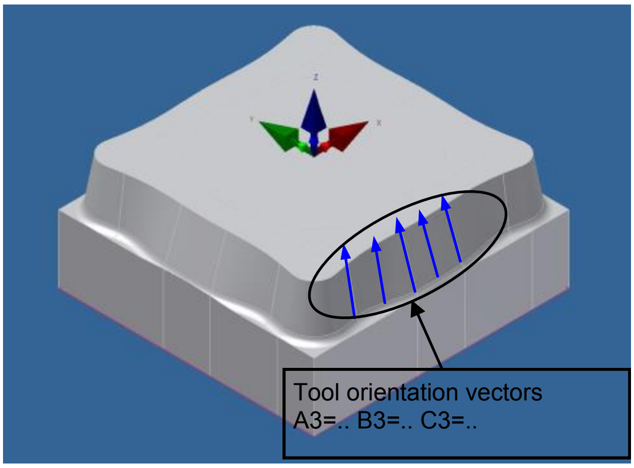
```
N125 G1 X44.906 Z2.2 A3=25.1449504 B3=0 C3=3.9288065 ;FI= 5
N126 G1 X44.87 Z2.419 A3=25.1247954 B3=0 C3=4.3205348 ;FI= 5.5
N127 G1 X44.831 Z2.638 A3=25.1027272 B3=0 C3=4.711934 ;FI= 6
N128 G1 X44.788 Z2.857 A3=25.0787472 B3=0 C3=5.1029744 ;FI= 6.5
N129 G1 X44.742 Z3.076 A3=25.0528574 B3=0 C3=5.4936262 ;FI= 7
……
N828 G1 X44.906 Z2.2 A3=25.1449504 B3=0 C3=3.9288065 ;FI= 5
N829 G1 X44.939 Z1.98 A3=25.1631904 B3=0 C3=3.5367791 ;FI= 4.5
N830 G1 X44.968 Z1.761 A3=25.1795142 B3=0 C3=3.1444823 ;FI= 4
N831 G1 X44.994 Z1.541 A3=25.1939204 B3=0 C3=2.751946 ;FI= 3.5
N832 G1 X45.016 Z1.321 A3=25.2064081 B3=0 C3=2.3592002 ;FI= 3
N833 G1 X45.035 Z1.101 A3=25.2169762 B3=0 C3=1.9662747 ;FI= 2.5
N834 G1 X45.051 Z0.881 A3=25.2256239 B3=0 C3=1.5731995 ;FI= 2
N835 G1 X45.063 Z0.661 A3=25.2323505 B3=0 C3=1.1800045 ;FI= 1.5
N836 G1 X45.071 Z0.441 A3=25.2371557 B3=0 C3=0.7867196 ;FI= 1
N837 G1 X45.076 Z0.22 A3=25.2400389 B3=0 C3=0.3933748 ;FI= .5
N838 G1 X45.078 Z0 A3=25.241 B3=0 C3=0 ;FI= 0
N841 G1 X55.078 Z0 A3=25.241 B3=0 C3=0 ;FI= 0
N842 G1 X55.078 Z0 A3=0 B3=0 C3=1 ;FI= 0
N839 G1 X55.078 Z50 A3=0 B3=0 C3=1 ;FI= 0
N840 TRAFOOF
N841 M30
```

Notes

5-axis circumferential milling

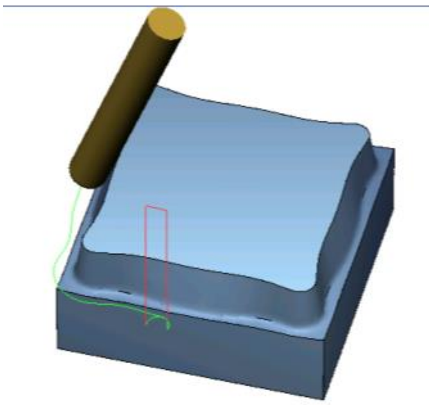
Example 3: Programming a cavity

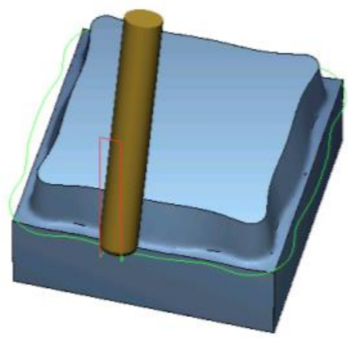
Cartesian vector tool orientation A3=, B3=, C3=
Orientation Interpolation ORIAXES/ORIWKS
3D Tool radius compensation CUT3DCCD

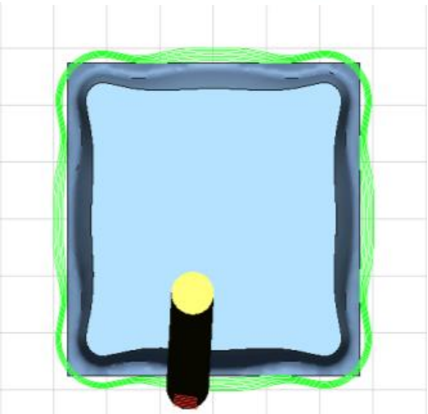


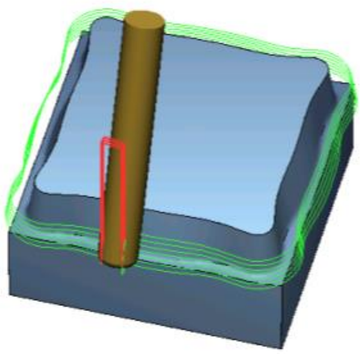
Programming notes:

The tool path is programmed to centre of the tool in the CAM. It is possible to use 3D tool radius compensation for circumferential milling. In case of **CUT3DCCD** tool radius compensation is computed by the NC as differential radius from the tool centre line between the standard tool and the real tool.
(For a detailed description see module M105 "3D-**T**ool **R**adius **C**ompensation").





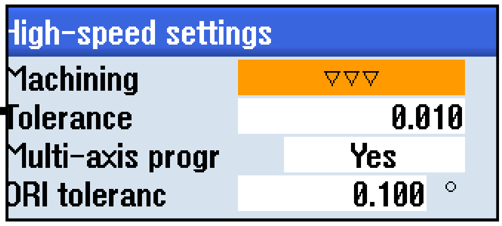




Notes

NC/WCS/CAD_CAM/KAVITY_5_AXIS

N10 CYCLE800(1,"TABLE",200000,57,0,0,0,0,0,0,0,0,0,1,100,1)
N11 CYCLE800(5,"0",100000,57,0,0,0,0,0,0,0,0,0,-1,100,1)
N12 G54 G17
N13 WORKPIECE(,,"","BOX",112,0,-50,0,-55,-55,55,55)
N14 T="TOROID_D16_R3"
N15 M6
N16 S14500 M03
N17 **CYCLE832(0.01,_ORI_FINISH0.1)**
N18 **TRAORI**; Switch on 5-axis transformation
N19 **ORISON**; Orientation reference is the WCS
N20 **ORIWKS**; Rotary axis interpolation
N21 **ORIAXES**; Switch onorientation smoothing
N22 **CUT3DCCD**; 3D tool radius compensation circumferential milling
N23 G54 D1
N24 G1 X0 Y-66 A3=0 B3=0 C3=1 F2500
N25 G1 **G41** X-5.945 Y-60.546 A3=0.021144 B3=0.341366 C3=0.939693 F2500
N26 Z5.502 A3=0.021144 B3=0.341366 C3=0.939693
N27 G1 Z0.502 A3=0.021144 B3=0.341366 C3=0.939693 F2500
N28 G1 X-5.743 Y-59.698 A3=0.021144 B3=0.341366 C3=0.939693 F500
N29 G1 X-5.691 Y-58.828 A3=0.021144 B3=0.341366 C3=0.939693
N30 G1 X-5.792 Y-57.963 A3=0.021144 B3=0.341366 C3=0.939693
N31 G1 X-6.041 Y-57.127 A3=0.021144 B3=0.341366 C3=0.939693
N32 G1 X-6.431 Y-56.348 A3=0.021144 B3=0.341366 C3=0.939693
N33 G1 X-6.951 Y-55.648 A3=0.021144 B3=0.341366 C3=0.939693
N34G1 X-7.584 Y-55.05 A3=0.021144 B3=0.341366 C3=0.939693
N35 G1 X-8.312 Y-54.57 A3=0.021144 B3=0.341366 C3=0.939693
N36 G1 X-9.112 Y-54.224 A3=0.021144 B3=0.341366 C3=0.939693
N37 G1 X-9.959 Y-54.022 A3=0.021144 B3=0.341366 C3=0.939693



N38 G1 X-10.445 Y-53.993 A3=0.020068 B3=0.341361 C3=0.939718
N39 G1 X-10.929 Y-53.965 Z0.499 A3=0.018938 B3=0.340982 C3=0.939879
N40 G1 X-11.412 Y-53.938 Z0.494 A3=0.017764 B3=0.340244 C3=0.94017
N41 G1 X-11.894 Y-53.913 Z0.487 A3=0.016557 B3=0.339162 C3=0.940582
N42 G1 X-12.374 Y-53.888 Z0.479 A3=0.015325 B3=0.33775 C3=0.941111
N43 G1 X-12.852 Y-53.865 Z0.468 A3=0.01408 B3=0.336025 C3=0.941748
N44 G1 X-13.329 Y-53.843 Z0.455 A3=0.012831 B3=0.333999 C3=0.942486
N45 G1 X-13.804 Y-53.822 Z0.441 A3=0.011589 B3=0.331686 C3=0.943319
……..
N17510 G1 X-14.128 Y-61.883 A3=0.021144 B3=0.341366 C3=0.939693
N17511 G1 X-14.609 Y-62.61 A3=0.021144 B3=0.341366 C3=0.939693
N17512 G1 X-14.957 Y-63.409 A3=0.021144 B3=0.341366 C3=0.939693
N17513 G1 X-15.16 Y-64.257 A3=0.021144 B3=0.341366 C3=0.939693
N17514 G1 X-15.213 Y-65.127 A3=0.021144 B3=0.341366 C3=0.939693
N17515 G1 X-15.114 Y-65.993 A3=0.021144 B3=0.341366 C3=0.939693
N17516 G1 **G40** X0 Y-66 A3=0 B3=0 C3=1
N17516 G0 Z50 A3=0.021144 B3=0.341366 C3=0.939693
N17517 **TRAFOOF ;**Switch off 5-axis transformation
N17518 **ORISOF ;** Switch off orientation smoothing
N17519 M30

Notes

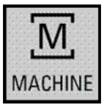
Information for the machine operator

Tool retraction with TOROT

Explanation of the function:

When 5-axis transformation is enabled, TOROT generates a frame whose Z-axis coincides with the current tool orientation. This allows the tool to be retracted in a 5-axis program, after a tool breakage, for example, without the risk of collision, by retracting the Z-axis. After tool orientation has been programmed with TOROT, all the programmed geometry axis movements refer to the frame (Tool reference) generated by this programming.

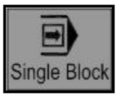
Programming of TOROT in MDA mode:



Select the “Machine” operating area.

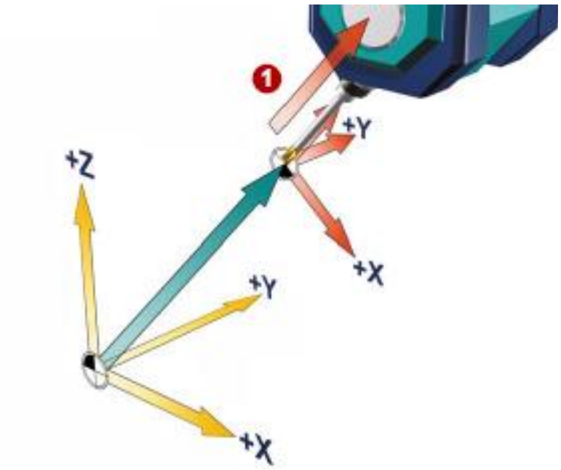
Select “MDA” mode. Enter the program as follows:

```
N10 TRAORI              ;5-axis transformation ON
N20 TOROT               ;Calculate and select retraction frame
N30 G1 G91 Z50 F500     ;incr. retract in Z direction by 50 mm
N40 TOROTOF
N50 M17                 ;End of subroutine
```



Select single block.

Start the program block by block with **“Cycle Start”**



With TOROT a FRAME is being generated, with the current tool orientation parallel to the Z-axis direction 1.

As an alternative to incremental retraction in MDA mode, the tool can be retracted in JOG mode by pressing the direction key in the tool direction.

Note:

Notice: For retraction in JOG mode the machine must be configured accordingly (Z-axis as the geometry axis). TOROT must be deselected before the start of the next program with **TOROTOF**

Notes

Programmable tool correction offset TOFFL, TOFF, ...

Explanation of the function:

The user can use the commands TOFFL/TOFF and TOFFR to modify the effective tool length or the effective tool radius in the NC program, without changing the tool offset data stored in the compensation memory.

These programmed offsets are deleted again at the end of the program.

Tool length offset (TOFFL and TOFF)

Depending on the type of programming, programmed tool length offsets are assigned either to the tool length components L1, L2 and L3 (TOFFL) stored in the compensation memory or to the geometry axes (TOFF). The programmed offsets are treated accordingly for a plane
change (G17/G18/G19 ↔ G17/G18/G19):

If the offset values are assigned to the tool length components, the directions in which the programmed offsets apply, are replaced accordingly.

If the offset values are assigned to the geometry axes, a plane change does not effect the assignment in relation to the coordinate axes.

Tool radius offset (TOFFR)

The command TOFFR is available for the programming of a tool radius offset.

Syntax:

Tool length-offset:
TOFFL=<value>
TOFFL[1]=<value>
TOFFL[2]=<value>
TOFFL[3]=<value>
TOFF[<geometry axis>]=<value>

Tool radius-offset:
TOFFR=<value>

Notes

Description of the NC commands:

Command	Description
TOFFL	Command for the compensation of the effective tool length TOFFL can be programmed with or without index: • Without index: TOFFL= The programmed offset value is applied in the same direction as the tool length component L1 stored in the compensation memory. • With index: TOFFL[1]=, TOFFL[2]= or TOFFL[3]= . The programmed offset value is applied in the same direction as the tool length component L1, L2 or L3 stored in the offset memory. The commands TOFFL and TOFFL[1] have an identical effect.
TOFF	Command for the compensation of the tool length in the component parallel to the specified geometry axis TOFF is applied in the direction of the tool length component, which is effective with non-rotated tool (orientable tool holder or orientation transformation) parallel to the <geometry axis>* specified in the index.
TOFFR	Command for the compensation of the effective tool radius TOFFR changes the effective tool radius with active tool radius compensation by the programmed offset value. <value>: Offset value for the tool length or radius Type: REAL

Note for TOFFL:

How these tool length offset values are calculated in the axes is determined by the tool type and the current working plane (G17/G18/G19).

Note for TOFF:

A frame does not influence the assignment of the programmed values to the tool length components, i.e. the workpiece coordinate system (WCS) is not used for the assignment of the tool length components to the geometry axes, but the tool in the basic tool position.

Note for TOFFR:

The TOFFR command has almost the same effect as the OFFN command (see "Tool radius compensation (G40, G41, G42, OFFN). There is only a difference with active peripheral curve transformation (TRACYL) and active slot side compensation. In this case, the tool radius is affected by OFFN with a negative sign, but by TOFFR with a positive sign.
OFFN and TOFFR can be effective simultaneously. They then generally have an additive effect (except for slot side compensation).

Notes

Further syntax rules:

- The tool length can be changed simultaneously in all three components. However, commands of the TOFFL/TOFFL [1..3] group and commands of the TOFF[<geometry axis>] may not be used simultaneously in one block. TOFFL and TOFFL[1] may also not be written simultaneously in one block.

- If all three tool length components are not programmed in a block, the components not programmed remain unchanged. In this way, it is possible to build up offsets for several components block-by-block. However, this only applies as long as the tool components have been modified either only with TOFFL or only with TOFF. Changing the programming type from TOFFL to TOFF or vice versa deletes any previously programmed tool length offsets (see example 3).

Supplementary conditions:

- Evaluation of setting data: The following setting data is evaluated when assigning the programmed offset values to the tool length components:

  SD42940 $SC_TOOL_LENGTH_CONST (change of tool length on change of planes).
  SD42950 $SC_TOOL_LENGTH_TYPE (assignment of the tool length independent of tool type)

  If this setting data has valid values not equal to 0, then these take preference over the contents of G code group 6 (plane selection G17 - G19) or the tool type ($TC_DP1[<T no.>, <D no.>]) contained in the tool data, i.e. this setting data influences the evaluation of the offsets in the same way as the tool length components L1 to L3.

- Tool change: All offset values are retained during a tool change (cutting edge change), e.g. they are also effective for the new tool (new cutting edge).

Examples:

Example 1: Positive tool length offset

The active tool is a drill with length L1 = 100 mm.
The active plane is G17, i.e. the drill points in the Z direction.
The effective drill length is to be increased by 1 mm.

The following variants are available for programming the tool length offset:

TOFFL=1
or
TOFFL[1]=1
or
TOFF[Z]=1

Notes

Example 2: Negative tool length offset

The active tool is a drill with length L1 = 100 mm.
The active plane is G18, i.e. the drill points in the Y direction.
The effective drill length is to be decreased by 1 mm. The following variants are available for the programming of this tool length offset:
TOFFL=-1
or
TOFFL[1]=-1
or
TOFF[Y]=1

Example 3: Changing the programming type from TOFFL to TOFF

The active tool is a milling tool. The active plane is G17.

N10 TOFFL[1]=3 TOFFL[3]=5 ; Effective offsets: L1=3, L2=0, L3=5
N20 TOFFL[2]=4 ; Effective offsets: L1=3, L2=4, L3=5
N30 TOFF[Z]=1.3 ; Effective offsets: L1=0, L2=0, L3=1.3

Example 4: Plane change

In this example, the offset of 1 mm in the Z-axis is retained when changing to G18 in block N60; the effective tool length in the Y axis is the unchanged tool length of 100 mm.
However, in block N100, the offset is effective in the Y axis when changing to G18 as it was assigned to tool length L1 in the programming and this length component is effective in the Y axis with G18.

N10 $TC_DP1[1,1]=120
N20 $TC_DP3[1,1]= 100 ; Tool change L1=100mm
N30 T1 D1 G17
N40 TOFF[Z]=1.0 ; Offset in Z direction (corresponds to L1 for G17).
N50 G0 X0 Y0 Z0 ; Machine axis position X0 Y0 Z101
N60 G18 G0 X0 Y0 Z0 ; Machine axis position X0 Y100 Z1
N70 G17
N80 TOFFL=1.0 ; Offset in L1 direction (corresponds to Z for G17).
N90 G0 X0 Y0 Z0 ; Machine axis position X0 Y0 Z101.
N100 G18 G0 X0 Y0 Z0

Further information

Applications

The "Programmable tool offset" function is especially interesting for ball mills and milling tools with corner radii as these are often calculated in the CAM-system to the ball center instead of the ball tip. However, generally the tool tip is measured when measuring the tool and stored as tool length in the compensation memory.

Notes

Setup 5-axis transformation TRAORI

Configuration of machine data

Prerequisites:

5-axis transformation TRAORI is an option and must be licensed.

A set of transformation data for the 5-axis transformation **TRAORI** must be set up in the machine data "**$MC_TRAFO_ ....**" .

Parameterization of orientable tool carrier data

Machine types for which the table or tool can be rotated, can either be operated as true 5-axis machines or as conventional machines with orientable tool carriers. In both cases, machine kinematics is determined by the same data, which, due to different parameters, previously had to be entered twice - for tool holder via system variables and for transformations via machine data. The new transformation type 72 can be used to specify that these two machine types access identical data.

Transformation type 72

The following machine data can be used to define a generic 5-axis transformation for transformation type 72 with kinematic data read from the data for an orientable tool holder:

MD24100 $MC_TRAFO_TYPE_1 = 72 (definition of 1st transformation)
MD24200 $MC_TRAFO_TYPE_2 = 72 (definition of 2nd transformation)

MD24300 $MC_TRAFO_TYPE_3 = 72 (definition of 3rd transformation)
MD24400 $MC_TRAFO_TYPE_4 = 72 (definition of 4th transformation)

In den folgenden Maschinendaten kann der mit Typ 72 definierten 5-Achstransformation im Kanal die Nummer des entsprechenden Schwenkdatensatzes zugeordnet werden (maximal 4):

MD24582 $MC_TRAFO5_TCARR_NO_1 = 1 (e.g. TCARR=1 for the 1st 5-axis transformation) or
MD24682 $MC_TRAFO5_TCARR_NO_2 = 2 (e.g. TCARR=2 for the 2nd 5-axis transformation) or
MD25282 $MC_TRAFO5_TCARR_NO_3 = 3 (e.g. TCARR=3 for the 3rd 5-axis transformation) or
MD25382 $MC_TRAFO5_TCARR_NO_4 = 4 (e.g. TCARR=4 for the 4th 5-axis transformation).

The corresponding transformation type can then be derived from the content of the kinematic type with parameter

Machine type	**1**	**2**	**3**	**4**
Swivel-/rotary table:	Tool	Workpiece	Tool/workpiece	Orientable tool holder TCARR
Kinematic Type:	T	P	M	T,P,M
Transformation type:	24	40	56	72 from content of $TC_CARR23

$TC_CARR23 (see following table).

Note:

It is possible to define up to 20 Transformation types per channel, from which a maximum of 4 can be 5-axis transformations.

Notes

Axis assignment

MD 24110[0] - [19] $MC_TRAFO_AXES_IN_1 (Channel axis assignment transformation 1)
MD 24210[0] - [19] $MC_TRAFO_AXES_IN_2 (Channel axis assignment transformation 2)
......

The axis assignment at the start of the 5-axis transformation defines the axis that will be mapped by the transformation internally onto a channel axis [n]. Thus, the following is defined in the machine data below:

MD24110[0] $MC_TRAFO_AXES_IN_1 (Channel axis 1 of transformation 1)
MD24110[1] $MC_TRAFO_AXES_IN_1 (Channel axis 2 of transformation 1)
MD24110[2] $MC_TRAFO_AXES_IN_1 (Channel axis 3 of transformation 1)
MD24110[3] $MC_TRAFO_AXES_IN_1 (Channel axis 4 of transformation 1)
MD24110[4] $MC_TRAFO_AXES_IN_1 (Channel axis 5 of transformation 1)
......

MD 24120[0] - [2] $MC_TRAFO_GEOAX_ASSIGN_TAB_1 (Geometry axes assignment trafo 1)
MD 24220[0] - [2] $MC_TRAFO_GEOAX_ASSIGN_TAB_2 (Geometry axes assignment trafo 2)
......

This MD states the geometry axes on which the axes **[n]** of the Cartesian coordinate system are mapped for the active transformation 1 - 10.
See following example:

MD 24120[0] $MC_TRAFO_GEOAX_ASSIGN_TAB_1 (Geometry axis 1 of transformation 1)
MD 24120[1] $MC_TRAFO_GEOAX_ASSIGN_TAB_1 (Geometry axis 2 of transformation 1)
MD 24120[2] $MC_TRAFO_GEOAX_ASSIGN_TAB_1 (Geometry axis 3 of transformation 1)

Corresponds with:
MD20050[0] - [2] $MC_AXCONF_GEOAX_ASSIGN_TAB wenn keine Transformation aktiv ist.

Settings the tool reference point with active 5-axis transformation

MD 24130 $MC_TRAFO_INCLUDES_TOOL_1 (Tool handling with 1st active transformation)
MD 24230 $MC_TRAFO_INCLUDES_TOOL_2 (Tool handling with 2nd active transformation)
.....

This machine data states for each channel weather the tool is included in the transformation or not.

It is evaluated on the condition that the orientation of the tool with reference to the basic coordinate system can not be changed by the transformation. In standard transformation, only the “inclined-axis transformation” fulfils this condition.

If this MD is set, the basic coordinate system (BCS) refers to the tool reference point even with active transformation. Otherwise it refers to the tool tip point (TTP).

Notes

Assignment of Rotary axis direction

MD24520[0] - [1] $MC_TRAFO5_ROT_SIGN_IS_PLUS_1
(Sign of rotary axis 1/2 for the 2nd 5-axis transformation)
MD24620[0] - [1] $MC_TRAFO5_ROT_SIGN_IS_PLUS_1
(Sign of rotary axis 1/2 for the 1st 5-axis transformation)

This machine data designates the sign with which the two rotary axes are included in the first 5-axis transformation of a channel.

MD = 0 (FALSE):
Sign is reversed.

MD = 1 (TRUE) :
Sign is not reversed and the traversing direction is defined according to
MD32100 $MA_AX_MOTION_DIR.

This machine data does not mean that the rotational direction of the rotary axis concerned is to be reversed, but specifies whether its motion is in the mathematically positive or negative direction
when the axis is moving in the positive direction.

The result of a change to this machine data is not therefore a change in the rotational direction, but a change in the compensatory motion of the linear axes.

However, if a directional vector and thus, implicitly, a compensatory motion is specified, the result is a change in the rotational direction of the rotary axis concerned.

On a real machine, therefore, the machine data may be set to FALSE (or zero) only if the rotary axis is turning in an anti-clockwise direction when moving in a positive direction.

Setting the kinematics initial orientation

Basic tool orientation vector

MD24574 $MC_TRAFO5_BASE_ORIENT_1 [0..2] (Tool orient. vector channel axis 1st 5-axis trafo)
MD24674 $MC_TRAFO5_BASE_ORIENT_2 [0..2] (Tool orient. vector channel axis 2nd 5-axis trafo)
.....

This MD indicates the vector of the base tool orientation (initial kinematic setting) in the general 5-axis transformation (TRAFO_TYPE_ = 24, 40, 56, 72) if this is not defined on the transformation call (G17, G18, G19) or read from a programmed tool in the program.

MD24574[0] $MC_TRAFO5_BASE_ORIENT_1 (tool base orientation vector X)
MD24574[1] $MC_TRAFO5_BASE_ORIENT_1 (tool base orientation vector Y)
MD24574[2] $MC_TRAFO5_BASE_ORIENT_1 (tool base orientation vector Z)

MD24580 $MC_TRAFO5_TOOL_VECTOR_1 (direction of orientation vector for 1st 5-axis trafo)

This machine data indicates the direction of the orientation vector for the first 5-axis transformation for each channel
0: Tool vector in X direction
1: Tool vector in Y direction
2: Tool vector in Z direction (default)

Notes

Setting of the tool orientation normal vector

MD24576[0]-[2] $MC_TRAFO6_BASE_ORIENT_NORMAL_1 (tool normal vector 1st 5-axis trafo)
MD24676[0]-[2] $MC_TRAFO6_BASE_ORIENT_NORMAL_2 (tool normal vector 2nd 5-axis trafo)
.....

The initial setting of the orientation normal vector for angular head attachments in the orientation transformation with TRAO-RI can be defined in one of the following three ways:

1. Vector components are programmed with TRAORI (<n>, <X>, <Y>, <Z>) and transferred as parameters 2 to 4 :
   Parameter 1: Transformation No. (<n>)
   Parameter 2 - 4: Orientation normal vector (<X>, <Y>, <Z>),

2. If no orientation normal vector has been programmed and a tool is active, the vector is taken from the tool parameters in the tool list (tool types 130,131).

3. If no orientation normal vector has been programmed and also no tool is active, the vector defined in the following machine data is used.
   MD24576[0] $MC_TRAFO6_BASE_ORIENT_1 (tool normal vector X)
   MD24576[1] $MC_TRAFO6_BASE_ORIENT_1 (tool normal vector Y)
   MD24576[2] $MC_TRAFO6_BASE_ORIENT_1 (tool normal vector Z)

The position of the orientation coordinate system of a standard tool depends on the active plane G17, G18, G19 according to the following table:

Table 1- 5 Position of the orientation coordinate system

	G17	**G18**	**G19**
Direction of tool orientation vector	Z	Y	X
Direction of orientation normal vector	Y	X	Z

Note:

This machine data is only important when working with angle head attachments in combination with 5-axis transformation, whereby the angle head has a fixed angle set parallel to one of the work planes (G17,G18,G19) or can change the fixed tool setting angle.

Notes

Orientation movements with axis limits

MD21180 $MC_ROT_AX_SWL_CHECK_MODE
(check software limits for orientation axes)

Calculate rotary axis position

If the tool orientation with active 5-axis transformation is programmed kinematic independent in an NC block by means of a Euler-, RPY-angle or direction vector, it is necessary to calculate the rotary axis positions that produce the desired orientation.

This calculation has no unique result and there are always at least two essentially different solutions. In addition, any number of solutions can result from a modification to the rotary axis positions by any multiple of 360 degrees.

The control system chooses the solution which represents the shortest distance from the current starting point, allowing for the programmed interpolation type.

Determining permissible axis limits

The control system attempts to define another permissible solution if the axis limits are violated, by approaching the desired axis position along the shortest path. The second solution is then verified, and if this solution also violates the axis limits, the axis positions for both solutions are modified by multiples of 360 until a valid position is found.

The following conditions must be met in order to monitor the axis limits of a rotary axis and modify the calculated end positions:

- A generic 5-axis transformation of type 24, 40 56 or 72 must be active.
- The axis must be referenced.
- The axis must not be a modulo rotary axis (MD30310).
- The following machine data may not be equal to zero:
  MD21180 $MC_ROT_AX_SWL_CHECK_MODE (check software limits
  for orientation axes)

The following machine data specifies the conditions under which the rotary axis positions may be modified: ***MD21180 $MC_ROT_AX_SWL_CHECK_MODE***

Value 0: No modification permitted (default, equivalent to previous behaviour).

Value 1: Modification is only permitted if axis interpolation is active (ORIAXES or ORIMKS).

Value 2: Modification is always permitted, even if vector interpolation (large circle interpolation, conical interpolation, etc.) was active originally.

Example for the modification of rotary axis motion

The machine is of transformation type 40 with a AC-swivel rotary table kinematic.
The first rotary axis is parallel to X (A-axis) and has a traversing range from –10 to+120°.
The second rotary axis is a modulo axis parallel to Z (C-axis).

Notes

To allow modification at any time, following machine data has the value 2:
MD21180 $MC_ROT_AX_SWL_CHECK_MODE = 2
(check software limits for orientation axes)

```
N10 X0 Y0 Z0 A0 C0
N20 TRAORI                    ; basic orientation 5-axis transformation
N30 A-1 C10                   ; Rotary axis positions A-1 and C10
N40 A3=-1 C3=1 ORIWKS         ; large circle interpolation in WCS
N50 M30
```

At the start of block N30 in the example program, the machine is positioned at rotary axis positions A-1 C10. The programmed end orientation can be achieved with either of the axis positions A-45 C0 (1st solution) or A45 C180 (2nd solution).

The first solution is selected initially, because it is nearest to the starting orientation and, unlike the second solution, can be achieved using large circle interpolation (ORIVECT). However, this position cannot be reached because of the axis limits of the A axis.

The second solution is therefore used instead, i.e. the end position is A45 C180. The end orientation is achieved by axis interpolation. The programmed orientation path cannot be followed.

Settings for specification of the rotary axes

MD30310 $MA_ROT_IS_MODULO (Axis specific modulo conversion)

1: A modulo conversion is performed on the setpoints for the rotary axes. The software limit switches and work area limitations are inactive. The traversing range is therefore unlimited
(MD30300 $MA_IS_ROT_AX must be set to “1”)

0: No modulo conversion

MD30320 $MA_DISPLAY_MODULO (Axis specific modulo pos. display)

1: “Modulo 360 degrees” position display is active
In case of a positive direction of rotation, the control resets the position display internally to 0.0000 degrees after one full revolution of the specified axis. The display range is always positive and lies between 0 and 359.999 degrees.

0: Absolute position display is active
Rotary axes positions are displayed as absolute positions (endless mode). E.g. position display +720 degrees after two revolutions of the specified axis.

MD30330 $MA_MODULO_RANGE: (Axis specific)
Defines the size of Modulo range (max. range of specified rot. Axis).

MD30340 $MA_MODULO_RANGE_START: (Axis specific)
Defines the start position of the modulo range
Start at 0 degrees = modulo range 360 = 360 degree
Start at 180 degrees = modulo range 180 = 540 degree
Start at –180 degrees = modulo range –180 = 180 degree

Notes

<u>Settings for work offsets with active 5-axis transformation</u>

MD10602 $MN_FRAME_GEOAX_CHANGE_MODE (Frames when changing geometry axes)

With this machine data it is possible to set the reset behaviour of the current work offset upon activation of 5-axis transformation.
If it is not clear how to set this machine data then it is recommended to program a work offset (G54)and tool offset number (D1) after the TRAORI command (recommended setting =1).

Settings for MD10602
0= The current total frame is cancelled when geometry axes are switched over.
1= The current total frame remains active when geometry axes are switched over.
2= The current total frame remains active. If rotations or rotary axes translations are active before geometry axes are switched over, then the switchover is aborted with a alarm.
3= The current total frame is deleted when selecting the 5-axis transformation. In case of programming the command GEOX() the frame is not cancelled.

<u>Settings for rotary axes offsets with active 5-axis transformation</u>

MD 24510 $MC_TRAFO5_ROT_AX_OFFSET_1 [0..2] (Rotary axes offset 1st 5-axis trafo)
MD 24610 $MC_TRAFO5_ROT_AX_OFFSET_2 [0..2] (Rotary axes offset 2nd 5-axis trafo)
....

MD24510[0] $MC_TRAFO5_ROT_AX_OFFSET_1 (Rotary axis offset for 1st rotary axis)
MD24510[1] $MC_TRAFO5_ROT_AX_OFFSET_1 (Rotary axis offset for 2nd rotary axis)
MD24510[2] $MC_TRAFO5_ROT_AX_OFFSET_1 (Rotary axis offset for 3rd rotary axis)

This machine data is evaluated with the use of the G-code command **ORIRESET** and allows setting of a rotary axis offset in degrees from MCS for the 1/2/3 rotary axis in the above machine data.

MD24590 $MC_TRAFO5_ROT_OFFSET_FROM_FR_1 (rotary axes offset for WO)

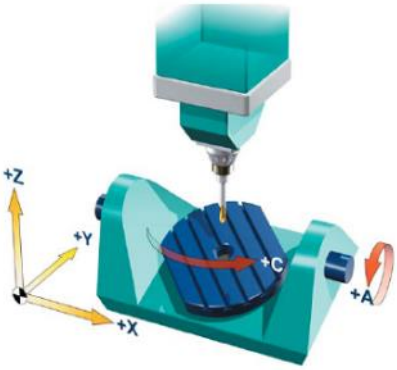
This machine data is evaluated with the use of the G-code command ORIWKS and allows setting of a rotary axis offset in degrees from MCS for the 1/2/3 rotary axis in a storable work offset (e.g. G54).

1: A rotary axes offset in the storable work offset is automatically accepted with active 5-axis trafo.
0: A rotary axes offset in the storable work offset is not accepted with active 5-axis trafo.

Notes

Setup example with kinematics type 72

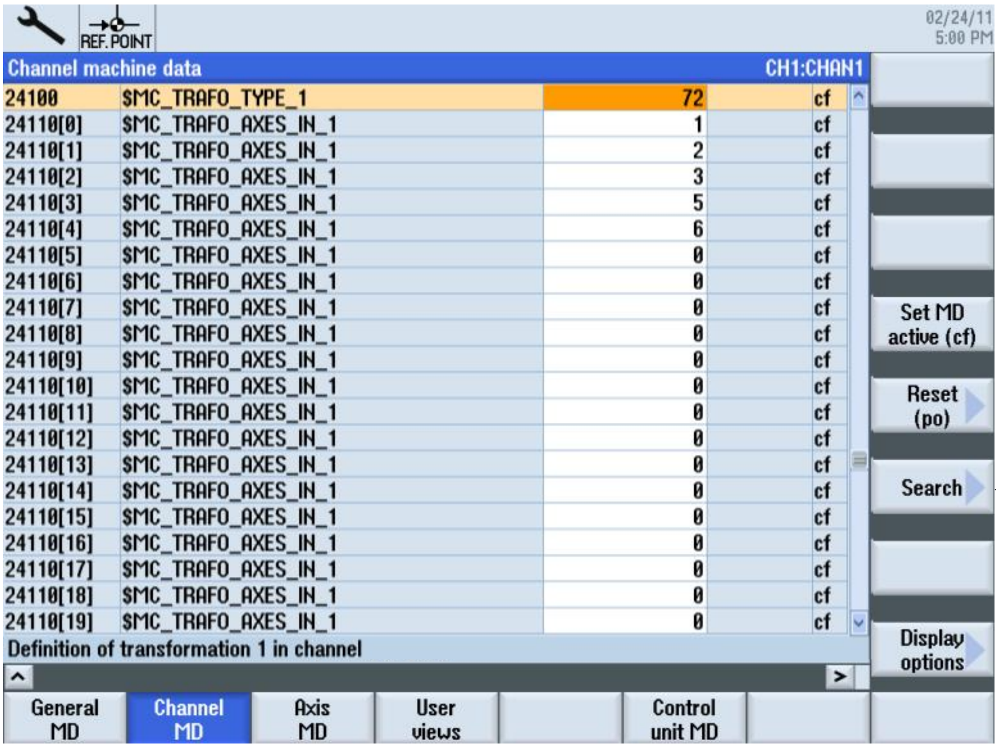
Swivel rotary table



To simplify the commissioning procedure you can select “**TRAFO_TYPE_1 = 72**” for setup of the 1st 5-axis transformation.

Press following softkeys to enter machine data for setup of a 5-axis transformation.

The following machine data meet the minimum requirement for setting up a 5-axis transformation “TRAORI” with “TRAFO_TYPE 72”.



24100	$MC_TRAFO_TYPE_1	72	cf
24110[0]	$MC_TRAFO_AXES_IN_1	1	cf
24110[1]	$MC_TRAFO_AXES_IN_1	2	cf
24110[2]	$MC_TRAFO_AXES_IN_1	3	cf
24110[3]	$MC_TRAFO_AXES_IN_1	5	cf
24110[4]	$MC_TRAFO_AXES_IN_1	6	cf
24110[5]	$MC_TRAFO_AXES_IN_1	0	cf
24110[6]	$MC_TRAFO_AXES_IN_1	0	cf
24110[7]	$MC_TRAFO_AXES_IN_1	0	cf
24110[8]	$MC_TRAFO_AXES_IN_1	0	cf
24110[9]	$MC_TRAFO_AXES_IN_1	0	cf
24110[10]	$MC_TRAFO_AXES_IN_1	0	cf
24110[11]	$MC_TRAFO_AXES_IN_1	0	cf
24110[12]	$MC_TRAFO_AXES_IN_1	0	cf
24110[13]	$MC_TRAFO_AXES_IN_1	0	cf
24110[14]	$MC_TRAFO_AXES_IN_1	0	cf
24110[15]	$MC_TRAFO_AXES_IN_1	0	cf
24110[16]	$MC_TRAFO_AXES_IN_1	0	cf
24110[17]	$MC_TRAFO_AXES_IN_1	0	cf
24110[18]	$MC_TRAFO_AXES_IN_1	0	cf
24110[19]	$MC_TRAFO_AXES_IN_1	0	cf

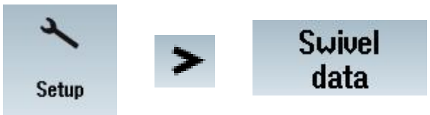
Search for **MD24100**. In this trafo machine data set you can now setup the first 5-axis transformation in the channel.

Notes

Machine data	Value	Description
MD24100 $MC_TRAFO_TYPE_1	=72	Kinematic type for 1st 5-axis transformation. With Trafotyp „**72**" the tool carrier parameters ($TC_CARR...) are being used for definition of the machine kinematic.
MD24110[0] $MC_TRAFO_AXES_IN_1 **MD24110[1] $MC_TRAFO_AXES_IN_1** **MD24110[2] $MC_TRAFO_AXES_IN_1** **MD24110[3] $MC_TRAFO_AXES_IN_1** **MD24110[4] $MC_TRAFO_AXES_IN_1**	**=1** **=2** **=3** **=5** **=6**	Assignment of channel axes for the 1st 5-axis transformation [0] = 1st channel axis (e.g. X) [1] = 2nd channel axis (e.g. Y) [2] = 3rd channel axis (e.g. Z) [3] = 5th channel axis (e.g. A) [4] = 6th channel axis (e.g. C)
MD24120[0] $MC_TRAFO_GEOAX_ASSIGN_TAB_1 **MD24120[1] $MC_TRAFO_GEOAX_ASSIGN_TAB_1** **MD24120[2] $MC_TRAFO_GEOAX_ASSIGN_TAB_1**	**=1** **=2** **=3**	Assignment of geometry axes to channel axes for the 1st 5-axis transformation [0] = First geometry axis X (G17) [1] = Second geometry axis Y (G17) [2] = Third geometry axis Z (G17)
MD24130 $MC_TRAFO_INCLUDES_TOOL_1	**=1**	Tool handling with 1st 5-axis trafo
MD24520[0] $MC_TRAFO5_ROT_SIGN_IS_PLUS_1 **MD24520[0] $MC_TRAFO5_ROT_SIGN_IS_PLUS_1**	**=1** **=1**	Sign of 1st rotary axis is plus (e.g. A) Sign of 2nd rotary axis is plus (e.g. C)
MD24574[0] $MC_TRAFO5_BASE_ORIENT_1 **MD24574[1] $MC_TRAFO5_BASE_ORIENT_1** **MD24574[2] $MC_TRAFO5_BASE_ORIENT_1**	=0 =0 =1	Basic tool orientation vector (e.g. G17) [0] = First geometry axis X [1] = Second geometry axis Y **[2] = Third geometry axis Z**
MD24580 $MC_TRAFO5_TOOL_VECTOR_1	=1	Direction of orientation tool vector in Z
MD24582 $MC_TRAFO5_TCARR_NO_1	**=1**	Assignment of 1st orientable tool carrier data (e.g. TCARR=1) to 1st 5-axis trafo.
MD21180 $MC_ROT_AX_SWL_CHECK_MODE	=2	Check software limits for orientation axes
MD24590 $MC_TRAFO5_ROT_OFFSET_FROM_FR_1	=1	Allow rotary axes offset in WO
MD10602 $MN_FRAME_GEOAX_CHANGE_MODE	=1	Total frame remains active after TRAORI
Activate the modified machine data with the "**Set MD active (cf)"** Softkey.		

Notes

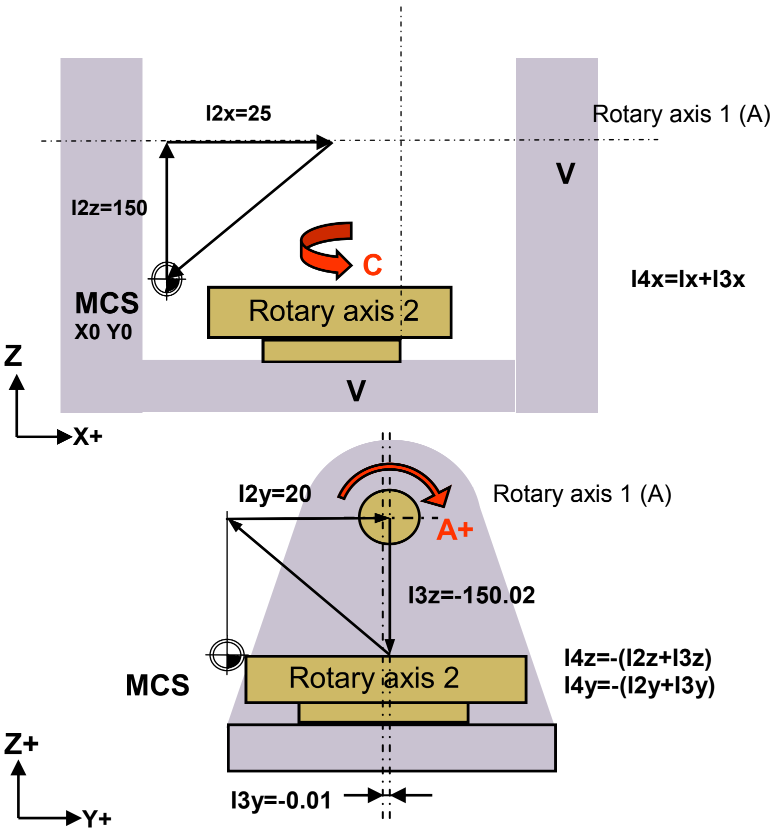
The Kinematic of the rotary swivel table can be set up through the input mask of the swivel data record (e.g. TCARR=1). Press the following softkeys to open the swivel data record:



Kinematic channel1				Name of swivel data record
Name: TABLE	Kinematics	Swivel table		No.: 1
Retract:	Z or Z, XY or max. in tool direction or inc. in tool dir.			
	X	Y	Z	
Retract position	200.000	200.000	300.000	[mm]
Offset vector I2	250.000000	200.000000	150.000000	[mm]
Rotary axis vector V1	-1.000000	0.000000	0.000000	
Offset vector I3	0.000000	-0.010000	-150.020000	[mm]
Rotary axis vector V2	0.000000	0.000000	-1.000000	
Offset vector I4	-250.000000	-199.990000	0.020000	[mm]

Only the definition of the kinematics type (Swivel table), Offset vectors I2,I3,I4 and Rotary axis vectors V1,V2 are evaluated for the 5 -axis trafo with type 72

F



Front view in +Y direction of the machine

Front view in +X direction of the machine

Notes

Example of a swivel data record with TCARR-variables

*Swivel data record AC Swivel rotary table type "**P**"*

```
$TC_CARR1[1]=0             ;Offset vector l1 (X)
$TC_CARR2[1]=0             ;Offset vector l1 (Y)
$TC_CARR3[1]=0             ;Offset vector l1 (Z)
$TC_CARR4[1]=250           ;Offset vector l2 (X)
$TC_CARR5[1]=200           ;Offset vector l2 (Y)
$TC_CARR6[1]=150           ;Offset vector l2 (Z)
$TC_CARR7[1]=-1            ;Rotary axis vector V1 (X)
$TC_CARR8[1]=0             ;Rotary axis vector V1 (Y)
$TC_CARR9[1]=0             ;Rotary axis vector V1 (Z)
$TC_CARR10[1]=0            ;Rotary axis vector V2 (X)
$TC_CARR11[1]=0            ;Rotary axis vector V2 (Y)
$TC_CARR12[1]=-1           ;Rotary axis vector V2 (Z)
$TC_CARR13[1]=0
$TC_CARR14[1]=0
$TC_CARR15[1]=0            ;Offset vector  l3 (X)
$TC_CARR16[1]=-0.01        ;Offset vector  l3 (Y)
$TC_CARR17[1]=-150.02      ;Offset vector  l3 (Z)
$TC_CARR18[1]=-250         ;Offset vector  l4 (X)
$TC_CARR19[1]=-199.99      ;Offset vector  l4 (Y)
$TC_CARR20[1]=0.02         ;Offset vector  l4 (Z)
$TC_CARR23[1]="P"          ;Kinematic type
$TC_CARR24[1]=0            ;Offset of rotary axis 1 in degrees *
```

```
$TC_CARR25[1]=0            ;Offset of rotary axis 2 in degrees *
$TC_CARR26[1]=0            ;Offset of the Hirth gearing in degrees for rotary axis 1 *
$TC_CARR27[1]=0            ;Offset of the Hirth gearing in degrees for rotary axis 2 *
$TC_CARR28[1]=0            ;Increment of the Hirth gearing in degrees for rotary axis 1 *
$TC_CARR29[1]=0            ;Increment of the Hirth gearing in degrees for rotary axis 2 *
$TC_CARR30[1]=-100         ;1st rotary axis min. range (only used by CYCLE800)
$TC_CARR31[1]=0            ;2nd rotary axis min. range (only used by CYCLE800)
$TC_CARR32[1]=100          ;1st rotary axis max. range (only used by CYCLE800)
$TC_CARR33[1]=360          ;2nd rotary axis max. range (only used by CYCLE800)
$TC_CARR34[1]="TABLE"      ;Name of swivel data record (only used by CYCLE800)
$TC_CARR35[1]="A"          ;Rotary axis 1 identifier (only used by CYCLE800)
$TC_CARR36[1]="C"          ;Rotary axis 2 identifier (only used by CYCLE800)
$TC_CARR37[1]=415018005    ;Display variants swivel mode (only used by CYCLE800)
$TC_CARR38[1]=200          ;Retract position X (only used by CYCLE800)
$TC_CARR39[1]=200          ;Retract position Y (only used by CYCLE800)
$TC_CARR40[1]=300          ;Retract position Z (only used by CYCLE800)
M30
```

* Only relevant for CYCLE800

M103: END

Notes

M104

Sinumerik
Operate

CYCLE832

Module Description:

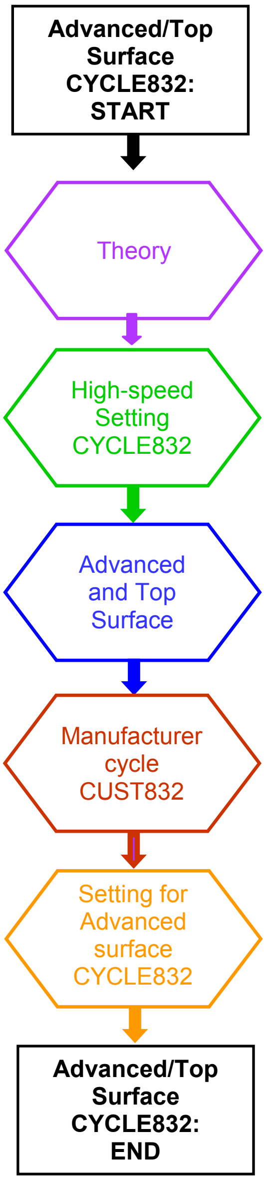
This Module explains the function and applications of the “High-speed setting” CYCLE832.
With use of programming examples it is shown, how to optimize HSC programs with CYCLE832.
In addition you find a detailed explanation of all commands and their functions, that are associated with “Advanced Surface and Top Surface”. At the end you will find a list of all important machine data settings that are required for setting up “Advanced Surface”.

Module Objective:

This module explains the use and application of the “High-speed setting” CYCLE832 in conjunction with the option “Advanced Surface and Top Surface“ and all associated commands.

Content:

840D sl SINUMERIK Operate


Advanced/Top Surface CYCLE832: START
Theory
High-speed Setting CYCLE832
Advanced and Top Surface
Manufacturer cycle CUST832
Setting for Advanced surface CYCLE832
Advanced/Top Surface CYCLE832: END

Notes

Theory

Precision, speed, surface quality

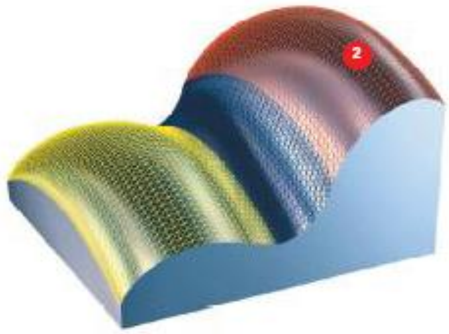
Special attention must be paid to the **CAD -> CAM -> (post processor) -> CNC** process chain when machining three-dimensional geometries, e.g. free-form surfaces.

CAM systems generate NC programs for free-form surface machining. The CAM system receives the workpiece geometry from a CAD system.

The CNC machine has to process the NC data generated by the post processor and convert it into axis movements.

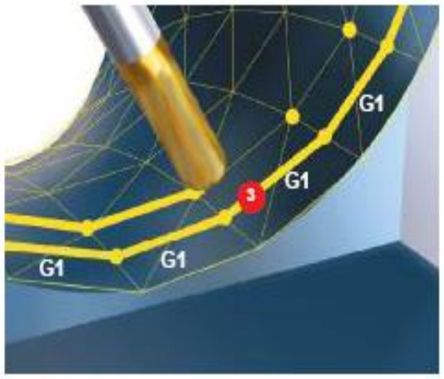
In CAD systems, surfaces (1) of higher orders are constructed (free-form surfaces).

For example, in order to be able to mill an entire surface - or for collision checking - the CAM system generally converts the CAD freeform surface into a polyhedron .



This means that the smooth design surface is approximated by a number of individual small planes (2).

This produces deviations from the original free-form surface.



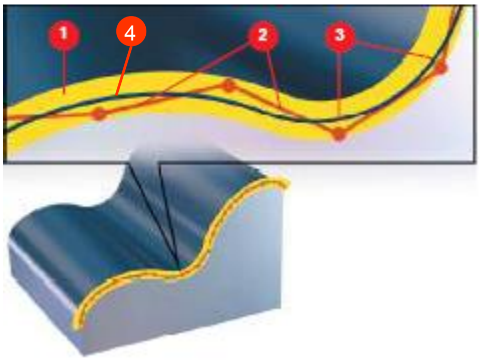
The CAM programmer overlays this polyhedron with tool paths. From these, the post processor generates NC blocks within the specified error tolerances (chord tolerance). These usually comprise many short straight line elements, G1 X Y Z (3).

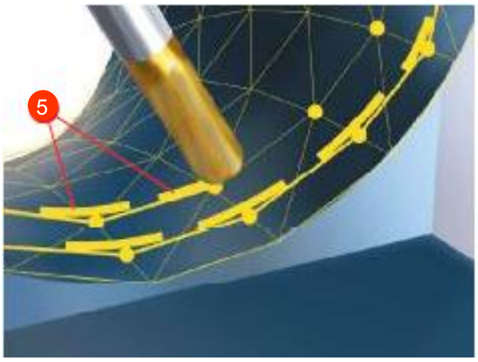
Thus, the machining result is no longer a freeform surface, but a polyhedron. The small planes of the polyhedron can be visibly mapped on the surface.

This can result in undesirable re-machining.

Notes

The control offers various functions which can help to avoid rework, such as:



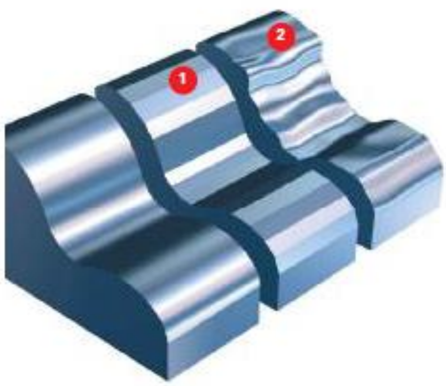


Compressor function (COMPCAD)

CAD/CAM-Systems generate usually linear blocks, in respect of the parameterized accuracy. This can cause extremely high amount of data, when working with complex contours (freeform surfaces) and short path sections.
These path sections (**3**) limit the machining velocity.

According to the specified tolerance band (**1**) the compressor combines a sequence of G1 commands (**2**) and compresses them into a spline (**4**) which is directly executable by the control system.

The compressor generates smooth paths and paths with constant curvature. The constant curvature results in a steady velocity and acceleration characteristic, meaning that the machine can run at higher speeds, thereby increasing productivity.



Programmable contour smoothing (G645)

Defined corner rounding at block boundaries involves inserting geometrical elements (**5**) at the corners.

The tolerance (chord tolerance) can be altered.

Linear interpolation at the block transitions leads to acceleration jumps in the machine axes, which in turn can cause resonance in the machine elements and can ultimately be detected on the workpiece surface as a bevelled pattern (**1**) or as vibration (**2**).

Use of the compressor and programmable smoothing functions has the following effect on the program, the workpiece and the machine

- Reduction of part program blocks that define the workpiece (smaller programs).
- Smooth block to block transitions that make a smooth surface (machine resonance is avoided).
- Higher path velocity and reduction of machine load.

Notes

CYCLE832 "High-speed settings"

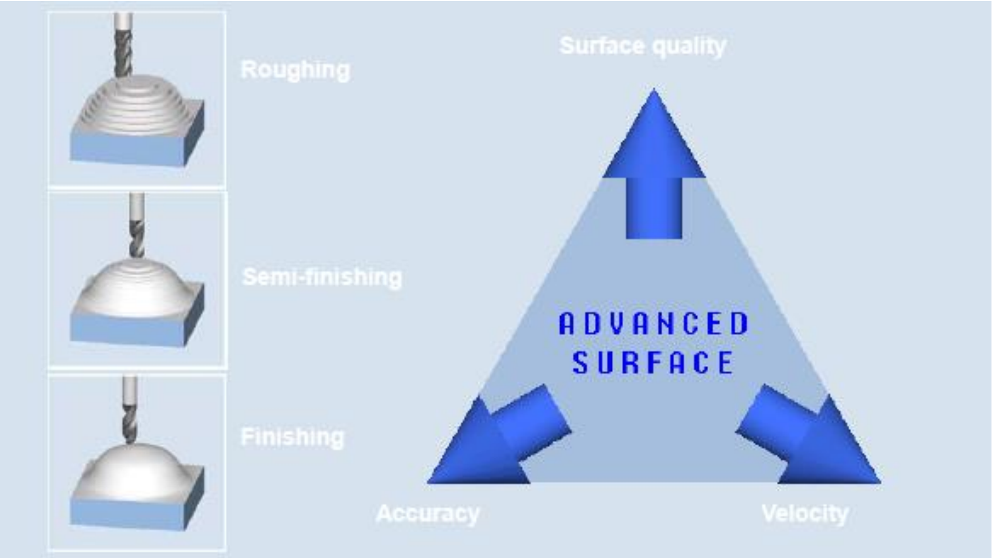
Fundamentals

CYCLE832 provides optimal support for the execution of NC programs during the milling of free-form surfaces in the HSC range of 3- and 5-axis machining operations.
When executing CAM programs in the HSC range, the control must be able to calculate a feed profile in advance for high feedrates with short path sections (NC blocks). Smooth surfaces with high accuracy in the µm range at high feedrates of >10 m/min are expected.
Four different machining types of the "**Dynamic G-code group 59**" can be selected in CYCLE832 "High-Speed Setting" and their dynamic response parameters activated.

CYCLE832 can be set by the machine operator or, within the context of NC program generation, by the post processor or programmer. Dynamic values and NC commands can be adapted to be user-specific and they depend on the settings of the machine data (machine manufacturer).

Machining type	Dynamic G-code group 59	Field index
Roughing	DYNROUGH (HSC settings)	3
Semifinishing	DYNSEMIFINISH (HSC settings)	2
Finishing	DYNFINISH (HSC settings)	1
Deselection	DYNNORM (Standard settings)	0

The four machining types in CYCLE832 are in direct conection with surface quality, accuracy and velocity of the contour path (see picture).



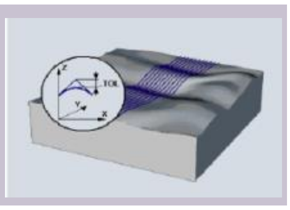
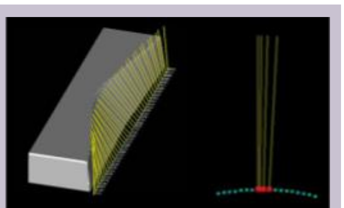
In all cases, specifying a tolerance ensures that the correct machining contour is achieved in order to obtain the desired surface quality and accuracy.
Generally, a higher tolerance is selected for roughing than for finishing.

Notes

Programming procedure for CYCLE832

With the function High-speed settings (CYCLE832) machine data for machining of freeform surfaces are being preset for best possible machining results.

The call of CYCLE832 „High-speed settings“ includes the following parameters:

Machining 	Machining type (Plain text) • Deselect = _OFF • Finishing ▼▼▼ = _FINISH • Prefinish ▼▼ = _SEMIFIN • Roughing ▼ = _ROUGH For „**Multi-axis programming yes**“, the following plain texts are generated in accordance with the machining type: • Finishing ▼▼▼ = _ORI_FINISH • Prefinish ▼▼ = _ORI_SEMIFIN • Roughing ▼ = _ORI_ROUGH	Selection of machining parameters for the particular technologies. With "Deselet" all active G-codes are reset to the settings agreed upon in the manufacturer cycle CUST_832.
Tolerance	Chord tolerance in mm (must be imported from CAM system 1:1) 	Contour smoothing tolerance for linear axes calculated internally in CYCLE832(CAM tolerance * sqrt3) and passed to ***MD33100 $MA_COMPRESS_POS_TOL*** for linear axes.
Multi-axis programing	• Yes: The field „**ORI tolerance**“ appears. Input value > 0 in degrees • No: The value 1 (version recognition) is entered automatically.	Selection for 5-axis machining with TRAORI. Enables the input field „ORI tolerance“ **Note:** This field can be hidden with **SD55220 Bit1=1**.
ORI tolerance	Orientation tolerance in degrees (≈ **factor 10** of linear tolerance) 	Has two meanings: 1. Smooting tolerance for rotary axes with active TRAORI. 2. Vektor orientation smooting with active ORISON withing the specified tolerance.

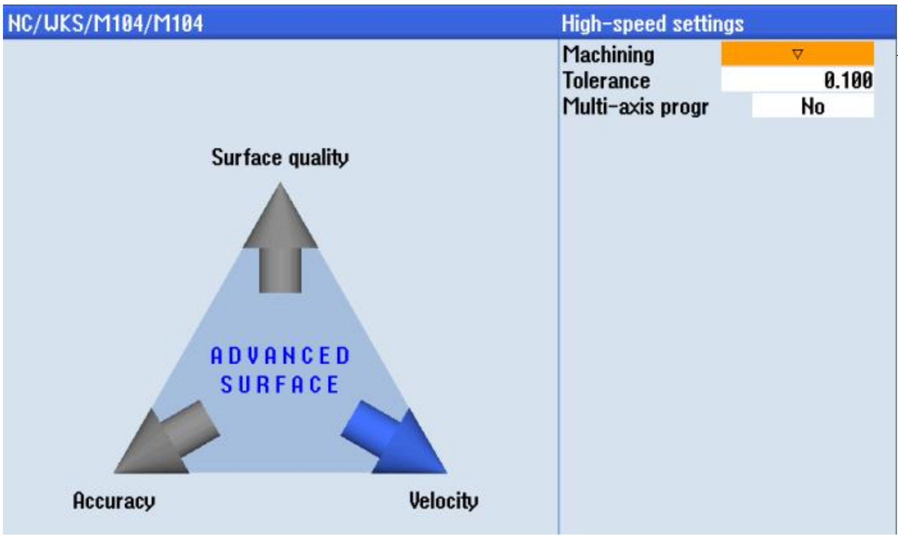
The operator can freely select between accuracy, path velocity and surface quality, through selection of the “**machining**” type and input of a “**tolerance**” value. With selection of „**Multi-axis program yes**“ the smooting tolerance for rotary axes can be entered. With activation of the orientation smoothing function ORISON additional vector smoothing is preformed within the defined tolerance in the input field „**ORI tolerance**“.

Press the following softkeys in the program editor to open CYCLE832 „High-speed settings“:

→

Notes

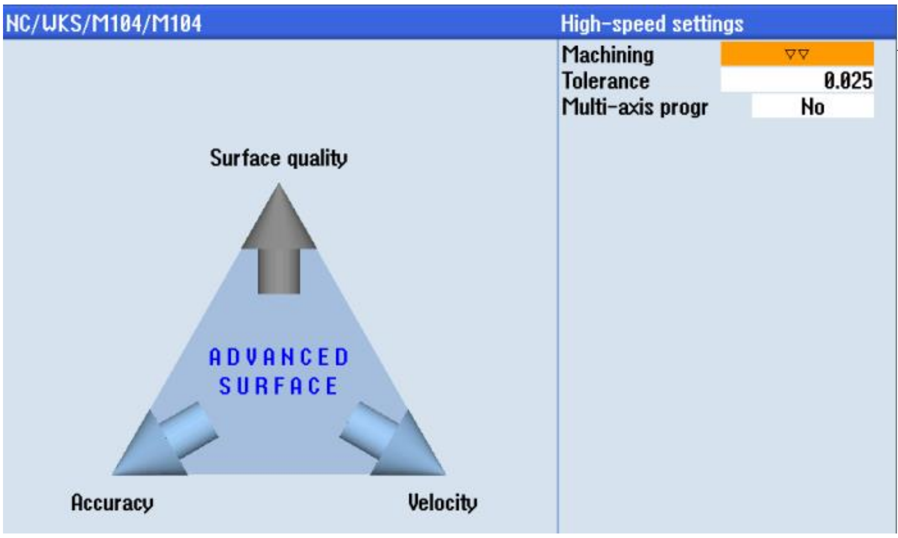
- When **roughing**, the emphasis is on velocity 1 due to large contour smoothing tolerance, accuracy and surface quality are here neglected (dynamic parameters of technology G-code group DYNROUGH are activated).



The parameters in CYCLE832 are transferred with the "**Accept**" softkey into the program as follows: **CYCLE832(0.1,_ROUGH,1)**

Tolerance | Machining type | Version recognition

- When **prefinish,** a compromise between accuracy 2 and velocity 1 is made, surface quality 3 is slightly improved (dynamic parameters of technology G-code group DYNSEMIFIN are activated).

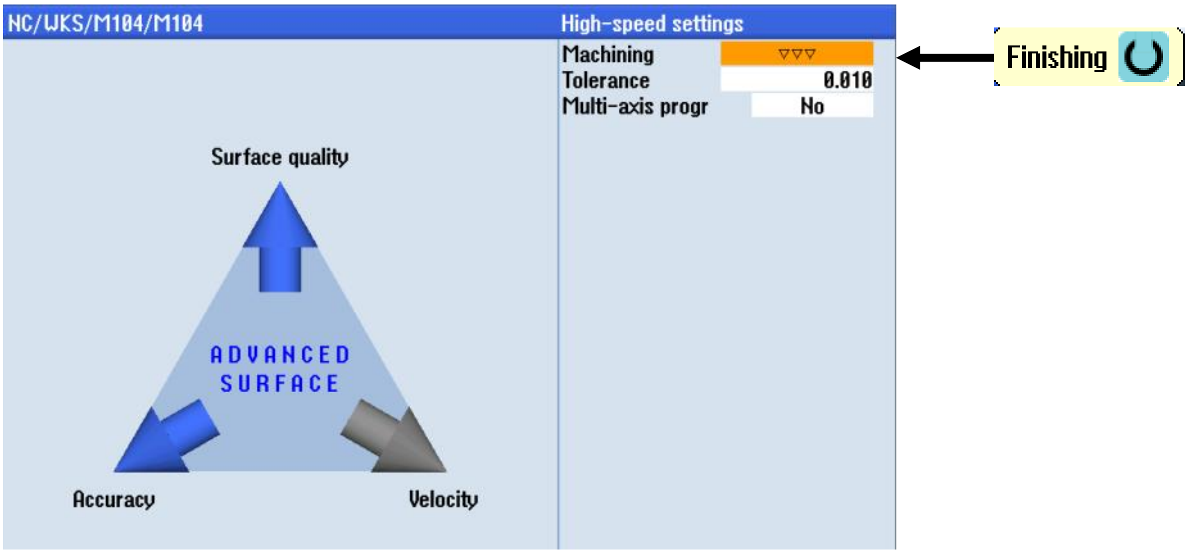


The parameters in CYCLE832 are transferred with the "**Accept**" softkey into the program as follows: **CYCLE832(0.02,_SEMIFIN,1)**

Tolerance | Machining type | Version recognition

Notes

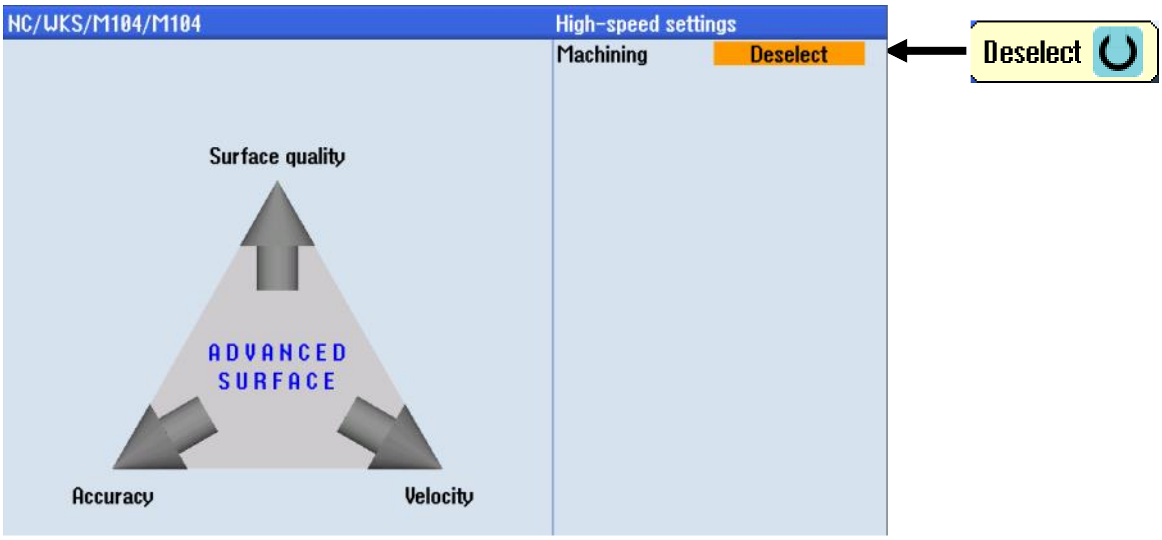
- When **finishing,** the emphasis is on accuracy **2** and surface quality **3,** the velocity is neglected to a certain extend (dynamic parameters of technology G-code group DYNFINISH are activated).



The parameters in CYCLE832 are transferred with the "**Accept**" softkey into the program as follows: **CYCLE832(0.01,_FINISH,1)**

Tolerance | Machining type | Version recognition

- When **deselect**, the standard settings for machining without "Advanced Surface" are restored (dynamic parameters of technology G-code group DYNNORM are activated).



The parameters in CYCLE832 are transferred with the "**Accept**" softkey into the program as follows: **CYCLE832(0,_OFF,1)**

Tolerance | Machining type | Version recognition

Notes

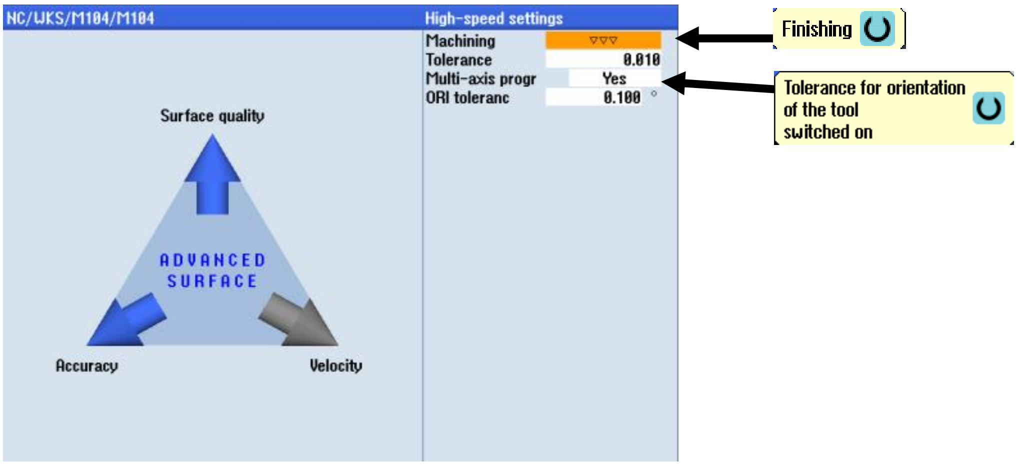
Multi-axis programms with mit active 5-axis transformation TRAORI

With selection of „**Multi-axis programming yes**“ additional smoothing tolerance for rotary axes can be defined, which is passed to the compressor position tolerance for each rotary axes. In general this tolerance value should be approximately **10** times higher then the linear tolerance.

With programming of „**ORISON**“ additional smoothing of fluctuating vector orientations across several blocks within the specified orientation tolerance can be applied.

Since the orientation tolerance also has an effect on ORISON by default, it is important when activating ORISON to ensure that the orientation tolerance is not too great, because otherwise too much orientation smoothing will result. In most cases, over-smoothing occurs during roughing operations, which can cause unexpected contour violations.

With selection of „**Multi-axis programming yes**“ the default value „1“ (Version recognition) is over written with the input of a value > 0 in the input field „**ORI tolerance**“.



The parameters in CYCLE832 are transferred with the “**Accept**” softkey into the program as follows: **CYCLE832(0.01,_ORI_FINISH,0.1)**

Tolerance	Machining type	Oreintation tolerance

Note:

The cycle support through the input mask for the CYCLE832 stands in direct connection with the option "Advanced Surface". Should the option "Advanced Surface" not be activated then CYCLE832 is also not supported.

Notes

Program structure with CYCLE832

Ideally, you should program CYCLE832 in the higher-level NC master program that then calls the geometry program. This means that you can apply the cycle to the complete geometry or depending on the transparency of the CAM program - to individual program sections.

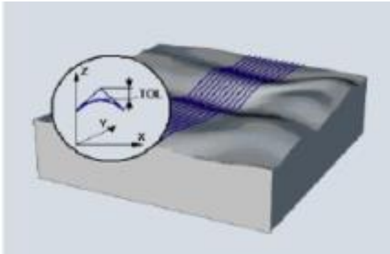
CYCLE832 Parameter from SW 2.6

Programming the cycle: **CYCLE832(Tolerance, Machining type, Version recognition)**

- Tolerance (imported 1:1 from CAM)
- Machining type
  0 = Deselect
  1 = Finishing ▼▼▼
  2 = Prefinish ▼▼
  3 = Roughing ▼

- CYCLE832 Parameter from SW 2.6

Programming example with CYCLE832 from SW 2.6



The tolerance for contour smoothing is generally calculated internaly in CYCLE832 as follows:
CTOL = sqrt(3) * CAM-tolerance

The tolerance for orientation smoothing of 5-axis programs with **TRAORI** is calculated internaly based on the computed value for CTOL in CYCLE832 as follows:
OTOL=CTOL*Factor(e.g. finishing default value from SD55441 = 10).

The smoothing tolerances for CTOL and OTOL indicate the maximum permissible path deviation for linear and rotary axes during smoothing with G645 and are passed to ***MD33100$MA_COMPRESS _POS_TOL*** for each axis. Vector orientation smoothing „ORISON“ is in SW 2.6 **not** yet **functioning.**

Programming example with CYCLE832 from SW 2.6

;*3-AXIS-PROGRAM*****	
N10 T1 D1	;Tool call
N11 M6	;Tool change
N12 G54 G710	;Work offset, metric programming
N13 M3 S12000 F10000	;Spindle ON CW, spindle speed, feed
N14 CYCLE832(0.1,3,1)	;Tolerance “**0.1**” mm, machining type “**Roughing**”
N15 EXTCALL"CAM_ROUGH"	;External subprogram call “**CAM_ROUGH**”
;*5-AXIS-PROGRAM*****	
N16 T2 D1	;Tool call
N17 M6	;Tool change
N18 M3 S12000 F10000	;Spindle ON CW, spindle speed, feed
N19 CYCLE832(0.005,1,1)	;Tolerance “**0.005**” mm, machining type “**Finishing**”
N20 TRAORI	;Activate 5-axis transformation
N21 ORIAXES	;Orientation interpolation
N22 ORIWKS	;Orientation reference WKS
N23 EXTCALL"CAM_FINISH"	;External subprogram call “**CAM_FINISH**”
N24 M30	;End of program

Notes

Programming example with CYCLE832 as of SW 2.7 to SW 4.4

;*3-AXIS PROGRAM*****	
N10 T1 D1	;Tool call
N11 M6	;Tool change
N12 G54 G710	;Work offset, metric programming
N13 M3 S12000 F10000	;Spindle ON CW, spindle speed, feed
N14 CYCLE832(0.1,3,1)	;Tolerance “**0.1**” mm, machining type “**Roughing**”
N15 EXTCALL"CAM_ROUGH"	;External subprogram call “**CAM_ROUGH**”
;*5-AXIS PROGRAM*****	
N16 T2 D1	;Tool call
N17 M6	;Tool change
N18 G54 G710	;Work offset, metric program
N19 M3 S12000 F10000	;Spindle ON CW, spindle speed, feed
N20 CYCLE832(0.005,1,1)	;Tolerance “**0.005**” mm, machining type “**Finishing**”
N21 TRAORI	;Activate 5-axis transformation
N22 ORIAXES	;Orientation interpolation
N23 ORIWKS	;Orientation reference WKS
N24 **ORISON**	;Orientation vector smoothing “on”
N25 **OTOL=0.05**	;Smoothing tolerance for rotary axes and vectors
N26 EXTCALL"CAM_FINISH"	;External subprogram call “**CAM_FINISH**”
N27 M30	;End of program

The tolerance for contour smoothing is generally calculated internally in CYCLE832 as follows:
CTOL = sqrt(3) * CAM-tolerance

The tolerance for orientation smoothing of 5-axis programs with **TRAORI** is calculated internally based on the computed value for CTOL in CYCLE832 as follows:
OTOL=CTOL*Factor (e.g. finishing default value from SD55441 = 10), or can be overwritten in the program with the command “**OTOL=**“.

The calculated smoothing tolerances for CTOL and OTOL indicate the maximum permissible path deviation for linear and rotary axes during smoothing with G645 and are passed to ***MD33100$MA_COMPRESS _POS_TOL*** for each axis.

Attention! Since OTOL also has an effect on ORISON by default, it is important when activating ORISON to ensure that the orientation tolerance is not too great, because otherwise too much orientation smoothing will result.
In most cases, over-smoothing occurs during roughing operations, which can cause unexpected contour violations. Therefore it is recommended to program “**OTOL=**“ outside of CYCLE832 (see example 5-axis-program NC block N25).

The orientation smoothing "ORISON" **functions** as of SW 2.7. When the orientation smoothing "**ORISON**" is activated, any vector fluctuations are also smoothed in advance (Look Ahead) within the orientation tolerance specified with "**OTOL=**".

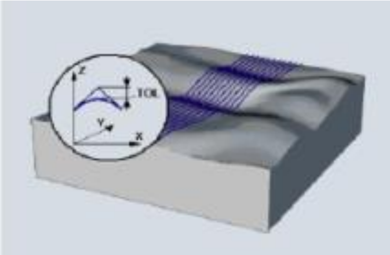
Note:

A detailed explanation for the programmable smoothing tolerances with CTOL and OTOL, and orientation smoothing with ORISON can be found in page 29

Notes

CYCLE832 parameters from SW 4.7

Programming the cycle: **CYCLE832(Tolerance, Machining, ORI-tolerance)**



- Tolerance (imported 1:1 from CAM)

- Machining type (Plain text)
  _OFF = Deselect
  _FINISH = Finishing ▼▼▼
  _SEMIFIN = Prefinish ▼▼
  _ROUGH = Roughing ▼

  For „Multi-axis programming yes"
  _ORI_FINISH = Finishing with ORI tolerance ▼▼▼
  _ORI_SEMIFIN = Prefinish with ORI tolerance ▼▼
  _ORI_ROUGH = Roughing ORI tolerance ▼

- Orientation tolerance (rotary axis and vector smoothing only if „Multi-axis programming yes")

Programming example with CYCLE832 from SW 4.7

;*3-AXIS-PROGRAM*****

N10 T1 D1	;Tool call
N11 M6	;Tool change
N12 G54 G710	;Work offset, metric programming
N13 M3 S12000 F10000	;Spindle ON CW, spindle speed, feed
N14 CYCLE832(0.1,_ROUGH,1)	;Tolerance "**0.1**" mm, machining type "**Roughing**"
N15 EXTCALL"CAM_ROUGH"	;External subprogram call "**CAM_ROUGH**"

;*5-AXIS-PROGRAM*****

N16 T2 D1	;Tool call
N17 M6	;Tool change
N18 G54 G710	;Work offset, metric program
N19 M3 S12000 F10000	;Spindle ON CW, spindle speed, feed
N20 CYCLE832(0.005,_ORI_FINISH,0.05)	;Tolerance "**0.005**" mm, machining type "**Finishing**" ;with ORI tolerance "**0,05**" degree
N21 TRAORI	;Activate 5-axis transformation
N22 ORIAXES	;Orientation interpolation
N23 ORIWKS	;Orientation reference WKS
N24 **ORISON**	;Orientation vector smoothing "on"
N25 EXTCALL"CAM_FINISH"	;External subprogram call "**CAM_FINISH**"
N26 M30	;End of program

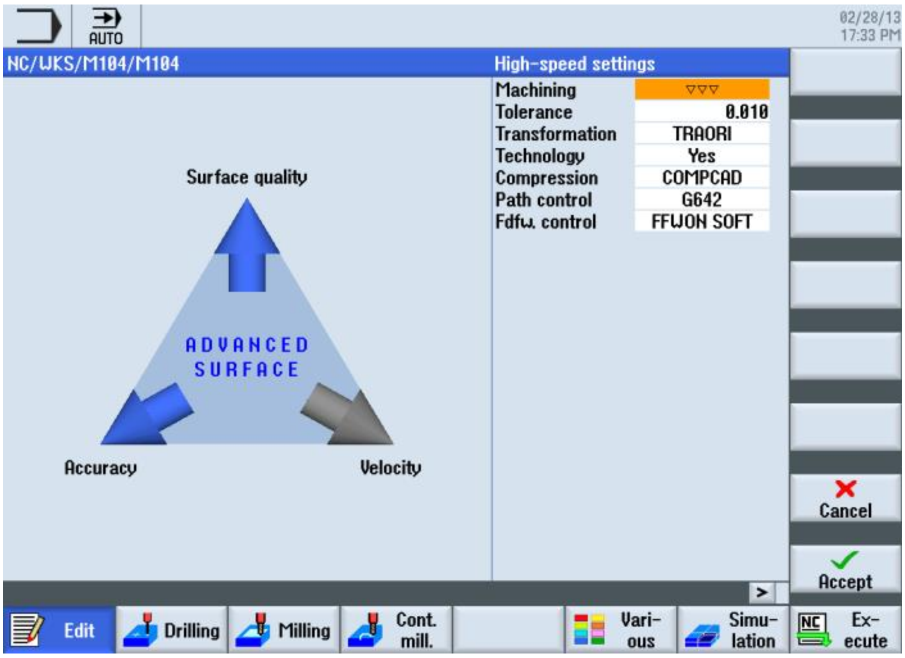
On selection of „**Multi-axis programming yes**" the tolerance for orientation smoothing in 5-axis programs with **TRAORI** can be directly defined in the parameter „**ORI-tolerance**".
With programming of „**ORISON**" additional smoothing of fluctuating vector orientations across several blocks within the specified orientation tolerance can be applied.

Attention ! Since OTOL also has an effect on ORISON by default, it is important when activating ORISON to ensure that the orientation tolerance is not too great, because otherwise too much orientation smoothing will result. In most cases, over-smoothing occurs during roughing operations, which can cause unexpected contour violations.

Notes

Compatibility for CYCLE832 up to SW 7.5

Compatibility of older versions of CYCLE832 and CYC_832T are taken into consideration in the new version of CYCLE832. The compatibility mode must be activated in the setting data **SD55220 $SCS_FUNCTION_MASK_MILL_TOL_SET** with setting of ***Bit 0 = 1***.



Programing of the cycle: **CYCLE832(Tolerance, Technology)**

- Tolerance
- Technology (bit coded)

  Bit 0: Machining type (0 = deselect, 1 = finishing, 2 = semifinishing, 3 = roughing)
  Bit 1: No function
  Bit 2: Transformation (0=TRAFOOF, 1=TRAORI, 2=TRAORI(2))
  Bit 3: Continuous path mode (0=G64,1=G641, 2=G642)
  Bit 4: Velocity control and jerk limitation (0=FFWON SOFT, 1=FFWOF SOFT, 2=FFWOF BRISK)
  Bit 5: Compression (0=nein/COMPOF, 1=COMPCAD, 2=COMPCURVE, 3=B-SPLINE)

The parameters in CYCLE832 are transferred with the “**Accept**” softkey into the program as follows: **CYCLE832(0.01,112101)**

Tolerance Technology

Notes

Advanced Surface

Fundamentals

General explanation:

Advanced Surface is a software option (license required) developed for the improvement of surface quality and reduction of machining time of freeform surfaces in the mould and die HSC (high speed cutting) machining sector.

SINUMERIK bundles a range of new functions which have been incorporated into the controller under the Advanced Surface concept. For you, the user, this new, intelligent motion control means an optimal workpiece surface while at the same time providing maximum machining velocity. Advanced Surface is automatically activated when you work with CYCLE832.

When the new motion control is used, an **optimized "Look Ahead"** function contributes to perfect surface quality through reproducible results in adjacent milling paths, accuracy, and increased velocity. The new, **optimized compressor** ensures exact **contour accuracy** and **maximum machining velocities**. Intelligent jerk limitation reduces wear in the machine's mechanics. It allows smooth acceleration and braking of the axes with full dynamics and extends the service life of the machine.

One important advance is the automatic harmonization of the velocity profiles on adjacent milling paths by means of CNC. It also works for the forward/backward line-by-line milling of contours and free-form surfaces and it leads directly to increased surface quality - or more precisely:
Perfect workpiece surfaces.

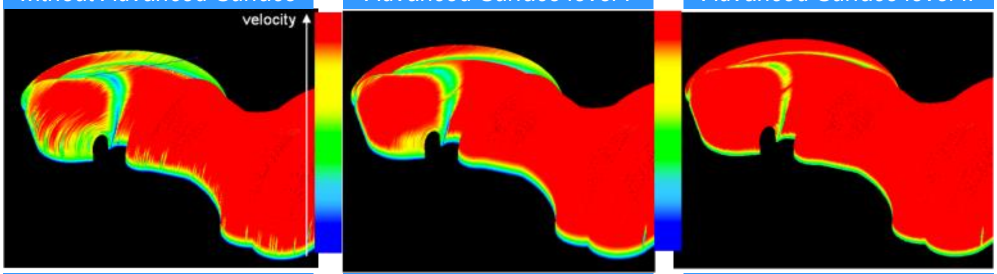
Advanced Surface level II - New Look Ahead level II

Level II is referred to as the further development of the Look Ahead of the freeform surface mode of level 1 (software version V2.6).

In software version V2.7 this function was further developed and improved, to increase quality and continuity of the path velocity profile and to reduce the machining time.

The main improvement lies herby in the acceleration and jerk limitation of the path profile (see pictures below).

without Advanced Surface | *Advanced Surface level I* | *Advanced Surface level II*



Machining time: 281 sec | *Machining time: 245 sec* | *Machining time: 231 sec*

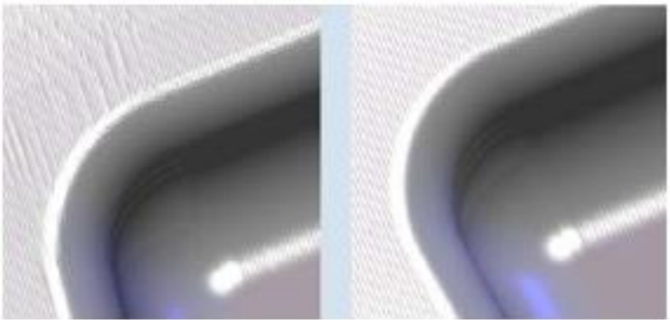
Velocity profile with standard path velocity control

Velocity profile with Advanced Surface - level I

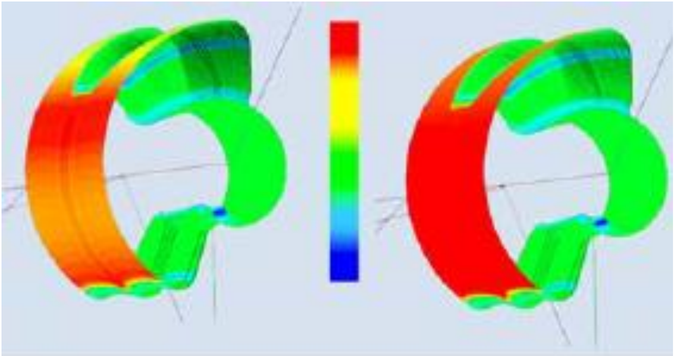
Velocity profile with Advanced Surface - level II

Increased machining velocity due to an improved velocity profile. The red areas show the maximum velocity.

Notes



Perfect surface quality thanks to reproducible results in adjacent milling paths. The surface is much more homogeneous.



The compressor does not only work via G1 sets. It also compresses rapid traverse movements G0 by means of smoothing.
The velocity remains at a constant high level during the entire machining process (red areas).

Preconditions for "Advanced Surface" with CYCLE832

The use of the option „Advanced Surface“ requires a propper optimization of the CNC machine during installation and setup (comissioning) by the machine manufacturer.

The following is required in detail:

- Drive optimization of all machine axes
- The prerequisite machine data, axis machine data and setting data are set in accordance to the recommended values by the control builder (see section 5 in this module).
- The dynamic G-code groups have been setup and are parameterizes for the following machining types:
  - Deselect (DYNNORM)
  - Roughing (DYNROUGH)
  - Prefinish (DYNSEMIFIN)
  - Finishing (DYNFINISH)

The following classification of the dynamic G-code group 59 is recommended:

DYNNORM ➞ 2.5D machining without AS
DYNPOS ➞ Positioning mode (ATC, taping, drilling) without AS
DYNROUGH ➞ 3D machining with AS
DYNSEMIFIN ➞ 3D machining with AS
DYNFINISH ➞ 3D machining with AS

ATC* = Automatic tool change

Notes

Associated NC commands

Important NC commands with Advanced Surface

The following G-code commands are preset in the CUST_832.SPF and activated on selection of the technology groups in CYCLE832 in conjunction with the option “Advanced Surface” in a optimum way.

- **DYNNORM, DYNROUGH, DYNSEMIFIN, DYNFINISH** (G-code group 59). With these NC commands you can activate the predefined dynamic parameter in the axes- and channel specific machine data for the appropriate technology (machining type). DYNNORM is the initial setting of the G-code group (default) called (see section 3.4).

- **COMPCAD** (G-code group 30) is used to combine part programs with short linear blocks (G1), with the associated tolerance, using polynomials. The path compression technique has been developed so that the surface character, which the milling paths form, is retained (see section 3.5-3.6)

- **G645** (G-code group 10) is used to switch into continuous path mode with Look Ahead and is used to insert smoothing elements in existing contour elements so that acceleration steps (jumps) do not occur (also see section 3.10)

- **SOFT** (G-code group 21) activates the Jerk limited velocity control. Must be active with Advanced Surface level II (see section 3.11).

- **FFWON** (G-code group 24) is used to switch on parameterized precontrol (speed or acceleration precontrol). FFWON can only be used if pre-control is parameterized. This must be ensured by the machine manufacturer (see section 3.12).

- **FIFOCTRL** (G-code group 4) is used to switch in the automatic pre-processing memory control. The feed is adapted so that the pre-processing memory does not become empty (see section 3.14).

Important NC commands for 5-axis HSC machining

The following G-code commands can be preset by the machine manufacturer in the CUST_832.SPF

- **TRAORI** enables the defined transformation and must be programmed separately in a block in the NC program (refer to modul M103 „5-axis transformation“).

- **UPATH** (G-code group 45) is used to switch the path parameters on for 5-axis interpolation (see section 3.13) .

- **ORIAXES** (G-code group 51) linearly interpolates the orientation axes in the block up to the end block (refer to modul M103 „5-axis transformation“).

- **ORIWKS** (G-code group 25) defines the workpiece coordinate system for orientation interpolation (refer to modul M103 „5-axis transformation“).

- **ORISON** (G-code group 61) activates orientation smoothing characteristics for 5-axis machining with active 5-axis transformation (TRAORI) (see section 3.16, also refer to modul M103 „5-axis transformation“).

Notes

Technology commands of the dynamic G-code group 59

General explanation:

Using the "Technology" G-code group, the appropriate dynamic response can be activated on the machine for five varying technological machining operations. The dynamic values and G codes are configured and set by the machine manufacturer. The G groups are switched and activated automatically when the machining method (e.g. finishing or roughing with CYCLE832 or CUST_832) is selected.

Machining type	Dynamic G-code group 59	Field index
Deselection	DYNNORM (Standard-Settings)	[0]
Positioning	DYNPOS (Not used in CYCLE832)	[1]
Roughing	DYNROUGH (HSC-Settings)	[2]
Prefinish	DYNSEMIFINISH (HSC-Settings)	[3]
Finishing	DYNFINISH (HSC-Settings)	[4]

Dynamic parameters can be adapted to the respective machining operation with the technology G groups. Using the commands of technology G group 59, the value of channel and axis-specific machine data is activated using the corresponding array index. These are, for example, jerk and acceleration values.

MD	Name	Description
MD20600	$MC_MAX_PATH_JERK[0..4]	Path oriented maximum jerk
MD20602	$MC_CURV_EFFECT_ON_PATH_ACCEL[0..4]	Influence of path curvature on path dynamic.
MD20603	$MC_CURV_EFFECT_ON_PATH_JERK[0..4]	Influence of path curvature on path jerk.
MD32300	$MA_MAX_AX_ACCEL[0..4]	Maximum axes acceleration
MD32310	$MA_MAX_ACCEL_OVL_FACTOR[0..4]	Overload factor for axial velocity jumps.
MD32431	$MA_ MAX_AX_JERK[0..4]	Maximum axial jerk for path motion.
MD32432	$MA_PATH_TRANS_JERK_LIM[0..4]	Maximum axial jerk at the block transition in continuous-path mode.
MD32433	$MA_SOFT_ACCEL_FACTOR[0..4]	Scaling of acceleration limitation for SOFT

Note:

The machine data for drive optimization are only to be set up by the machine manufacturer (danger of machine damage) through a certified service technician.

The dynamic settings and values are automatically activated, when calling the G-code command of the appropriate technology group (DYNNORM, DYNPOS, DYNROUGH, DYNSEMIFIN, DYNFINISH). No program stop occurs on call of the technology parameter.

Notes

Compressor function COMPCAD

General explanation:

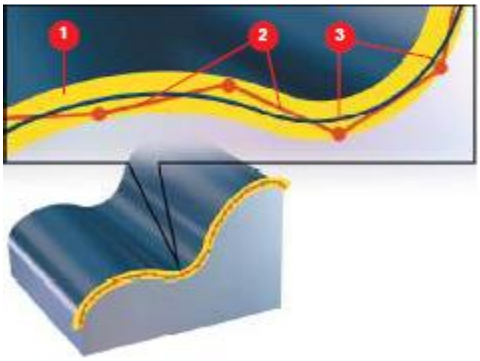
The compressor functions **COMPCAD** can generate one polynomial block from theoretically as many linear blocks as required. The polynomial blocks have constant velocity and acceleration at the block transitions. Corners that are desirable are identified as such and taken into account.

The maximum tolerable deviation of the calculated path to the programmed points can be specified using machine data for all compressor functions. Different than for COMPON and COMPCURV, for COMPCAD the specified tolerances are not utilized in different directions in neighboring paths.
In fact, COMPCAD attempts to achieve - under similar conditions - also similar deviations from the programmed points.

The common objective of compressor functions is to optimize the surface quality and machining speed by achieving continuous block transitions and increasing the path length for each block.

Operating principle of the compressor

Description of NC commands:



In accordance with the specified tolerance band (**1**), the compressor takes a sequence of G1 commands (**2**), combines them and compresses them into a spline (**3**), which can be directly executed by the control. A new contour is created whose characteristic lies within the specified tolerance range.

The compressor generates smooth paths and paths with constant curvature. The constant curvature results in a steady velocity and acceleration characteristic, meaning that the machine can run at higher speeds, thereby increasing productivity.

Notes

Compressor mode settings

In channel machine data MD20482 $MC_COMPRESSOR_MODE various settings for the compressor function can be made (see prerequiste machine data settings at the end of this module)

The following basic settings can be made:

- The compressor function COMPCAD compress straight blocks of the form: **N... G01 X... Y... Z... F…** and **N... G0 X... Y... Z...**
- The COMPCAD compressor function also compresses all types of circular blocks of the form: **N... G02/G03 X... Y... I... J... F...**
- When orientation transformation (TRAORI) is active, and under certain preconditions, the compressor function COMPCAD can also compress motion blocks for tool orientation and tool rotation.
- The position data in the blocks to be compressed can be realized as required, e.g. **X100, X=AC(100), X=R1*(R2+R3)**
- The compression operation is not interrupted by other commands, e.g. auxiliary function output, in and between the blocks to be compressed.

NC command	Description
COMPCAD	Compressor on: COMPCAD smoothes the point profile before approximation (B spline) and at high path speed offers maximum precision with constant-acceleration transitions (compression rate unlimited, but max. path length 5 mm) **Note:** Starting with SW 2.6 CYCLE832 uses **only COMPCAD** and is recommended for the milling of **freeform surfaces.**
COMPCURV	Compressor on: Continuous velocity and acceleration block transitions with approximation by polynomial, 5th degree. G1 blocks are approximated by a polynomial. The block transitions are jerk-free. **Note:** Was used up to SW 7.5 HMI Advanced in CYCLE832 and recommended for **circumferential milling**.
COMPOF	Compressor off

Notes

Continuous-path mode and Look Ahead G64xx

General explanation:

In the continuous-path mode the path velocity is not decelerated for the block change in order to permit the fulfillment of an exact stop criterion. The objective of this mode is to avoid rapid deceleration of the path axes at the block-change point so that the axis velocity remains as constant as possible when the program moves to the next block. To achieve this objective, the "**Look Ahead**" function is also activated when the continuous path mode is selected.
Continuous-path mode causes the smoothing and tangential shaping of angular block transitions by local changes in the programmed contour. The extent of the change relative to the programmed contour can be limited by specifying the **overload factor** or **rounding criteria**.

The purpose of continuous-path control is to increase speed and harmonize traversing performance. This is achieved with two functions:

Description of NC commands:

NC command	Description
G64	Continuous-path mode without corner smoothing (recommended for default setting with DYNNORM)
G642	Continuous-path mode with smoothing in compliance with defined tolerances
G645	Continuous-path mode with smoothing of tangential block transitions. Smoothing blocks are also generated on tangential block transitions if the curvature of the original contour exhibits a jump in at least one axis, otherwise same as G642. **Note:** Starting with SW 2.6 **only G645** is used in CYCLE832.

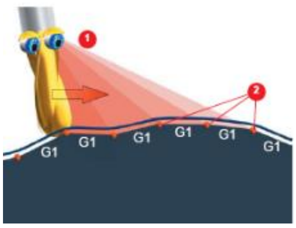
- **Look ahead - look-ahead speed control**

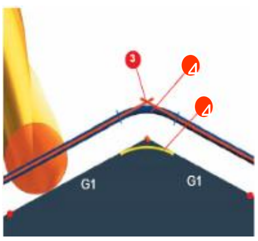
  The control system calculates several CNC blocks ahead **(1)** and determines a modal speed profile. The way in which This speed control is calculated can be set with functions G64, etc.

- **Programmable corner smoothing**

  The look ahead function also means, that the control system is able to round the corners **(2)** it detects. The programmed Corner points are therefore not approached exactly. Sharp corners are rounded. To round sharp corners **(3)**, the continuous-path commands **G642** and **G645** form transition elements **(4)** at the block boundaries. The continuous-path commands differ in the way they form these transition elements.

  With **G645**, rounding blocks are also generated on tangential block transitions if the curvature of the original contour exhibits a jump in at least one axis.





Notes

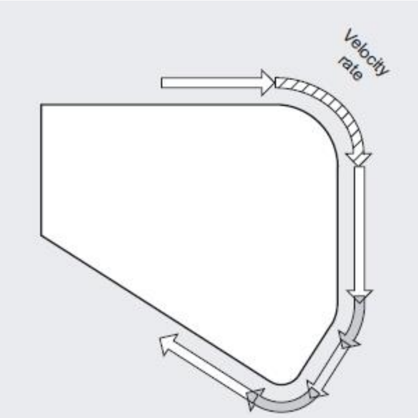
Continuous-path mode, G64
Without corner smoothing

Explanation of Function:

In continuous-path mode, the tool travels across tangential contour transitions with a path velocity as constant as possible (no deceleration at block boundaries).

Look Ahead deceleration takes place before corners (G9) and blocks with exact stop ("Look Ahead", see pages 39-40). 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 overload factor.

In comparison to continuous path mode G641-G645 contour corner points are not rounded with G64.



Activation/deactivation

Continuous-path mode with a reduction in speed according to the overload factor can be activated in any NC part program block by the modal command **G64**.

Selecting the exact stop with **G9** which works on a block-by-block basis enables rounding to be interrupted.

Continuous-path mode **G64** can be deactivated by selecting:

- Modal exact stop G60
- Rounding G642 or G645

Note:

The extent of smoothing of the contour transitions depends on the feedrate and the overload factor. The overload factor can be set in ***MD32310 $MA_MAX_ACCEL_OVL_FACTOR***

Notes

Continuous-path mode G642
Smoothing in compliance with defined tolerances

Explanation of Function:

In continuous-path mode involving rounding in compliance with defined tolerances, the rounding normally takes place while adhering to the maximum permissible path deviation.
Instead of these axis-specific tolerances, the maintenance of the maximum contour deviation (contour tolerance) or the maximum angular deviation of the tool orientation (orientation tolerance) can be configured.

Activation/deactivation

Continuous-path mode with rounding in compliance with defined tolerances can be activated in any NC part program block by the modal command **G642**.
Selecting the exact stop which works on a block-by-block basis enables rounding to be interrupted **G9**.

Continuous-path mode with rounding in compliance with defined tolerances G642 can be deactivated by selecting:

- Modal exact stop G60
- Continuous-path mode G64 or G645

Parameterisation of maximum path deviation

The maximum path deviation permitted with G642 is set for each axis in the machine data:
MD33100 $MA_COMPRESS_POS_TOL

Contour tolerance and orientation tolerance

The contour tolerance and orientation tolerance are set in the channel-specific setting data:
SD42465 $SC_SMOOTH_CONTUR_TOL (maximum contour deviation)
SD42466 $SC_SMOOTH_ORI_TOL (max. angular orientation deviation)

The settings data can be programmed in the NC program and can in this way be specified differently for each block transition.

Note:

Expansion to include contour and orientation tolerance is only supported on systems featuring the "**Polynomial interpolation**" option. When rounding in compliance with the orientation tolerance,
the additional option of "**orientation transformation**" is also necessary.

The setting data ***SD42466 $SC_SMOOTH_ORI_TOL*** is effective only with active “orientation transformation” (TRAORI).

Notes

Continuous-path mode G645 (recommended together with Advanced Surface II SW2.7
Rounding of tangential block transitions

Explanation of Function:

In continuous-path mode with rounding, rounding blocks are also only generated on tangential block transitions if the curvature of the original contour exhibits a jump in at least one axis.

The rounding movement is here defined so that the acceleration of all axes involved remains smooth (no jumps) and the parameterized maximum deviations from the original contour (***MD33120 $MA_PATH_TRANS_ POS_ TOL***) are not exceeded.

Activation/deactivation

Continuous-path mode with rounding of tangential block transitions can be activated in any NC part program block by the modal command **G645**.
Selecting the exact stop which works on a block-by-block basis enables rounding to be interrupted with **G9**.

Continuous-path mode with rounding of tangential block transitions (G645) can be deactivated by selecting:

- Modal exact stop G60
- Continuous-path mode G64 or G642

Parameterisation of maximum path deviation

The following machine data indicates the maximum permissible path deviation for each axis during rounding with G645:
MD33120 $MA_PATH_TRANS_POS_TOL

This value is only of relevance to tangential block transitions with variable acceleration. When angular, non-tangential block transitions are rounded, (as G642) the tolerance from axis machine data ***MD33100 $MA_COMPRESS_POS_TOL*** or from the programmed value in **CTOL** becomes effective.

Comparison between G642 and G645

When rounding with G642, the only block transitions rounded are those which form a corner, i.e. the velocity of at least one axis jumps. However, if a block transition is tangential, but there is a jump in the curvature, no rounding block is inserted with G642.

If this block transition is traversed with finite velocity, the axes experience some degree of jump in acceleration which (with the jerk limit activated!) may not exceed the parameterized limit set in the machine data ***MD32432 $MA_PATH_TRANS_JERK_LIM***.

Depending on the level of the limit, the path velocity at the block transition may be greatly reduced as a result. This constraint is avoided by using G645 because the rounding movement is defined here in such a way that no jumps occur in acceleration.

In the case of angular, non-tangential block transitions, the rounding behaviour is the same as with G642.

Notes

Jerk limitation and feed forward control SOFT, FFWON

General explanation:

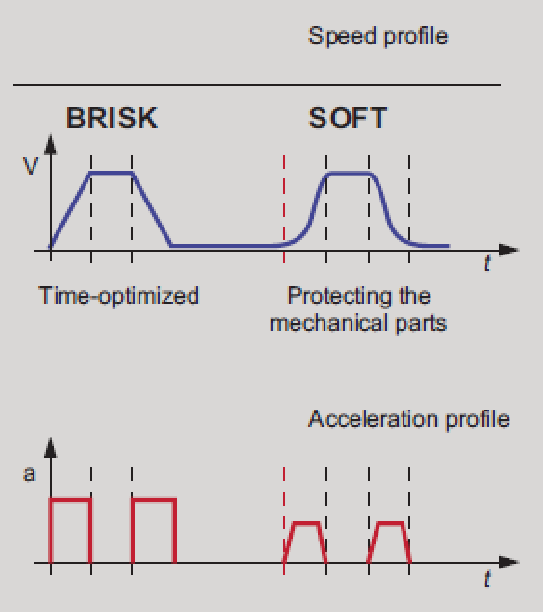
Feed forward control and jerk limitation are activated in CUST_832.SPF in a combination of the two functions. This is because this combination offers ideal conditions for freeform surface milling. Both functions can of course also be programmed separately.

Description of NC commands:

Jerk limitation function SOFT, BRISK:

NC command	Description
FFWON	Feed forward on (recommended)
FFWOF	Feed forward off
BRISK	Without jerk limitation Abrupt acceleration of path axes
SOFT	With jerk limitation (recommended) Soft acceleration of path axes. Axial acceleration, axial jerk limitation and maximum axial jerk set in machine data: ***MD32300 $MA_MAX_AX_ACCEL[0-4,AX]*** ***MD32400 $MA_AX_JERK_ENABLE[AX]*** ***MD32431 $MA_MAX_AX_JERK[AX]***

To make acceleration as gentle on the machine as possible, the acceleration profile of the axes can be influenced by means of the commands SOFT and BRISK.
If **SOFT** is activated, the acceleration behavior does not change abruptly but is increased by a linear characteristic. This reduces the load on the machine. It also has a beneficial effect on the surface quality of workpieces, since machine resonance is excited far less frequently.



Acceleration behavior with SOFT:
Acceleration behavior: soft acceleration of path axes.
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.

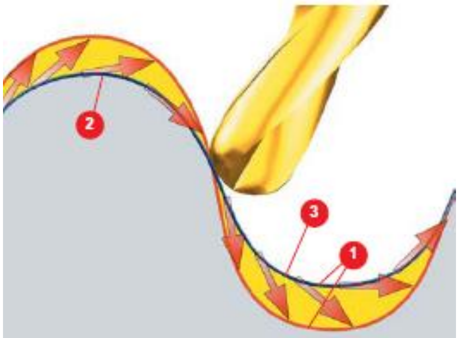
Acceleration behavior with BRISK:
Acceleration behavior: abrupt acceleration of the path axes according to the specified machine data. The axis slides travel with maximum acceleration until the feedrate is reached. BRISK enables time-optimized machining, but with jumps in the acceleration curve.

Notes

Feed forward control function FFWON:

The FFWON feed forward control function brings the velocity-dependent following error down toward zero during path traversal. Traversing with feed forward control permits higher path accuracy and thus improved machining results.

In cases where axes are not feed forward-controlled, the following error results in a contour error **(1)**, whose severity is determined by the inertia in the system. The cutter tends to leave the set point contour **(2)** tangentially, i.e. the actual contour **(3)** that is produced, deviates from the set point contour. This will manifest itself in the form of a narrowing of the radius on curved contours. The following error depends on the servo gain factor that is set (dependent on mechanics) and the axis velocity.



Following combinations are possible:

- **FFWON SOFT:** The emphasis is on high path accuracy. This is achieved by a soft speed control which is largely free from following errors (recommended setting).
- **FFWOF SOFT:** High path accuracy is not a priority. Additional rounding is achieved by means of following errors
- **FFWON BRISK:** not useful
- **FFWOF BRISK:** not recommended

Note:

The feed forward control for compensation of the following error is machine dependent and stands in direct correlation with the KV factor (Commissioning of the machine). Recommended setting for CYCLE832 is **FFWON** in combination with **SOFT**. If feed forward control is not required for the particular machine tool, then a modification must be made in the manufacturer Cycle CUST_832.SPF.

Notes

Path reference UPATH / SPATH

Explanation of Function:

During polynomial interpolation the user may require two different relationships between the velocity-determining FGROUP axes and the other path axes. The axes not contained in FGROUP are to be controlled:

- either synchronized with the path of the FGROUP axes
- or synchronized with the curve parameter.

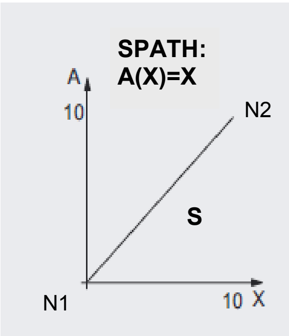
Therefore, for the axes not contained in FGROUP there are two ways to follow the path:

- **SPATH** travel synchronized with path S
- **UPATH** synchronized with the curve parameter U of FGROUP

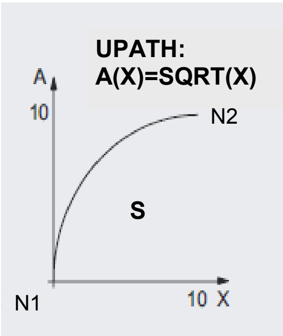
Both types of path interpolation are used in different applications and can be switched via G-codes **SPATH** and **UPATH**.

- With **SPATH** the path reference for FGROUP- axes is path length
- With **UPATH** the path reference for FGROUP- axes is curve parameter

Example:
The following picture shows the different geometrical relationship between the axes at SPATH and UPATH. Parameterization of the orientation axis equals to the path axes X,Y,Z



N10 G1 X0 A0 F1000
SPATH
N20 POLY PO[X]=(10, 10)
A10



N10 G1 X0 A0 F1000
UPATH
N20 POLY PO[X]=(10, 10)
A10

For UPATH in Block 20, the path S of the FGROUP depends on the square root of the curve parameter U: A(X)=SQRT(X)**.** According to this parameter the result are various positions for the synchronized axes A along the X-axis traversal with active UPATH in comparison to SPATH.

Note:

UPATH is recommended for 5-axis machining with active 5-axis transformation (TRAORI) and is preset in the manufacturer cycle CUST_832.

Notes

Control of pre-processing buffer FIFOCTRL

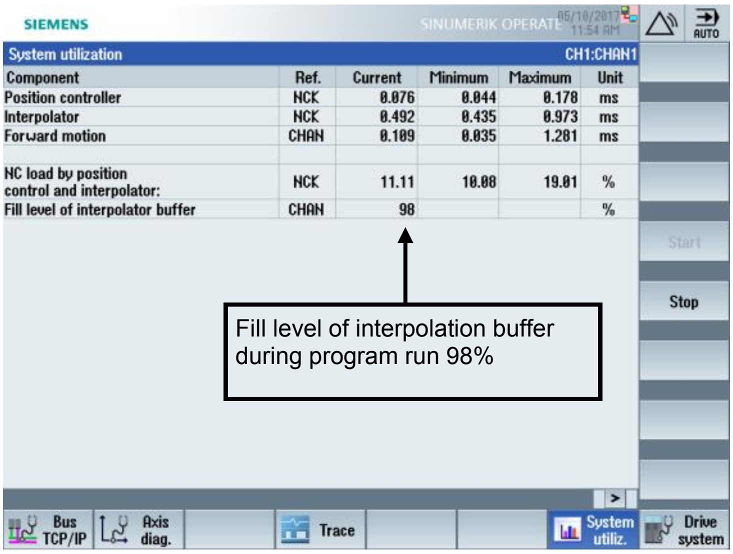
Explanation of Function:

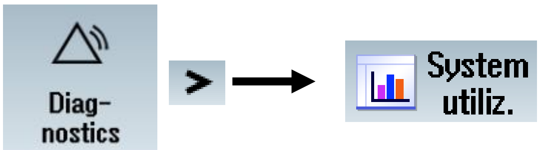
Mould & die programs are in generally executed directly from the hard disk or a external storage device (e.g. USB-stick or CF-card). The maximum number of NC-blocks that can be loaded into the NC-memory is settable in the machine data.

During machining of large NC-Files for example freeform surfaces, it is possible for the pre-processing buffer to run low. This can cause hesitations during machining or even a complete stop.

Because of this exact reason the function **FIFOCTRL** is being used, to avoid the pre-processing buffer from running low. For NC-programs with very small point to point distances, it is necessary to keep the "**Fill level of interpolation buffer"** as high as possible, in order to avoid the buffer from running low towards 0% (interpolation-crash).

The fill level of the interpolation buffer is displayed in the following area:





Note:

CYCLE832 by means CUST_832.SPF activates the pre-processing buffer **FIFOCTRL**.

Notes

Programmable smoothing tolerance CTOL /OTOL

General explanation:

The contour smoothing tolerance (CTOL) is calculated internally in CYCLE832 based on the CAM tolerance (**CTOL = square root(3) * CAM tolerance value**) and is passed to the CNC through the NC-variable CTOL. The orientation tolerance for rotary axes (OTOL) is automatically calculated in the cycle. (**OTOL = CTOL * Factor**). For 5-axis machining with active 5-axis transformation (TRAORI) this tolerance value is then passed through the NC-variable OTOL to the CNC. The factor is settable for each technology group in the channel specific setting data of the G-code group 59.

MD	Name (G-code group 59)	Default value
SD55440	$SCS_MILL_TOL_FACTOR_NORM	10
SD55441	$SCS_MILL_TOL_FACTOR_ROUGH	10
SD55442	$SCS_MILL_TOL_FACTOR_SEMIFIN	10
SD55443	$SCS_MILL_TOL_FACTOR_FINISH	10

The commands CTOL (Chord tolerance) and OTOL (Orientation tolerance) can also be used, if the tolerance is to be programmed without the support of CYCLE832.

In this case the tolerance value for the linear axes and rotary axes is then directly passed to the CNC in the program with the command “CTOL=<Value>“ and “OTOL=<Value>“.

Since OTOL also has an effect on ORISON by default, it is important when activating ORISON to ensure that the orientation tolerance is not too great, because otherwise too much orientation smoothing will result. In most cases, over-smoothing occurs during roughing operations.

The following NC commands are available to overwrite the parameterized values:
*CTOL and OTOL have priority over ATOL

The CTOL, OTOL, and ATOL commands can be used to adapt the machining tolerances defined for the compressor func-

Command	Syntax	Meaning
CTOL	CTOL=<Value>	Contour tolerance
OTOL	OTOL=<Value>	Orientation tolerance
ATOL*	ATOL[<Axis>]=<Value>	Axis specific tolerance

tions (COMPCAD), the smoothing types G645, and the orientation ORISON using machine and setting data in the NC program.

Note:

If setting data **SD55441-SD55443** have a value of 0, then there is no orientation tolerance (OTOL= -1) programmed when calling CYCLE832 with the selection of a machining type.

Whether the orientation tolerance (OTOL) is evaluated as factor of the tolerance CTOL or can be programmed directly, depends on ***SD55220$SCS_FUNCTION_MASK_MILL_TOL_SET*** (Bit1) .

Notes

Orientation smooting characteristics ORISON

General explanation:

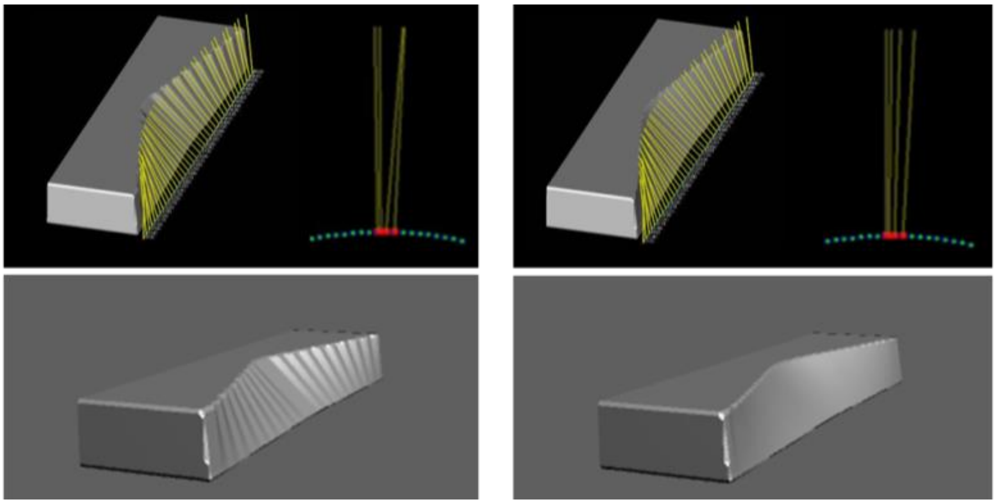
The ORISON function can be used to smooth fluctuating orientation across several blocks. The aim is to achieve a smooth characteristic for both the orientation and the contour, thus achieving a more harmonious movement of the axes.

For 5-axis programs generated by CAD/CAM systems, in which the milling paths and direction vectors for the tool are defined, the programs generally contain minimal inconsistencies in the tool alignment. Even if these deviations are only minimal, they will lead to compensatory movements in the linear axes, which will manifest themselves in slowed movements or even stopping in the path. The consequences are visible traces on the workpiece surface and longer machining time.

With ORISON the orientation is smoothed independently of the contour and greater tolerances can thus be used in the rotary axes. This results in greater machining velocities or shorter machining times, because the rotary axes are braked less due to the tolerance presets.

NC command	Description
ORISON	ORISON = **Ori**entation **S**moothing **ON** Activates the vector orientation smoothing characteristics (modal).
ORISOF	ORISOF = **Ori**entation **S**moothing **OF** Deactivates the vector orientation smoothing characteristics

Without orientation smoothing | **With orientation smoothing**



Note:

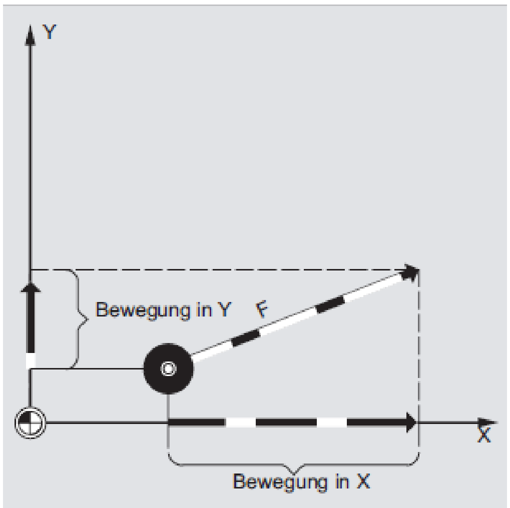
ORISON orientation smoothing is not part of CYCLE832 and it must therefore be programmed separately with the desired vector smoothing in the NC program in the case of a 5-axis program with orientation interpolation. If the ORISON function is to be automatically activated with active 5-axis transformation, then this setting can be made in CUST_832 by the OEM. The NC command ORISON is activated depending on the orientation tolerance in the manufacturer's cycle CUST_832.

Notes

Feedrate control F, FL, FGROUP, FGREF

Feed (F) for path axes (X, Y, Z):

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 tool.



The following syntax is possible:

F100 or F 100
F.5
F=2*FEED

General explanation:

NC command	Description
G93	Inverse-time feedrate. Path feedrate for rotary axes participating in movement specified in rev/min.
G94	Path feedrate on axes participating in movement specified in mm/min, inch/min, deg/min.
G95	Revolutional feedrate (in mm/revolution or inch/revolution). G95 refers to revolutions of the master spindle (as a rule, the milling spindle or main spindle of the turning machine).
F	Feedrate F for path axes or synchronized axes involved in the movement. Unit of G93/G94 applies.
FL[<Axis>]=<Value>	Axes to be included in the feed group calculation. Feedrate F applies to all axes programmed under FGROUP (geometry axes and rotary axes).
FGROUP(<Axis1>, <Axis2>,...)	No axis specified, the default configuration is restored.
FGROUP( )	No axis specified; the default is restored.
FGREF[<Rot. axis>]= <Ref. radius>	Reference radius of the individual rotary axes for the calculation of the tangential feedrate in mm/min.

Notes

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 (G70/G71 or G700/G710).

One F value can be programmed per NC block. The feedrate unit is defined using one of the G-commands G93/G94/G95. The feedrate F acts only on path axes and remains active until a new feedrate is programmed. Separators are permitted after the address F.

Feedrate type (G93,G94,G95)

The G commands G93, G94 and G95 are modal. In the event of switching between G93, G94 and G95 the path feedrate value has to be reprogrammed. When machining with rotary axes, the feed-rate can also be specified in degrees/min.

Feed F for synchronized axes (e.g. rotary axes A,B,C):

The feedrate programmed with address F applies to all the path axes programmed in the block, but not to synchronized axes. The synchronized axes are controlled in such a way, that they require the same time for their path as the path axes, and all axes reach their end point at the same time.

Limit velocity for synchronized axes (FL)

The FL command can be used to program a limit velocity for synchronized axes. In the absence of a programmed FL, the rapid traverse velocity applies. FL is deselected by MD assignment.
(***MD36200 $MA_AX_VELO_LIMIT***).

Traverse path axis as synchronized axis (FGROUP)

The command FGROUP defines whether a path axis is to be traversed with path feed or as a synchronized axis. The feed programmed with F only applies to the path axes (geometry axes) programmed in the block. FGROUP can be used to include a synchronized axis (e.g. rotary axes) in the calculation of the path feed or to exclude a path axis from the calculation.

Change FGROUP

The setting made with FGROUP can be changed:

1. By reprogramming FGROUP: e.g. FGROUP(X,Y,Z)
2. By programming FGROUP without a specific axis: FGROUP()

In accordance with FGROUP(), the initial setting in the machine data applies:
Geometry axes are now once again traversed in the path axis grouping.

Notes

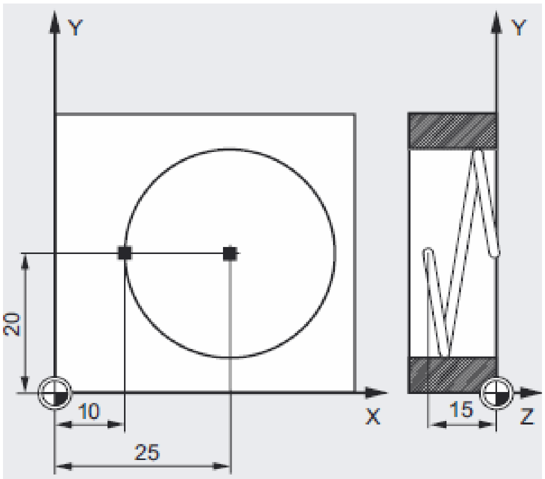
Example 1: Mode of operation of FGROUP

The following example is intended to demonstrate the effect of FGROUP on the path and path feedrate. The variable $AC_TIME contains the time of the block start in seconds. It can only be used in synchronized actions.

```
N100 G0 X0 A0
N110 FGROUP(X,A)
N120 G91 G1 G710 F100      ; Feedrate= 100mm/min or 100degrees/min
N130 DO $R1=$AC_TIME
N140 X10                   ; Feedrate= 100mm/min, path= 10mm, R1= approx. 6s
N150 DO $R2=$AC_TIME
N160 X10 A10               ; Feedrate= 100mm/min, path= 14.14mm, R2= approx. 8s
N170 DO $R3=$AC_TIME
N180 A10                   ; Feedrate= 100Grad/min, path= 10Grad, R3= approx. 6s
N190 DO $R4=$AC_TIME
N200 X0.001 A10            ; Feedrate= 100mm/min, path= 10mm, R4= approx. 6s
```

Example 2: Helical interpolation

Helical interpolation with two geometries axis X and Y that are being interpolated with the programmed feed. The infeed axis Z in this case is the synchronized axis with a limited velocity specified with FL.




```
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 FL[Z]=200 F1000    ;On the circular path, the feedrate is 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
N110 M30                                      ;End of program
```

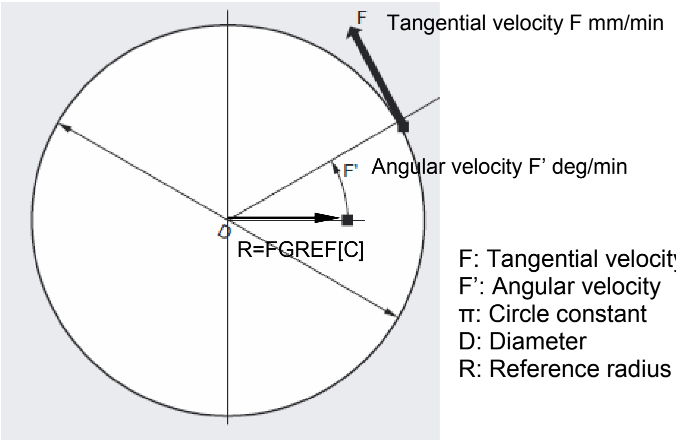
Notes

Unit for rotary and linear axes

If linear and rotary axes are associated via FGROUP, the feedrate is interpreted in the unit of measurement of the linear axis (depending on the default of G94/G95 in 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:

$$F\,[\text{mm/min}] = \frac{F'\,[\text{Grad/min}] * \pi * D}{360}$$



F: Tangential velocity
F': Angular velocity
π: Circle constant
D: Diameter
R: Reference radius

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 shown in the example, in order to be evaluated in the calculation of the path feed.

R is the reference radius of the rotary axis, and can be defined with FGREF[axis].
If no FGREF[axis] is programmed, the following reference radius applies:

In order to ensure compatibility with the behavior with no FGREF programming, the factor 1 degree = 1 mm is activated on system power up and RESET. This corresponds to a reference radius of FGREF= 360 mm/(2π) = 57.296 mm.

Special situation:

```
N10 G54 G642 G710 G90
N20 FGROUP(X,Y,Z,A)
N30 G1 G91 A10 F100
N40 G1 G91 A10 X0.0001 F100
```

In this type of programming, the value programmed in F is interpreted in N30 as the rotary axis feedrate in degree/min, whereas the feedrate evaluation in N40 is in mm/min.

Notes

Read reference radius

The value of the reference radius of a rotary axis can be read using system variables:

- In synchronized actions or with preprocessing stop in the part program via system variable:

  $AA_FGREF[<axis>] Current main run value

- Without preprocessing stop in the part program via system variable:

  $PA_FGREF[<axis>] Programmed value

If no values are programmed, the default 360 mm/(2π) = 57.296 mm (corresponding to 1 mm per degree) will be read in both variables.

For linear axes, the value in both variables is always 1 mm.

Read path axes affecting velocity

The axes involved in path interpolation can be read using system variables:

- In synchronized actions or with preprocessing stop in the part program via system variables:

$AA_FGROUP[<axis>]	Returns the value "1" if the specified axis affects the path velocity in the current main run record by means of the basic setting or through FGROUP programming. Otherwise, the variable returns the value "0".
$AC_FGROUP_MASK	Returns a bit key of the channel axes programmed with FGROUP which are to affect the path velocity.

- Without preprocessing stop in the part program via system variables:

$PA_FGROUP[<axis>]	Returns the value "1" if the specified axis affects the path velocity by means of the basic setting or through FGROUP programming. Otherwise, the variable returns the value "0".
$P_FGROUP_MASK	Returns a bit key of the channel axes programmed with FGROUP which are to affect the path velocity.

Notes

Path feedrate factor for rotary axis with FGREF

The programming of FGREF[..] is useful when machining with active 5-axis transformation (e.g. TRAORI).

In this case, in CUST_832.SPF, the variable _FGREF is pre-assigned a value of 10. This value can also be modified. In CUST_832.SPF, the value of variable _FGREF is written to the rotary axes involved in the machining, which are declared as orientation axis of a 5-axis transformation, using the FGREF[rotary axis] command.

With active G70/G700 the value of _FGREF is converted into inch before writing to the command FGREF[..]=...

The effect of FGREF depends if a change in orientation of the tool takes place through rotary axis interpolation (direct) or vector interpolation.

Direct orientation interpolation:

For direct rotary axes interpolation **without** 5-axis transformation the FGREF factors of the orientation axes are computed like for rotary axes as reference radius for the path of the axes and must
be programmed for change in rotary axes position in each NC block.
Example: G1 X.. Y.. Z.. A.. C.. FGREF[A]=.. FGREF[C]=.. F500

Vector interpolation:

For vector interpolation **with** active 5-axis transformation (TRAORI) a effective FGREF factor becomes active, which is determined by the individual FGREF factors of the geometrical average value.

FGREF[eff] = n-sqrt of [(FGREF[A] * FGREF[B] * FGREF[C])]

A: Axis identifierr of 1. orientation axis
B: Axis identifierr of 2. orientation axis
C: Axis identifierr of 3. orientation axis
n: number of orientation axis

Since there are two orientation axes for a standard 5-axis transformation e.g. A and C, the effective factor is, therefore, the root of the product of the two axial factors:

FGREF[eff] = square root of [(FGREF[A] * FGREF[C])]

It is, therefore, possible to use the effective factor for orientation axes FGREF to define a reference point on the tool to which the programmed path feedrate refers.

Example: TRAORI
ORIVECT
G1 X.. Y.. Z.. A3=.. B3=.. C3=.. F500

or

TRAORI
ORIAXES
G1 X.. Y.. Z.. A.. B.. C.. F500

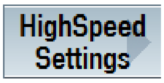
Note:

The effective FGREF Factor is automatically calculated by specifying the Parameter **_FGREF** in
the manufacturer cycle **CUST_832** (recommended value 10).

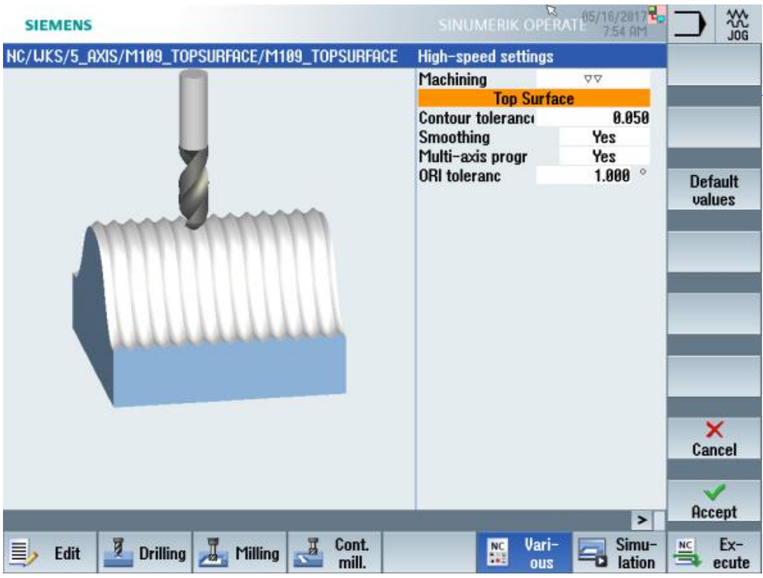
Notes

TOP Surface:

After pressing 



you can choose Top Surface.

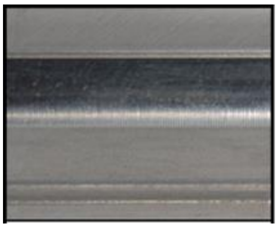


Top Surface

Top Surface is an extention of the Cycle832 with Advancend Surface.

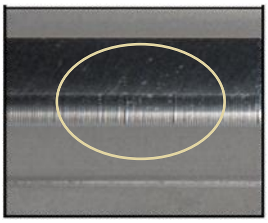
If both possibilities are avaliable in the Cycle832 you can choose between Advanced Surface or Top Surface.

Surface quality



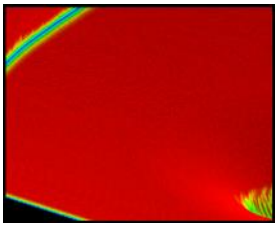
Top Surface:

Best surface qualitiy largely independent from CAM Input



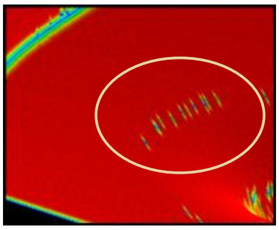
Notes

velocity



Top Surface:

Avoid unexpected stops.



.

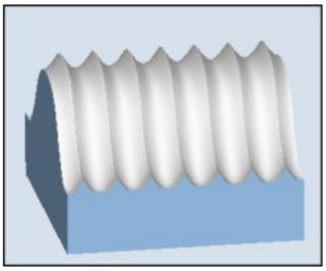
Accuracy

Top Surface: Bring to light high precision contours perfectly

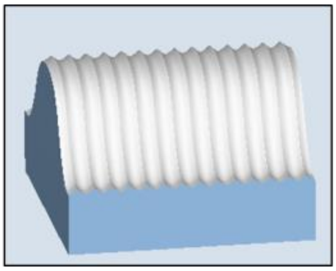
Notes

The machining operation is displayed in three different steps.

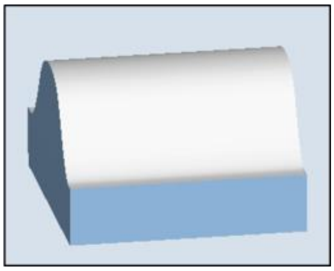
Roughing



Semifinisihing

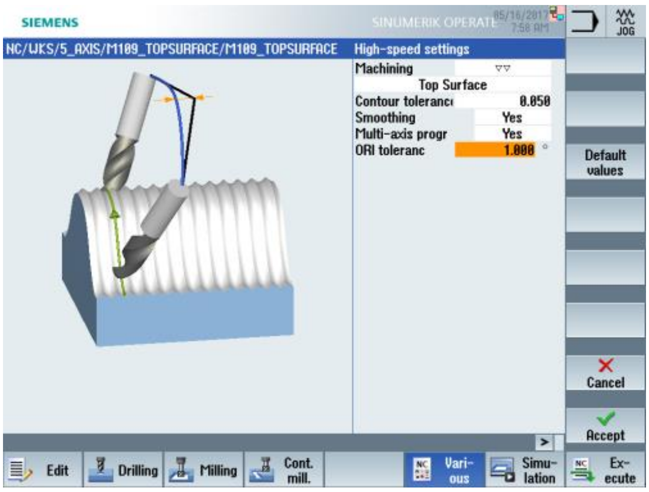


Finishing



A wave is displayed with the Top Surface option to display the tool orientation.

The orientation tolerance is displayed as follows.

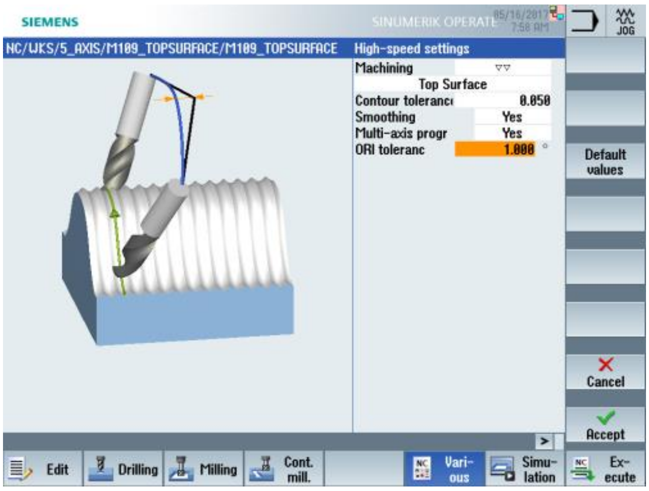


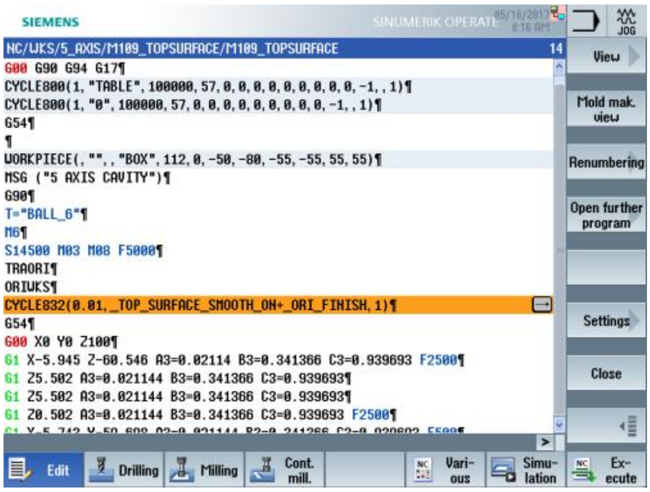
Ori - tolerance:
max. deviation form the programmed tool orientation

Notes

Extension by the smoothing functionality.

Extension by the function default values for the contour and orientation tolerance.





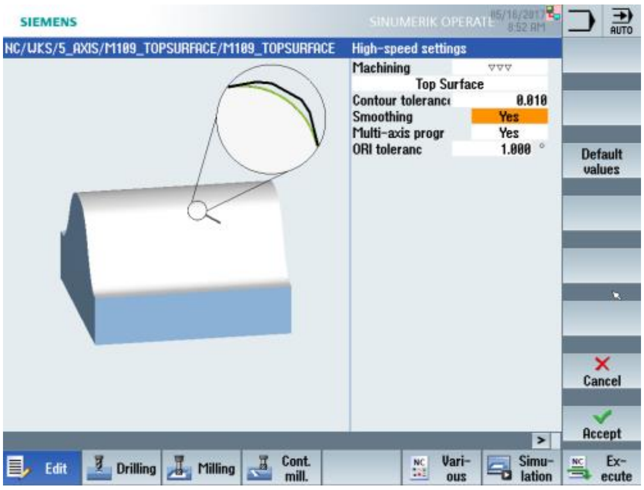
The contour tolerance is displayed by a magnifier.

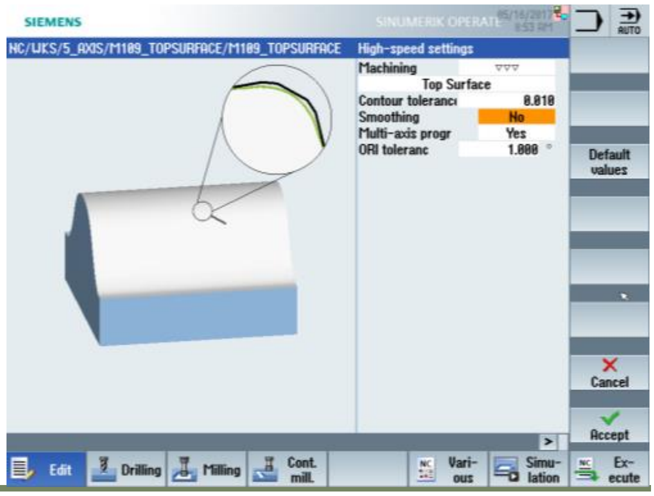
Default values for roughing 0.1 - semi finishing 0.05 - finishing 0.01

Smoothing is also displayed by a magnifier.

With smoothing the surface becomes more brilliant.

Without smoothing the tolerance is approached to the contour.





Notes

Manufacturer cycle CUST_832

Customization by machine manufacturer

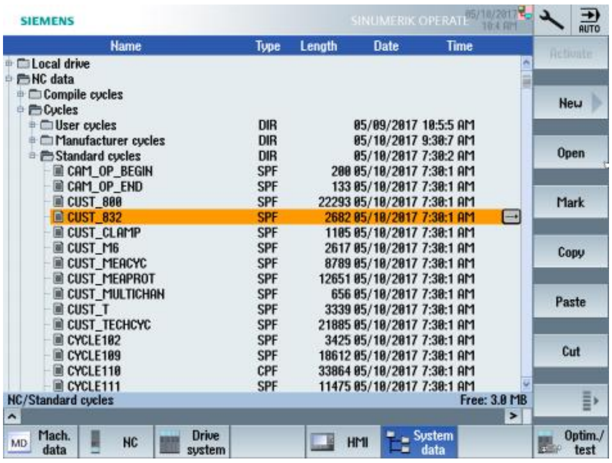
General explanation:

If the machine manufacturer requires additional responses from CYCLE 832 beyond those, which can be achieved by customizing the technology, the settings (G-commands) in CYCLE832.SPF can be modified in the manufacturer cycle CUST_832. Proceed as follows:

1. Copy cycle CUST_832.SPF from
   directory /NC-data/Cycles/Standard cycles.

2. Past the cycle CUST_832.SPF in the directory
   /NC-data/Cycles/Manufacturer cycles.

3. Open the cycle



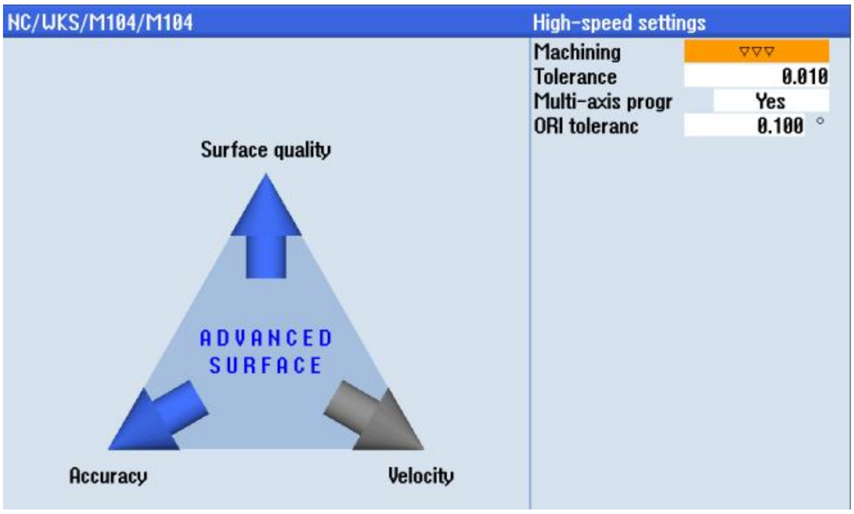
Following markers are prepared in CUST_832.SPF for each technology group, in which important NC commands are called in dependency of the selection made in CYCLE832.

_M_INIT:
_M_NORM:
_M_FINISH:
_M_SEMIFINISH:
_M_ROUGH:

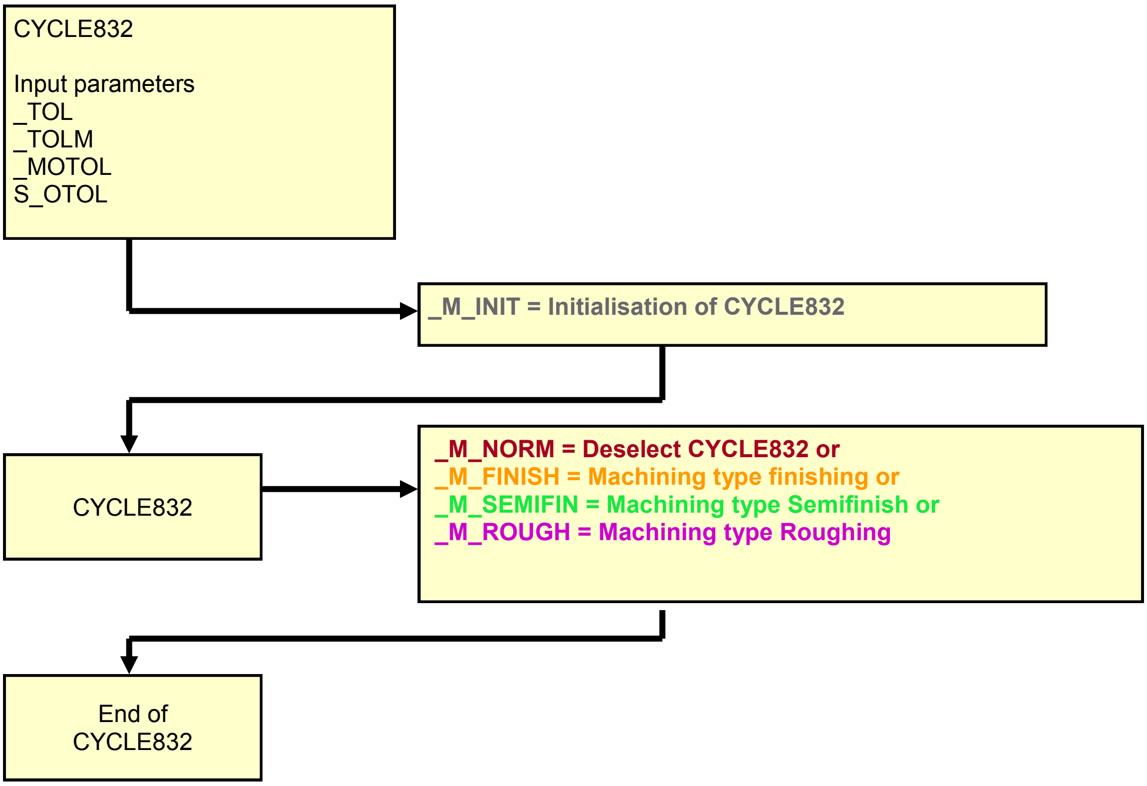
Notes

Structure of CYCLE832 and CUST_832

With the input of the Tolerance value **"_TOL"** and the selection of the machining type **"_TOLM"** (Roughing, Prefinish, Finishing, Deselect) through the input mask of CYCLE832 or postprocessor, a subsequent jump is executed into the manufacturer cycle "**CUST_832**" which activates all required G-code commands and technology group (DYNNORM, DYNROUGH, DYNSEMIFIN, DYNFINISH) for the selected machining type that the were agreed upon.



CYCLE832.SPF **CUST_832.SPF**



Notes

Printout manufacturer cycle CUST_832.SPF

```
PROC CUST_832(INT _MCASE,VAR REAL _FACTOR,VAR REAL _FGREF) SBLOF DISPLOF
;VERSION: 04.05.30.00 ;DATE: 2012-12-13
;CHANGE : 04.05.22.00 ;DATE: 2012-04-20
;customer cycle to cycle High Speed Settings (CYCLE832)
; The user can modify marks _M_INIT, and _M_NORM to _M_ROUGH
; _M_INIT       = Init CYCLE832
; _M_NORM       = Deselect CYCLE832 (_OFF)
; _M_FINISH     = Machining Finish  (_FINISH or _ORI_FINISH)
; _M_SEMIFINISH = Machining Semifinish (_SEMIFIN or _ORI_SEMIFIN)
; _M_ROUGH      = Machining Rough (_ROUGH or _ORI_ROUGH)
; Tolerance of orientation see parameter CYCLE832 S_OTOL
; or _FACTOR for orientation see setting data from SD 55441 to 55443
; Multi-axis programming see CUST_832 parameter _MOTOL 0=no 1=yes
DEF INT _MOTOL
_MCASE=ABS(_MCASE)
IF ((_MCASE _DEC2)==2)
  _MOTOL=1 _MCASE=_MCASE - 10
ENDIF
N83201 CASE _MCASE OF 0 GOTOF _M0 1 GOTOF _M1 2 GOTOF _M2 3 GOTOF _M3 4 GOTOF _M4 10 GOTOF
_M_NORM 11 GOTOF _M_FINISH 12 GOTOF _M_SEMIFINISH 13 GOTOF _M_ROUGH 14 GOTOF _M_INIT
SETAL(61019,"_MCASE or Version CUST_832 false")
RET
```

```
_M_INIT:
_FGREF=10
GOTOF _MEND
;
_M_NORM:
GOTOF _MEND
;
_M_FINISH:
SOFT
COMPCAD
G645
FIFOCTRL
UPATH
;FFWON
DYNFINISH
GOTOF _MEND
;
_M_SEMIFINISH:
SOFT
COMPCAD
G645
FIFOCTRL
UPATH
;FFWON
DYNSEMIFIN
GOTOF _MEND
;
```

Notes

```
_M_ROUGH:
SOFT
COMPCAD
G645
FIFOCTRL
UPATH
;FFWON
DYNROUGH
GOTOF _MEND
;
;------ compatible with CYC_832T and Advanced Surface
_M4:
;Parameter _FACTOR see Setting Data 55441 $SCS_MILL_TOL_FACTOR_ROUGH and following
GOTOF _MEND
;
_M0:
DYNNORM
GOTOF _MEND
;
_M1:
SOFT
;COMPCAD
;G645
;FIFOCTRL
;UPATH
;FFWON
DYNFINISH
```

```
GOTOF _MEND
;
_M2:
SOFT
;COMPCAD
;G645
;FIFOCTRL
;UPATH
;FFWON
DYNSEMIFIN
GOTOF _MEND
;
_M3:
SOFT
;COMPCAD
;G645
;FIFOCTRL
;UPATH
;FFWON
DYNROUGH
GOTOF _MEND
;
_MEND:
RET ;M17
```

Notes

Settings for „Advanced Surface“ CYCLE832

Configuration of prerequisite machine data

The subsequent list of machine data are prerequisite for the function of CYCLE832 for 3-5 axis machining and should be pre-assigned as follows.

General machine data:

MD	Name	Description	Required or recommended values	Comment
10050	$MN_SYSLOCK_CYCLE_ TIME	Basic system cycle	0.002-0.004	Hardware dependent
10070	$MN_IPO_SYSLOCK_TIME_ RATIO	Factor IPO-cycle	1-2	
10200	$MN_INT_INCR_PER_MM	Internal calculation resolution linear axis	100000 - required	10000 for export version
10210	$MN_INT_INCR_PER_DEG	Internal calculation resolution rotary axis	100000 - required	10000 for export version
10680	$MN_MIN_CONTOUR_ SAMPLING_TIME	Minimal contour sampling time	100000 - required	10000 for export version
10682	$MN_CONTOUR_ SAMPLING_FACTOR	Contour sampling factor	1	Default
18360	$MN_MM_EXT_PROG_ BUFFER_SIZE	Maximum reload memory when executing from external	500	for 840D sl
18362	$MN_MM_EXT_PROG_NUM	Number of ext. programs that can be simultaneously processed	2	for 840D sl
19330	$ON_IPO_FUNCTION_ MASK	Compressor functionality for 3 and 5 axes	Bit 8 = 1 Compressor 3 axes Bit 2 = 1 Compressor 5 axes	If only the Bit 8 = 1 is activated and a 5 axes program will be executed, only the tool path will be compressed not orientation

Notes

Channel machine data:

MD	Name	Description	Required or recommended values	Comment
20150	$MC_GCODE_RESET_ VALUE[3]	Initial setting G group 4	3	FIFOCTRL
20150	$MC_GCODE_RESET_ VALUE[20]	Initial setting G group 21	2	SOFT
20150	$MC_GCODE_RESET_ VALUE[44]	Initial setting G group 45	2	UPATH (for 5-axis machining)
20150	$MC_GCODE_RESET_ VALUE[50]	Initial setting G group 51	2	ORIAXES (for 5-axis machining)
20170	$MC_COMPRESS_BLOCK_PATH_LIMIT	Maximum length of POLYNOM generated by the compressor	20 - recommended	
20172	$MC_COMPRESS_VELO_TOL	Max. deviation of the path feed for COMPCAD	1000 -recommended	
20442	$MC_LOOKAH_SYSTEM_PARAM [0-19]	System parameter for extended Look Ahead	0 - required	default
20443	$MC_LOOKAH_FFORM[0-1]	Activate extended Look Ahead for selected technology-specific dynamic values (DYNNORM, DYNPOS)	0 - required	default
20443	$MC_LOOKAH_FFORM[2-4]	Activate extended Look Ahead for selected technology-specific dynamic values (DYNROUGH, DYNSEMIFIN, DYNFINISH)	1 - required	activates new Look Ahead Function for Advanced Surface
20450	$MC_LOOAH_RELIEVE_BLOCK_ CYCLE	Relief factor for block cycle time	0 - required	default

Notes

MD	Name	Description	Required or recommended values	Comment
20455	$MC_ LOOKAH_FUNCTION_MASK	Look Ahead special function related to safety integrated	0 - recommended	default / deactivated
20460	$MC_LOOKAH_SMOOTH_FACTOR	Smoothing factor for Look	0 - recommended	default /
20465	$MC_ADAPT_PATH_DYNAMIC[0-1]	Adaptation path dynamic	1 - recommended	default /
20470	$MC_CPREC_WITH_FFW	Programmable contour accuracy CPRECON is active if FFWON =1	3 - required	
20476	$MC_ORISON_STEP_LENGH	Path length for block division with ORISON	0.5 -required	Default
20478	$MC_ORISON_MODE	Mode of orientation smoothing	100 - required	
20480	$MC_SMOOTHING_MODE	Behaviour of smoothing	0 - recommended	default /
20482	$MC_COMPRESSOR_MODE	Compressor Mode	300 - recommended	Alternative mode =100
20484	$MC_COMPRESSOR_PERFORMANCE	Compressor behaviour	9	9 = default
20485	$MC_COMPRESS_SMOOTH_FACTOR[0-1]	Smoothing by compressor	0	Should not work
20485	$MC_COMPRESS_SMOOTH_FACTOR[2-4]	Smoothing by compressor	0.0001	Small value 0.0001
20486	$MC_COMPRESS_SPLINE_DEGREE[0-1]	Compressor Spline degree	3	default
20486	$MC_COMPRESS_SPLINE_DEGREE[2-4]	Compressor Spline degree	5	3 = def./ 5 for COMPCAD
20487	$MC_COMPRESS_SMOOTH_FACTOR_2[0-1]	Compressor smoothing rotary axis	0	default
20487	$MC_COMPRESS_SMOOTH_FACTOR_2[2-4]	Compressor smoothing rotary axis	0.5	only for 5-axis application
20490	$MC_IGNORE_OVL_FACTOR_FOR_ADIS	Influence of G642	1	
20560	$MC_G0_TOLERANCE_FACTOR	Tolerance factor for G00 with COMPCAD, G64x	3 - recommended	This factor is used to make different settings for the tolerances for processing when G00 is active

Notes

MD	Name	Description	Required or recommended values	Comment
20600	$MC_MAX_PATH_JERK[0-1]	Path jerk		
20600	$MC_MAX_PATH_JERK[2-4]	Path jerk	10000 - recommended	Should not work
20602	$MC_CURV_EFFECT_ON_PATH_ACCEL[0-1]	Ratio, translatory to centripetal acceleration for selected	0 - recommended	Deactivated
20602	$MC_CURV_EFFECT_ON_PATH_ACCEL[2]	Ratio, translatory to centripetal acceleration for selected technology-specific dynamic values DYNROUGH	0.65 - recommended depends on machine mechanic	Must work so that with active curvature smoothing, the jerk is not too excesive!
20602	$MC_CURV_EFFECT_ON_PATH_ACCEL[3]	Ratio, translatory to centripetal acceleration for selected technology-specific dynamic values DYNSEMIFIN	0.6 - recommended depends on machine mechanic	
20602	$MC_CURV_EFFECT_ON_PATH_ACCEL[4]	Ratio, translatory to centripetal acceleration for selected technology-specific dynamic values DYNFINISH	0.5 - recommended depends on machine mechanic	Limit centripetal acceleration especially for "large" machines!
20603	$MC_CURV_EFFECT_ON_PATH_JERK[0-4]	Effect path curvature on path jerk	0 - recommended	Deactivated
20605	$MC_PREPDYN_SMOOTHING_FACTOR[0-4]	Factor for curve smoothing	1 - recommended	Default
20606	$MC_PREPDYN_SMOOTHING_ON [0-1]	Activates curvature smoothing for selected technology-specific dynamic values (DYNNORM, DYNPOS)	0 - required	Default
20606	$MC_PREPDYN_SMOOTHING_ON [2-4]	Activates curvature smoothing for selected technology-specific dynamic values (DYNROUGH, DYNSEMIFIN, DYNFINISH)	1 - required	Function for Advanced Surface

Notes

MD	Name	Description	Required or recommended values	Comment
20607	$MC_PREPDYN_MAX_FILT_LENGTH_GEO[0-4]	Maximum filter length for geometry axes	2 - recommended	Default
20608	$MC_PREPDYN_MAX_FILT_LENGTH_RD[0-4]	Maximum filter length for rotary axes	5 - recommended	Default
28060	$MC_MM_IPO_BUFFER_SIZE	Memory, interpolator for number of G1 blocks	150 - recommended	
28070	$MC_MM_NUM_BLOCKS_IN_PREP	Memory, preparation (pre-processing)	80 - recommended	
28520	$MC_MM_MAX_AXISPOLY_PER_BLOCK	Maximum number of axis polynomials per block	5 - required	
28530	$MC_MM_PATH_VELO_SEGMENTS	Number of memory elements for limiting the path velocity	5 - required	
28533	$MC_MM_LOOKAH_FFORM_UNITS	Memory for extended Look Ahead	18 - required	Function for Advanced Surface
28540	$MC_MM_ARCLENGTH_SEGMENTS	Number of memory elements to display the arc length function	10 - required	
28580	$MC_MM_ORIPATH_CONFIG	Configuration for path relative orientation ORIPATH with OST, OSD	1	(for 5-axis application) only without ORISON
28590	$MC_MM_ORISON_BLOCKS	Number of blocks for orientation smoothing	100	May have to be a multiple for 5-axis machining dependent on ORISON_TOL
28610	$MC_MM_PREPDYN_BLOCKS	Memory for curvature smoothing	10 - required	
29000	$OC_LOOKAH_NUM_CHECKED_BLOCKS	Number of Look Ahead blocks (this must be the same as MD28060)	150 - recommended	Must be the same as MD 28060

Notes

MD	Name	Description	Required or recommended values	Comment
42470	$SC_CRIT_SPLINE_ANGLE	Corner limit angle for compressor - COMPCAD criteria for target point analysis	36 - required	Default
42471	$SC_MIN_CURV_RADIUS	Factor for compressor tolerance	1- recommended	
42500	$SC_SD_MAX_PATH_ACCEL	Limits the path acceleration via Setting data	10000- recommended	For 840D sl
42502	$SC_IS_SD_MAX_PATH_ACCEL	Activates path acceleration via Setting data SD	0- recommended	Deactivated
42510	$SC_SD_MAX_PATH_JERK	Limits path jerk via Setting data	10000- recommended	For 840D sl
42512	$SC_IS_SD_MAX_PATH_JERK	Activates path jerk via Setting data	0- recommended	Deactivated

42674	$SC_ORI_SMOOTH_DIST	Maximum deviation for smoothing of tool orientation with compressor with OSD	5	The value depends on the machining/ cycle for 5-axis machining
42676	$SC_ORI_SMOOTH_TOL	Tolerance for smoothing of tool orientation with OST	1	The value depends on the machining/ cycle for 5-axis machining
42678	$SC_ORISON_TOL	Tolerance for smoothing of tool orientation with ORISON	1	The value depends on the machining/ cycle for 5-axis machining OTOL takes priority! Recommendation: =0 Circumferential milling =0.1 Face milling

Notes

Axis machine data:

MD	Name	Description	Required or recom-mended values	Comment
32300	$MA_MAX_AX_ACCEL[0-3,AX]	Axis acceleration for selected technology (DYNNORM, DYPOS, DYOUGH, DYNSEMIFIN)	Depends on machine mechanic	
32300	$MA_MAX_AX_ACCEL[4,AX]	Axis acceleration for selected technology DYNFINISH	1- 3 recom-mended	Low acceleration especially for "large" machines
32310	$MA_MAX_ACCEL_OVL_FACTOR [0-4,AX]	Overload factor for axial ve-locity steps for selected tech-nology (DYNNORM, DYNPOS, DYNROUGH, DYNSEMIFIN DYNFINISH)	1.2- recom-mended	Default
32400	$MA_AX_JERK_ENABLE[AX]	Enable axes filter		Determined by optimization positioning / circle test
32402	$MA_AX_JERK_MODE[AX]	Axes filter type 1 of 3	2	same in all axis
32410	$MA_AX_JERK_TIME[AX]	Filter time or filter sequence		Determined by optimization positioning / circle test

32432	$MA_PATH_TRANS_JERK_LIM [AX]	Axis jerk on block border	= MD 32431	
32640	$MA_STIFFNESS_CONTROL_ENABLE[AX]	Dynamic stiffness control	1	Same in all axis
33100	$MA_COMPRESS_POS_TOL[AX]	Compressor smoothing toler-ance for linear axes	0.01	Value depending on ma-chining / cycle usually = 1,2 cordial deviation CAM
33120	$MA_ PATH_TRANS_POS_TOL [AX]	Maximum deviation when smoothing with G645	0.005 - rec-ommended	Same in all axis
33000	$MA_FIPO_TYP[AX]	Type of fine interpolator (only effective, if factor IPO-cycle >1)	3	Same in all axis

M104 END

Notes

M105

Sinumerik
Operate

3D Tool Radius Compensation

Module Description:

This module explains the benefits of different tool radius compensations during 5-axis milling with active 5-axis transformation based on practical application examples.

Module Objective:

With the aid of this module you get to learn the purpose and use of tool radius compensation in conjunction with 3-axis surface machining and 5-axis multi-axis surface machining.

Content:

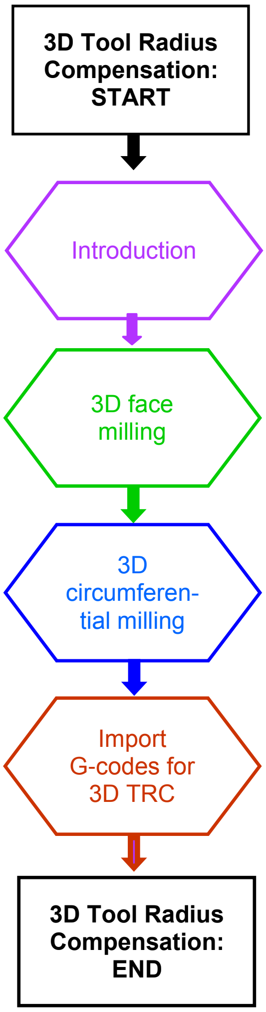
3D tool radius compensation

3D face milling

3D circumferential milling

Important G-code commands for 3D TRC

840D sl SINUMERIK Operate


3D Tool Radius Compensation: START
Introduction
3D face milling
3D circumferen-tial milling
Import G-codes for 3D TRC
3D Tool Radius Compensation: END

Notes

3D Tool radius compensation

Introduction

Tool compensation makes a CNC program independent of the tool radius. Tool radius compensation in the 2 ½ D range is a familiar concept. In the 3D range, however, especially with 5-axis milling, it can be very different.

The same programming commands apply to 3D tool radius compensation as to 2D tool radius compensation. The left/right offset is specified in the direction of motion using G41/G42.

The approach response is always controlled with NORM. The 3D radius compensation is only effective when 5-axis transformation "TRAORI" is selected.

Difference between 2 1/2 D and 3D tool radius compensation

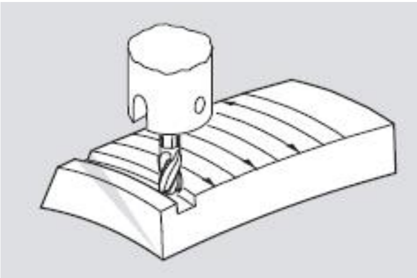
- With 21/2D TRC, it is assumed that the tool is always space-bound. Tools with constant orientation (cylindrical tools) are used for circumferential milling operations. While the orientation of the machining surface is not constant when other tools are used, it is determined by the contour and cannot thus be controlled independently of it.

- With 3D TRC, surfaces with variable orientation are generated. The prerequisite for circumferential milling is that the tool orientation can be changed, i.e. in addition to the 3 degrees of freedom needed to position the tool (normally 3 linear axes), a further two degrees of freedom (2 rotary axes) are required to set the tool orientation (5-axis machining). Face milling is possible with 3 or 5 degrees of freedom.

3D tool radius compensation can be applied for various machining operations such as:

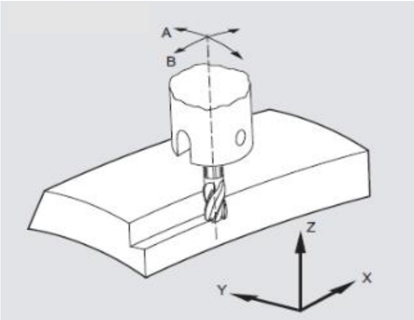
- ***Face milling of freeform surfaces***

  The tool is engaged only with the face side of a symmetrical tool (e.g. spherical ball mill or end mill with corner radius etc.). Radius compensations for various tool types can be applied in the NC-program. The standard tool can hereby deviate from the actual tool. With CUT3DF the NC can compensates for such a differential tool.



- ***Circumferential milling of freeform surfaces***

  The tool is engaged only with the side of a symmetrical tool (e.g. end mill or end mill with corner radius). Radius compensations for various tool types can be applied in the NC-program. The standard tool can hereby deviate from the actual tool. With CUT3DC for example the NC can compensates for such a differential tool.



Note:

3D Tool radius compensation is a option and requires a license in combination with the option 5-axis transformation (TRAORI).

Notes

3D Face milling

3D tool radius compensation with CUT3DFF/CUT3DF

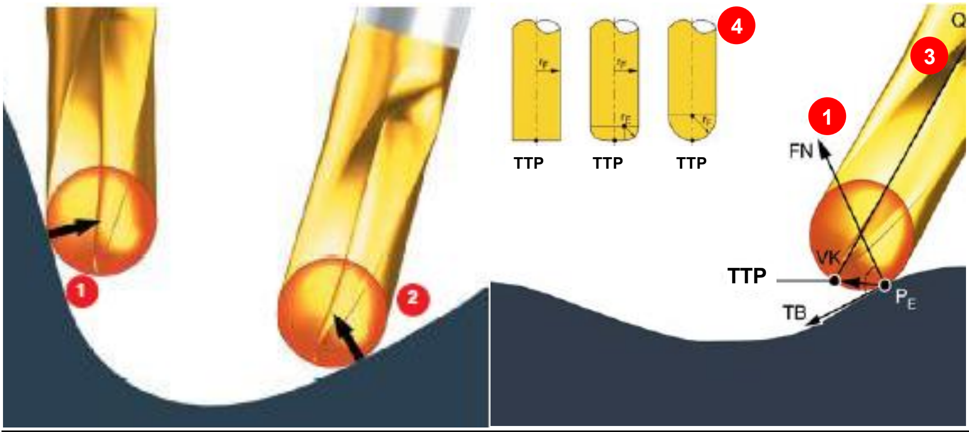
General explanation:

For radius compensation in face milling with **CUT3DFF** and **CUT3DF**, it is not enough simply to specify the cutter geometry as it is the case with 2 ½ D tool radius compensation. The compensation direction is also required. With **CUT3DF** the compensation direction is calculated from the surface normal, the tool orientation and the tool geometry. This perpendicular is known as the surface normal or surface normal vector. It is calculated from the current tool orientation and the surface normal from the workpiece surface.

Application (e.g. 5-axis face milling with e.g. spherical cutter)

For a path of a spherical cutter in the work area the compensation must be made perpendicularly to the surface on which the path is running. This means that the compensation direction **VK** is described by the surface normal vector **FN** (**1**) of the surface at the insertion point **PE**.

The CAM must provide the surface normal **FN** (**1** and **2**) and tool orientation **Q** (**3**) in every NC block. The geometry of the tool tip (tool type **4**) must be known, and is generally speaking defined in the tool parameters of the control. This information enables the control to perform radius compensation and to calculate the new tool insertion point **PE,** if tool radius compensation **CUT3DF** is enabled.



Q Tool orientation vector
FN Surface normal
PE Tool Insertion Point
TB Path tangent
VK Compensation vector
TTP Tool Tip Point

NC-command	Description
CUT3DFF	3D tool radius compensation with constant orientation for **3-axis** face milling of free form surfaces with programming of surface normal's. Tool direction in Z of the coordinate system currently defined with a frame.
CUT3DF	3D tool radius compensation with variable tool orientation for **5-axis** face milling of freeform surfaces with specification of surface normal's.

Note:

TTP (Tool Tip Point) ist die bevorzugte Programmier-Variante.

Notes

Tool shapes and tool data with CUT3DFF/CUT3DF

The tool list on the following pages gives an overview of the tool shapes, which may be used in face milling operations, as well as tool data limit values.

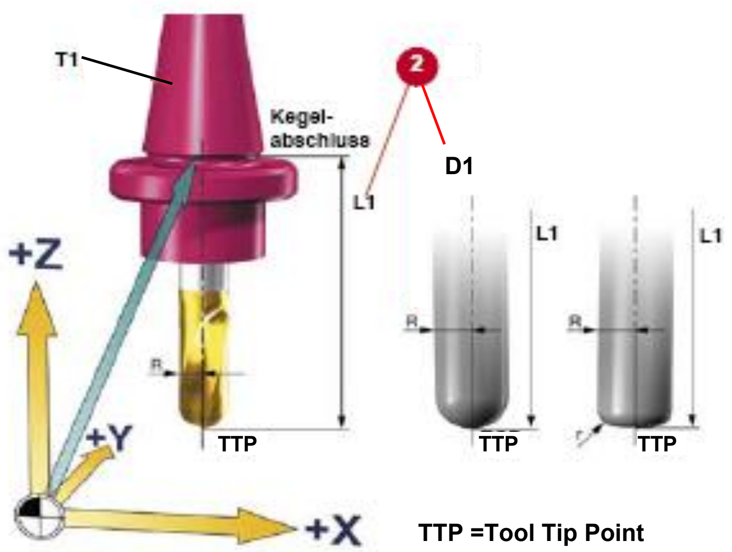
Milling cutter name	WKZ-Typ Nr.	R	r	a
Ball nose end mill	110	>0	X	X
Conical ball endmill	111	>0	>d	X
Endmill without corner rounding	120, 130	>0	X	X
Endmill with corner rounding (Toroid)	121, 131	>r	>0	X
Tapered cutter without corner rounding	155	>0	X	>0
Tapered cutter with corner rounding	156	>r	>0	>0
Taperd die-sink. cutter	157	>0	X	>0
X= not evaluated, R=shank radius or Geo. radius, r=corner radius, a=taper angle				

If a different type number is used in the NC program than the ones shown in the list, the system automatically uses tool type 110 (ball nose). An alarm is output if the tool data limit values are violated. Toll parameters marked with X in the tool list are not evaluated.

The shape of the tool shaft is not taken into account. The effect of tool types 120 and 156 is identical.
If, in the NC program, a type number is specified that differs from that in the diagram, then the system automatically uses tool type 110 (cylindrical die-sinking milling tool).

Tool length offset and cutting edge

The tool tip (TTP) is in general the reference point for length offset (intersection longitudinal axis/surface). The tool is called with the tool identification number (e.g. T1) with specification of the cutting edge number (D1) in the NC program. On tool change the Length L1 and cutting edge D1 is as
default automatically active (2).



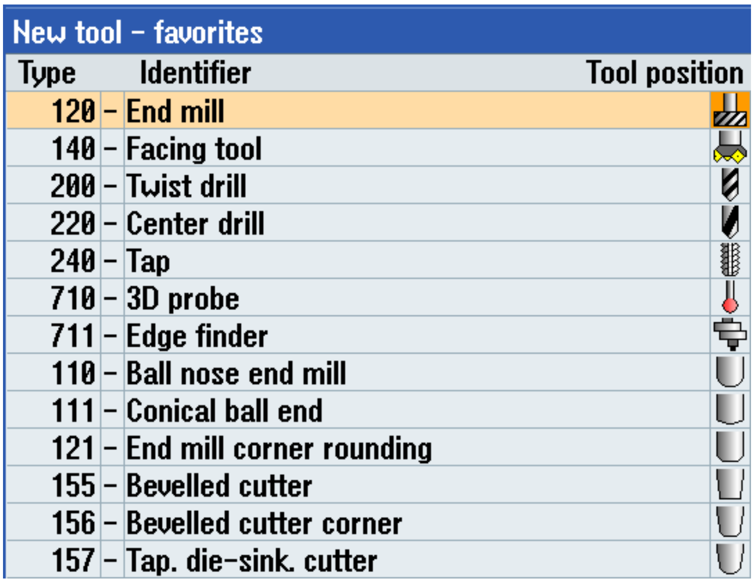
TTP =Tool Tip Point

Notes

3D tool radius compensation at tool change

After deselection of the radius compensation (G40), approach must be made with G41 or G42 again in a linear block after a tool change.

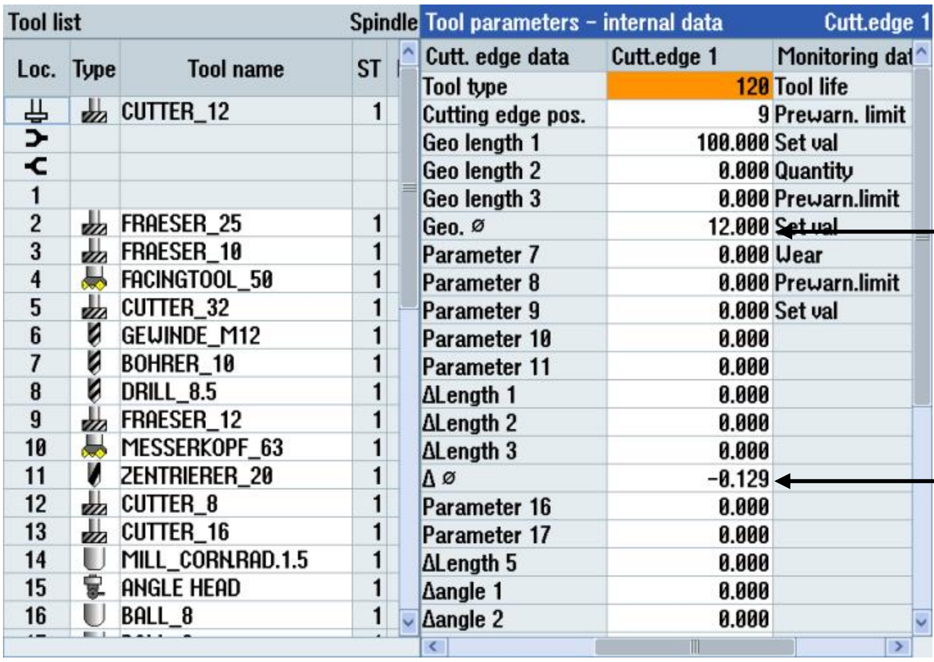
Milling tools and type designations (HMI):



New tool – favorites

Type	Identifier	Tool position
120 –	End mill	
140 –	Facing tool	
200 –	Twist drill	
220 –	Center drill	
240 –	Tap	
710 –	3D probe	
711 –	Edge finder	
110 –	Ball nose end mill	
111 –	Conical ball end	
121 –	End mill corner rounding	
155 –	Bevelled cutter	
156 –	Bevelled cutter corner	
157 –	Tap. die-sink. cutter	

Example tool list details (HMI): Tool type 120 (end mill)

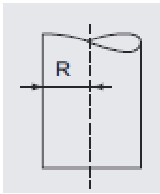


Tool list — Spindle

Loc.	Type	Tool name	ST
		CUTTER_12	1
1			
2		FRAESER_25	1
3		FRAESER_10	1
4		FACINGTOOL_50	1
5		CUTTER_32	1
6		GEWINDE_M12	1
7		BOHRER_10	1
8		DRILL_8.5	1
9		FRAESER_12	1
10		MESSERKOPF_63	1
11		ZENTRIERER_20	1
12		CUTTER_8	1
13		CUTTER_16	1
14		MILL_CORN.RAD.1.5	1
15		ANGLE HEAD	1
16		BALL_8	1

Tool parameters – internal data — Cutt.edge 1

Cutt. edge data	Cutt.edge 1	Monitoring dat
Tool type	120	Tool life
Cutting edge pos.	9	Prewarn. limit
Geo length 1	100.000	Set val
Geo length 2	0.000	Quantity
Geo length 3	0.000	Prewarn.limit
Geo. ⌀	12.000	Set val
Parameter 7	0.000	Wear
Parameter 8	0.000	Prewarn.limit
Parameter 9	0.000	Set val
Parameter 10	0.000	
Parameter 11	0.000	
ΔLength 1	0.000	
ΔLength 2	0.000	
ΔLength 3	0.000	
Δ ⌀	-0.129	
Parameter 16	0.000	
Parameter 17	0.000	
ΔLength 5	0.000	
Δangle 1	0.000	
Δangle 2	0.000	



D = geo diameter

ΔD = wear diameter

Notes

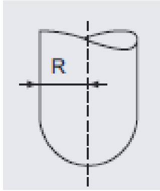
Example tool list details (HMI): Tool type 110

Tool list — Spindle

Loc.	Type	Tool name	ST
		BALL_END_CYL	1
1		CUTTER_12	1
2		FRAESER_25	1
3		FRAESER_10	1
4		FACINGTOOL_50	1
5		CUTTER_32	1
6		GEWINDE_M12	1
7		BOHRER_10	1
8		DRILL_8.5	1
9		FRAESER_12	1
10		MESSERKOPF_63	1
11		ZENTRIERER_20	1
12		CUTTER_8	1
13		CUTTER_16	1
14		MILL_CORNRAD.1.5	1
15		ANGLE HEAD	1
16		BALL_8	1

Tool parameters - internal data — Cutt.edge 1

Cutt. edge data	Cutt.edge 1	Monitoring dat
Tool type	110	Tool life
Cutting edge pos.	9	Prewarn. limit
Geo length 1	100.000	Set val
Geo length 2	0.000	Quantity
Geo length 3	0.000	Prewarn.limit
Geo. ⌀	6.000	Set val
Parameter 7	0.000	Wear
Parameter 8	0.000	Prewarn.limit
Parameter 9	0.000	Set val
Parameter 10	0.000	
Parameter 11	0.000	
ΔLength 1	0.000	
ΔLength 2	0.000	
ΔLength 3	0.000	
Δ ⌀	0.000	
Parameter 16	0.000	
Parameter 17	0.000	
ΔLength 5	0.000	
Δangle 1	0.000	
Δangle 2	0.000	



D = tool diameter

ΔD = wear diameter

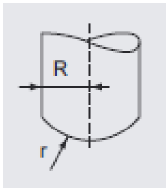
Example tool list details (HMI): Tool type 111

Tool list — Spindle

Loc.	Type	Tool name	ST
		BALL_END_CON	1
1		CUTTER_12	1
2		FRAESER_25	1
3		FRAESER_10	1
4		FACINGTOOL_50	1
5		CUTTER_32	1
6		GEWINDE_M12	1
7		BOHRER_10	1
8		DRILL_8.5	1
9		FRAESER_12	1
10		MESSERKOPF_63	1
11		ZENTRIERER_20	1
12		CUTTER_8	1
13		CUTTER_16	1
14		MILL_CORNRAD.1.5	1
15		ANGLE HEAD	1
16		BALL_8	1

Tool parameters - internal data — Cutt.edge 1

Cutt. edge data	Cutt.edge 1	Monitoring dat
Tool type	111	Tool life
Cutting edge pos.	9	Prewarn. limit
Geo length 1	100.000	Set val
Geo length 2	0.000	Quantity
Geo length 3	0.000	Prewarn.limit
Geo. ⌀	8.000	Set val
Parameter 7	5.000	Wear
Parameter 8	0.000	Prewarn.limit
Parameter 9	0.000	Set val
Parameter 10	0.000	
Parameter 11	0.000	
ΔLength 1	0.000	
ΔLength 2	0.000	
ΔLength 3	0.000	
Δ ⌀	0.000	
Parameter 16	0.000	
Parameter 17	0.000	
ΔLength 5	0.000	
Δangle 1	0.000	
Δangle 2	0.000	



D = tool diameter

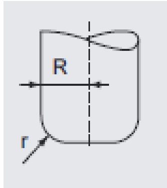
r = corner radius

ΔD = wear diameter

Δr = wear corner radius

Notes

Example tool list details (HMI): Tool type 121



Tool list			Spindle
Loc.	Type	Tool name	ST
		MILL_CRAD_D16_R3	1
1		CUTTER_12	1
2		FRAESER_25	1
3		FRAESER_10	1
4		FACINGTOOL_50	1
5		CUTTER_32	1
6		GEWINDE_M12	1
7		BOHRER_10	1
8		DRILL_8.5	1
9		FRAESER_12	1
10		MESSERKOPF_63	1
11		ZENTRIERER_20	1
12		CUTTER_8	1
13		CUTTER_16	1
14		MILL_CORN.RAD.1.5	1
15		ANGLE HEAD	1
16		BALL_8	1

Tool parameters – internal data		Cutt.edge 1
Cutt. edge data	Cutt.edge 1	Monitoring dat
Tool type	121	Tool life
Cutting edge pos.	9	Prewarn. limit
Geo length 1	100.000	Set val
Geo length 2	0.000	Quantity
Geo length 3	0.000	Prewarn.limit
Geo. ø	16.000	Set val
Parameter 7	3.000	Wear
Parameter 8	0.000	Prewarn.limit
Parameter 9	0.000	Set val
Parameter 10	0.000	
Parameter 11	0.000	
ΔLength 1	0.000	
ΔLength 2	0.000	
ΔLength 3	0.000	
Δ ø	0.000	
Parameter 16	0.000	
Parameter 17	0.000	
ΔLength 5	0.000	
Δangle 1	0.000	
Δangle 2	0.000	

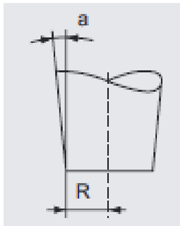
D = tool diameter

r = corner radius

ΔD = wear diameter

Δr = wear corner radius

Example tool list details (HMI): Tool type 155



Tool list			Spindle
Loc.	Type	Tool name	ST
		MILL_TAPER	1
1		CUTTER_12	1
2		FRAESER_25	1
3		FRAESER_10	1
4		FACINGTOOL_50	1
5		CUTTER_32	1
6		GEWINDE_M12	1
7		BOHRER_10	1
8		DRILL_8.5	1
9		FRAESER_12	1
10		MESSERKOPF_63	1
11		ZENTRIERER_20	1
12		CUTTER_8	1
13		CUTTER_16	1
14		MILL_CORN.RAD.1.5	1
15		ANGLE HEAD	1
16		BALL_8	1

Tool parameters – internal data		Cutt.edge
Cutt. edge data	Cutt.edge 1	Monitoring dat
Tool type	155	Tool life
Cutting edge pos.	9	Prewarn. limit
Geo length 1	100.000	Set val
Geo length 2	0.000	Quantity
Geo length 3	0.000	Prewarn.limit
Geo. ø	8.000	Set val
Parameter 7	0.000	Wear
Parameter 8	0.000	Prewarn.limit
Parameter 9	0.000	Set val
Parameter 10	0.000	
Parameter 11	10.000	
ΔLength 1	0.000	
ΔLength 2	0.000	
ΔLength 3	0.000	
Δ ø	0.000	
Parameter 16	0.000	
Parameter 17	0.000	
ΔLength 5	0.000	
Δangle 1	0.000	
Δangle 2	0.000	

D = tool diameter

a = taper angle

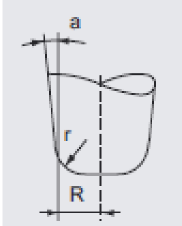
ΔD = wear diameter

Δr = wear corner radius

Δa = wear angle

Notes

Example tool list details (HMI): Tool type 156



Tool list				Spindle
Loc.	Type	Tool name	ST	
		MILL_TAPER_CRAD	1	
1		CUTTER_12	1	
2		FRAESER_25	1	
3		FRAESER_10	1	
4		FACINGTOOL_50	1	
5		CUTTER_32	1	
6		GEWINDE_M12	1	
7		BOHRER_10	1	
8		DRILL_8.5	1	
9		FRAESER_12	1	
10		MESSERKOPF_63	1	
11		ZENTRIERER_20	1	
12		CUTTER_8	1	
13		CUTTER_16	1	
14		MILL_CORN.RAD.1.5	1	
15		ANGLE HEAD	1	
16		BALL_8	1	

Tool parameters – internal data		Cutt.edge 1
Cutt. edge data	Cutt.edge 1	Monitoring dat
Tool type	156	Tool life
Cutting edge pos.	9	Prewarn. limit
Geo length 1	100.000	Set val
Geo length 2	0.000	Quantity
Geo length 3	0.000	Prewarn.limit
Geo. Ø	12.000	Set val
Parameter 7	2.000	Wear
Parameter 8	0.000	Prewarn.limit
Parameter 9	0.000	Set val
Parameter 10	0.000	
Parameter 11	10.000	
ΔLength 1	0.000	
ΔLength 2	0.000	
ΔLength 3	0.000	
Δ Ø	0.000	
Parameter 16	0.000	
Parameter 17	0.000	
ΔLength 5	0.000	
Δangle 1	0.000	
Δangle 2	0.000	

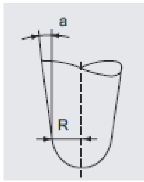
D = tool diameter

r = corner radius

Δr = wear corner radius

Δa = wear angle

Example tool list details (HMI): Tool type 157



Tool list				Spindle
Loc.	Type	Tool name	ST	
		DIEMILL_TAPER	1	
1		CUTTER_12	1	
2		FRAESER_25	1	
3		FRAESER_10	1	
4		FACINGTOOL_50	1	
5		CUTTER_32	1	
6		GEWINDE_M12	1	
7		BOHRER_10	1	
8		DRILL_8.5	1	
9		FRAESER_12	1	
10		MESSERKOPF_63	1	
11		ZENTRIERER_20	1	
12		CUTTER_8	1	
13		CUTTER_16	1	
14		MILL_CORN.RAD.1.5	1	
15		ANGLE HEAD	1	
16		BALL_8	1	

Tool parameters – internal data		Cutt.edge 1
Cutt. edge data	Cutt.edge 1	Monitoring dat
Tool type	157	Tool life
Cutting edge pos.	9	Prewarn. limit
Geo length 1	100.000	Set val
Geo length 2	0.000	Quantity
Geo length 3	0.000	Prewarn.limit
Geo. Ø	16.000	Set val
Parameter 7	0.000	Wear
Parameter 8	0.000	Prewarn.limit
Parameter 9	0.000	Set val
Parameter 10	0.000	
Parameter 11	10.000	
ΔLength 1	0.000	
ΔLength 2	0.000	
ΔLength 3	0.000	
Δ Ø	0.000	
Parameter 16	0.000	
Parameter 17	0.000	
ΔLength 5	0.000	
Δangle 1	0.000	
Δangle 2	0.000	

D = tool diameter

a = taper angle

Δr = wear corner radius

Δa = wear angle

Note:

The parameter Geo. R (radius) is assigned to the system variable $TC_DP6 and can be changed in setting date ***SD54215 $SNS_TM_FUNCTION_MASK_SET*** with Bit 0=1 to Geo. Ø (diameter) .

Notes

Programming of surface normal vectors

General explanation

Programming of surface normal vectors is required in conjunction with **CUT3DF** for face milling of curved surfaces (free form surfaces) of any kind.

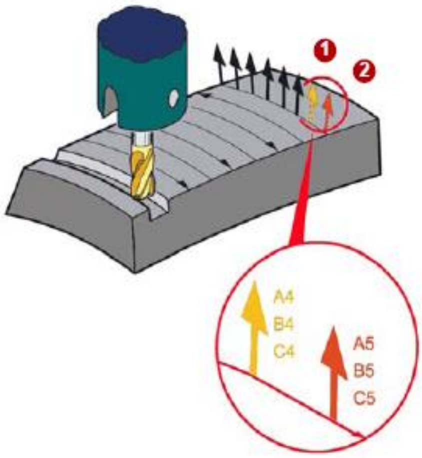
For **3-axis face milling** of curved surfaces with **CUT3DFF**, you require line-by-line definition of 3D paths (X Y Z) on the workpiece surface, and surface normal vectors (A4= B4= C4= and/or A5= B5= C5=). The tool direction must point in Z-direction of the work coordinate system defined by the currently active frame.

For **5-axis face milling** of curved surfaces with **CUT3DFF** additional information for definition of tool orientation, in form of a directional vector (A3= B3= C3=) or as orientation angle (LEAD and TILT) is required.

The tool shape and dimensions are taken into account in the calculations that are normally performed in the CAM system. In addition to the NC blocks, the postprocessor writes the tool orientations (for 5-axis machining) - the surface normal's and the G-code for 3D tool radius compensation into the NC-program.

This feature offers the machine operator the option of using slightly smaller tools, than originally programmed for in the CAM-system, to calculate the NC paths.
For example the NC blocks have been calculated with a end mill Ø 10mm. In this case, the workpiece could also be machined with a end mill of Ø 9.9 mm, which would otherwise result in a different surface profile.

Path curvature



The path curvature is described by surface normal vectors with the following components:

- **G1 X Y Z A4=, B4=, C4=**
  Programming of surface normal vector at block start (**1**)

- **G1 X Y Z A5=, B5=, C5=**
  Programming of surface normal vector at block end (**2**)

The surface normal vector must be perpendicular to the path tangent, within a limit value set via machine data, otherwise an alarm will be output.

Effect

- If a block only contains the **start vector**, the surface normal vector will remain constant throughout the block.

- If a block only contains the **end vector**, interpolation will run from the end value of the previous block via great circle interpolation to the programmed end value.

- If both **start vector** and **end vector** are programmed, interpolation runs between the two directions, also with great circle interpolation. This allows continuously smooth paths to be created.

Notes

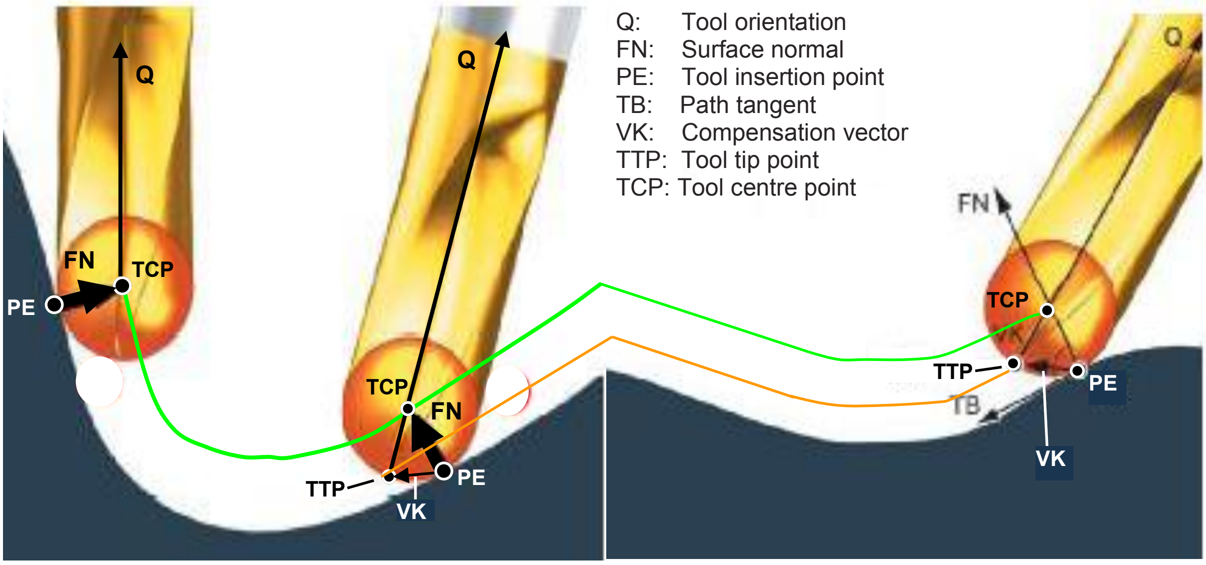
Programming

The following applies for programming:

- CUT3DF is only effective with active 5-axis transformation (TRAORI)
- A programmed surface normal vector is valid, until a new surface normal is programmed.
  The length of a vector is meaningless.
- Regardless of the active work plane (G17-G19), in the initial setting, surface normal vectors point in the Z direction.
- Vector components that have not been programmed are set to zero.
- With active ORIWKS the surface normal vectors relate to the active frame and rotate when the frame rotates. This applies for programmed orientations as well as for the orientations that are derived from the active plane. With active ORIWKS the surface normal vectors are being retained upon a frame change. If switching from ORIWKS to ORIMKS the modified orientation caused by the frame rotation is not being reversed.
- The surface normal vector must be perpendicular to the path tangent, within a limit value set via machine data ***MD21084 $MC_CUTCOM_PLANE_PATH_LIMIT*** (minimum angle between surface normal and path tangent), otherwise an alarm will be output.

Calculation of compensation vector for CUT3DF

The correction for the 3D tool radius compensation with CUT3DF is calculated through the vector components pointing from the tool insertion point (**PE**) to the Tool tip (**TTP**).
The resulting Vector (**VK**) is used for the tool radius compensation in CUT3DF.



The tool path can be output from the CAM system to either the tool tip or the tool centre.

If the tool path is output to the tool tip, it is referred to as **TTP-programming** (TTP=Tool Tip Point)

If the tool path is output to the tool centre, it is referred to as **TCP-programming** (TCP=Tool Centre Point)

Notes

Effect of CUT3DF on 5-axis face milling

Compensation of tool radius by the NC with CUT3DF (TTP programming)

The tool path is generated on the CAM to the too tip (**orange line**). For tool path calculation of a norm tool with a cutter radius **R>0** is used

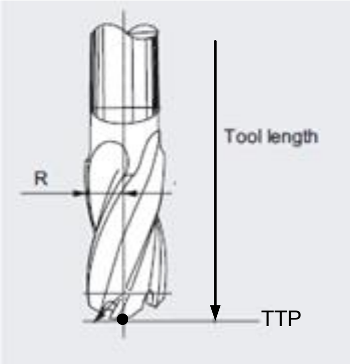
The NC program contains the following information in each NC block:
TTP= X,Y,Z coordinates of each point to be approached in reference to the tool tip (**orange line**)
FN= Surface normal vectors A4=,B4=,C4= and/or A5=,B5=,C5= at each point
Q= Tool orientation as LEAD/TILT angles, or vectors A3=,B3=,C3=

Setting the tool parameters on the control

The tool radius is set by the operator in the tool list parameter **R=0** The tool length is measured to the tool tip (**TTP**) and set in the tool list with the parameter **L1.**

Correction of tool radius on the control

Tool radius correction can be preformed by the operator on the control as differential radius from the norm tool radius, via tool wear offset parameters **ΔR** from the centre line of the tool.



Compensation of tool radius by the NC with CUT3DF (TCP programming)

The tool path is generated on the CAM to the tool centre line (**green line**). For tool path calculation of a norm tool with a cutter radius **R>0** is used

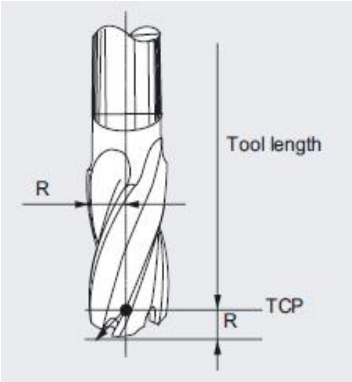
The NC program contains the following information in each NC block:
TCP= Coordinates of the tool centre path X,Y,Z (**green line**)
FN= Surface normal vectors A4=,B4=,C4= and/or A5=,B5=,C5= at each point
Q= Tool orientation as LEAD/TILT angles, or vectors A3=,B3=,C3= at each point

Setting the tool parameters on the control

The tool radius is set by the operator in the tool list with a tool radius **R=0** and tool length **L1** is measured to the tool centre point (**TCP**) or alternatively measured to the tip and then offset in the program with the G-code command **TOFFL=**<radius> from the tip to the centre point of the tool.

Correction of tool radius on the control

Tool radius correction can be preformed by the operator on the control as differential radius from the norm tool radius, via tool wear offset parameters **ΔR** from the centre line of the tool.



Notes

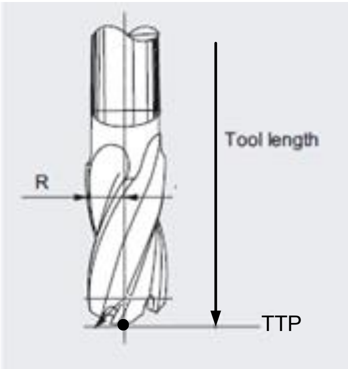
Compensation of tool radius by the CAM without CUT3DF (TTP programming)

The tool path is generated on the CAM to the tool tip (**orange line**). For tool path calculation of a norm tool with a cutter radius **R>0** is used

The NC program contains the following information in each NC block:
TTP= X,Y,Z coordinates of each point to be approached in reference to the tool tip (**orange line**)
Q= Tool orientation vectors A3=,B3=,C3= or direct rotary axis positions A, B, C to each point

Setting the tool parameters on the control

The tool radius is set by the operator in the tool list with a tool radius **R=0** and tool length **L1** is measured to the tool centre point (**TCP**) or alternatively measured to the tip and then offset in the program with the G-code command **TOFFL=**<radius> from the tip to the centre point of the tool.



No correction of tool radius on the control possible

The parameters in the tool list for tool radius R are not evaluated, since the CAM has already compensated for tool radius and no G-code (CUT3DF and G41/G42) for activation of radius compensation is output from postprocessor to the NC program.

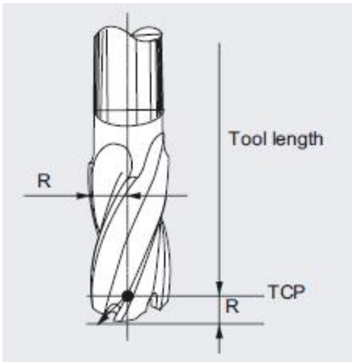
Compensation of tool radius by the CAM without CUT3DF (TCP programming)

The tool path is generated on the CAM to the tool centre line (**green line**). For tool path calculation of a norm tool with a cutter radius **R>0** is used

The NC program contains the following information in each NC block:
TCP= Coordinates of the tool centre path X,Y,Z (**green line**)
Q= Tool orientation vectors A3=,B3=,C3= or direct rotary axis positions A, B, C to each point

Setting the tool parameters on the control

The tool length is measured to the tool centre point (TCP) by the Operator on the machine or offset in the program with the G-code command **TOFFL=**<radius> from the tip to the centre point.



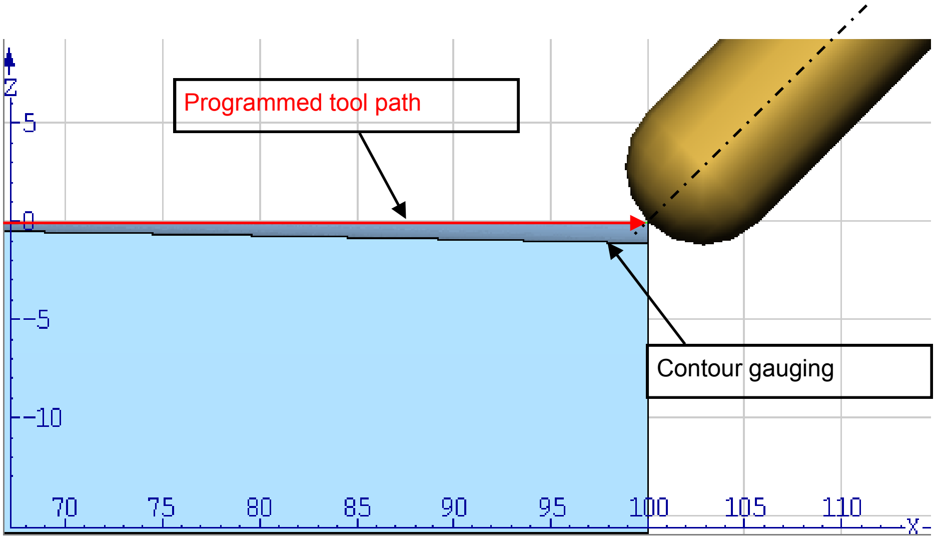
No correction of tool radius on the control possible

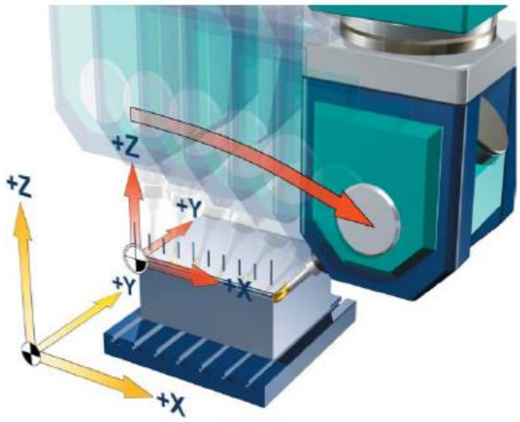
The parameters in the tool list for tool radius R are not evaluated, since the CAM has already compensated for tool radius and no G-code (CUT3DF and G41/G42) for activation of radius compensation is output from postprocessor to the NC program.

Notes

Example 1: Face milling without 3D cutter compensation:

The tool path is programmed to the workpiece surface (TTP) with active 5-axis transformation TRAORI. The tool tip (TTP) maintains position relative to the workpiece geometry along the programmed tool path. No compensation of tool radius takes place and the tool radius gauges into
the workpiece between traversal from start to end orientation.





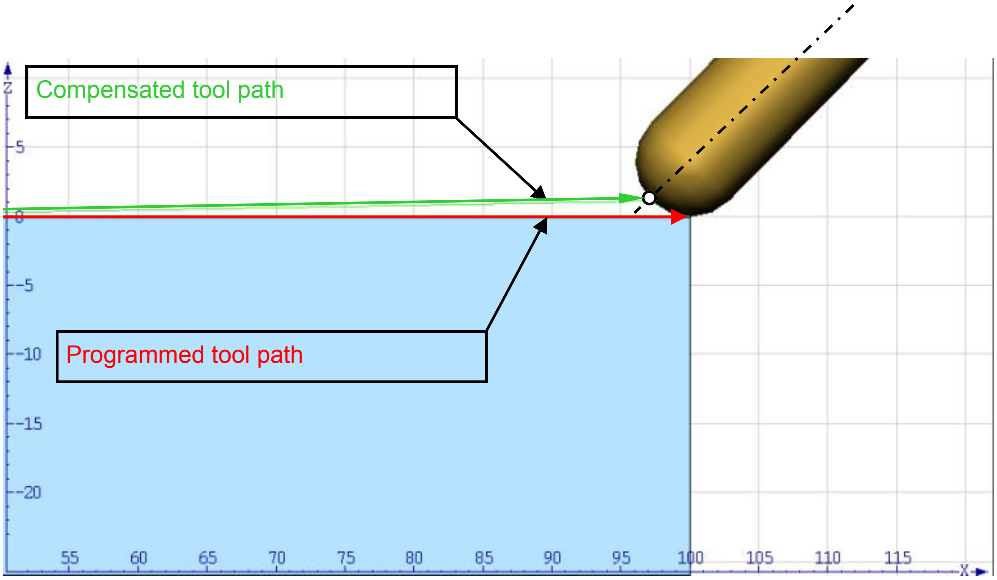
NC/WKS/M105_3D_TRC/EXAMPLE1_OHNE_CUT3DF

N10 CYCLE800(1,"TABLE",200000,57,0,0,0,0,0,0,0,0,0,1,,1)
N11 CYCLE800()
N12 WORKPIECE(,,,"BOX",112,0,-30,-80,0,0,100,60)
N13 T="BALL_CUTTER_D8"
N14 M6
N15 S10000 M3 F1000
N16 G54 G0 A0 C90
N17 **TRAORI**
N18 ORIVECT
N19 ORIWKS
N19 G0 X-10 Y0 Z5
N20 G1 Z0 F600
N21 X0 Y0 C3=1
N22 G1 Z0
N23 X100 Y0 A3=1 C3=1 ; Tool orientation change to +45° in ZX plane
N25 G0 Z100
N26 C3=1 ; Align tool parallel to Z axis
N27 TRAFOOF
N28 M30

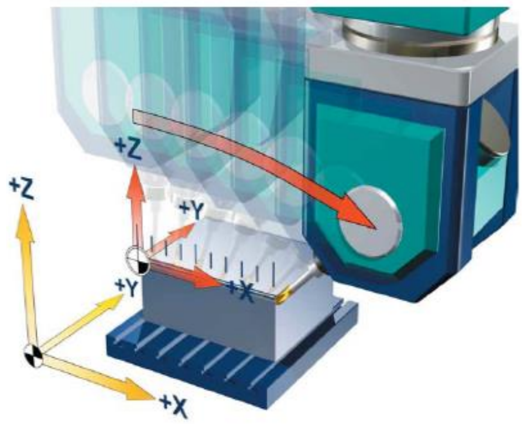
Notes

Example 2: Face milling with 3D cutter compensation tool type 110:

Compensation of tool radius takes place. The tool path is programmed to the workpiece surface (TTP) with active 5-axis transformation TRAORI. The tool tip (TTP) maintains position relative to the compensated tool path, the tool radius does here by not gauge into the workpiece between traversal from start to end orientation.



NC/WKS/M105_3D_TRC/EXAMPLE2_CUT3DF_TYPE_110

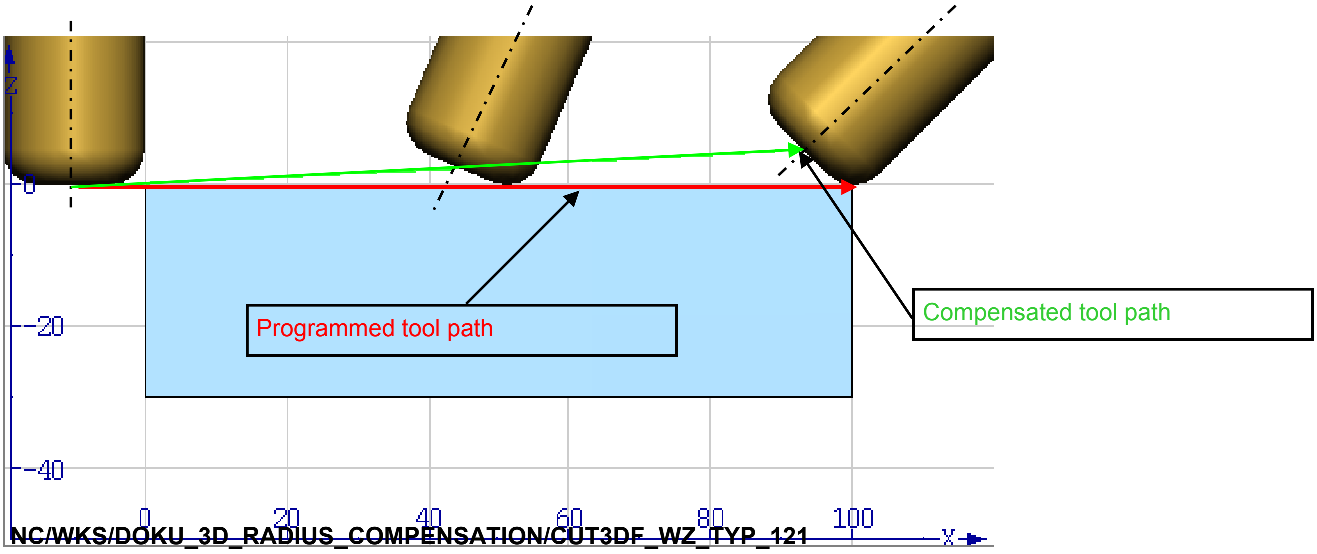



```
N10 CYCLE800(1,"TABLE",200000,57,0,0,0,0,0,0,0,0,0,1,,1)
N11 CYCLE800()
N12 WORKPIECE(,,,"BOX",112,0,-30,-80,0,0,100,60)
N13 T="BALL_CUTTER_D8"
N14  M6
N15 S10000 M3 F1000
N16 G54 G0 A0 C90
N17 TRAORI
N18 ORIVECT
N19 ORIWKS
N20 CUT3DF ; Cutter Compensation 3D Face milling
N21 G0 X-10 Y0 Z5
N22 G1 Z0 F600
N23 G42 X0 Y0 C3=1 A5=0 B5=0 C5=1; Activate cutter compensation and
set surface normal
N24 G1 Z0
N25 X100 Y0 A3=1 C3=1 ; Tool orientation change to +45° in ZX plane
N26 G40 X110 ; Cancel cutter compensation
N27 G0 Z100
N28 C3=1 ; Align tool parallel to Z axis
N29 TRAFOOF
N30 M30
```

Notes

Example 3: Face milling with 3D cutter compensation tool type 121:

Compensation of tool radius takes place. The tool path is programmed to the workpiece surface (TTP) with active 5-axis transformation TRAORI. The tool tip (TTP) maintains position relative to the compensated tool path, the tool radius does hereby not gauge into the workpiece between traversal from start to end orientation.




```
NC/WKS/DOKU_3D_RADIUS_COMPENSATION/CUT3DF_WZ_TYP_121
N10 R50=10;         START POSITION IN Y
```

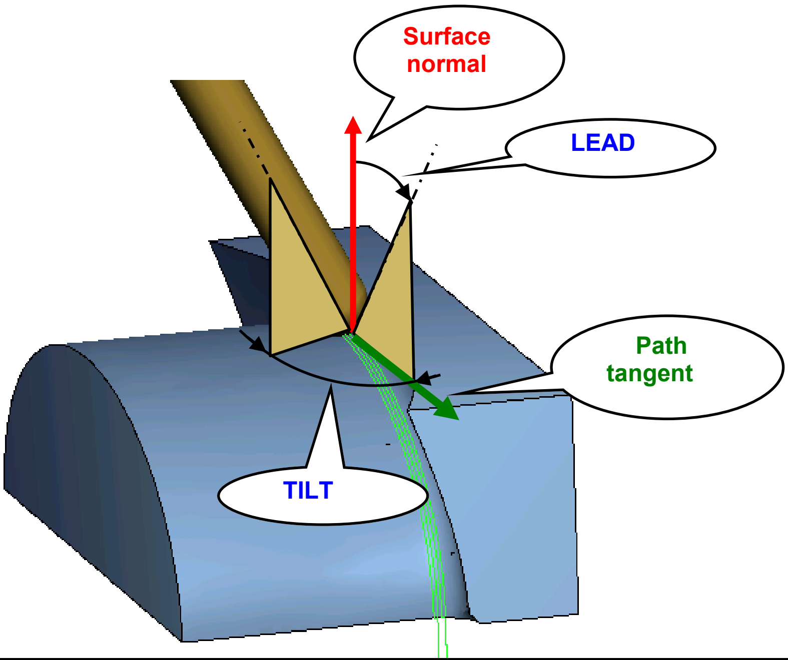
```
N20 R51=0.1;        STEP
N30 R55=20;         END POSITION
N40 G54 G40 G17
N50 CYCLE800(2,"TABLE",200000,57,0,0,0,0,0,0,0,0,0,1,,1)
N60 CYCLE800(2,"0",200000,57,0,0,0,0,0,0,0,0,0,1,,1)
N70 WORKPIECE(,"",,"BOX",112,0,-40,-80,0,0,105,105)
N80 T="D10_R1"
N90 M6
N100 S20000 M3
N110 G0 X-5 Y=R50
N120 G0 Z5
N130 TRAORI
N140 G54
N150 ORIVECT
N160 ORIWKS
N170 CUT3DF; Cutter Compensation 3D Face milling
N180 ANFANG:
N190 G0 Z5
N200 G0 X-5 Y=R50
N210 G1 Z0 F4000
N220 G42 X0 Y=R50 C3=1 A5=0 B5=0 C5=1; Activate tool radius compensation
N230 G1 X105 Y=R50 A3=1 C3=1
N240 G40 X110 Y=R50 C3=1; Deactivate tool radius compensation
N250 G0 Z5
N260 R50=R50+R51
N270 IF R50 <= R55 GOTOB START
N280 TRAFOOF
N290 CYCLE800(2,"TABLE",200000,57,0,0,0,0,0,0,0,0,0,1,,1)
```

Notes

Example 5-Axis face milling with CUT3DF

Programming of an ellipse (finishing with angle step FI=0.5°)
Tool path with constant tool setting angle to surface normal
Tool orientation with LEAD and TILT
Surface normal with A5=.. B5=.. C5=..
Orientation Interpolation ORIPATH/ORIWKS

(See example)



Note:

The orientation angles **Lead** and **Tilt** are defined by the surface normal. Whereby the surface normal can constantly change over the tool path with regard to the contour profile.

Notes

NC/WKS/DOKU_3D_RADIUS_COMPENSATION/CUT3DF_LEAD_TILT

```
N10 R50=10;        START POSITION IN Y
N20 R51=0.1;       STEP
N30 R55=20;        END POSITION
N40 G54 G40 G17
N50 CYCLE800(2,"TABLE",200000,57,0,0,0,0,0,0,0,0,0,1,,1)
N60 CYCLE800(2,"0",200000,57,0,0,0,0,0,0,0,0,0,1,,1)
N70 WORKPIECE(,"",,"BOX",112,0,-40,-80,0,0,105,105)
N80 T="BALL_D6"
N90 M6
N100 S20000 M3
N110 G0 X-5 Y=R50
N120 G0 Z5
N130 M82 ; ATTENTION ONLY USE ROTARY AXIS CLAMPING ON CHIRON
N140 H82 ; ATTENTION ONLY USE ROTARY AXIS CLAMPING ON CHIRON
N150 TRAORI
N160 G54
N170 ORIPATH
N180 ORIWKS
N190 CUT3DF
```

```
N200 START:
N210 G0 Z5
N220 G0 X-5 Y=R50
N230 G1 Z0 F4000
N240 G42 X0
N250 LEAD=45 TILT=-30
N260 G1 X105 Y=R50 A5=0 B5=0 C5=1
N270 G40 X110 Y=R50 LEAD=0 TILT=0 A5=0 B5=0 C5=1
N280 G0 Z5
N290 R50=R50+R51
N300 IF R50 <= R55 GOTOB START
N310 TRAFOOF
N320 CYCLE800(2,"TABLE",200000,57,0,0,0,0,0,0,0,0,0,1,,1)
N330 M30
```

Notes

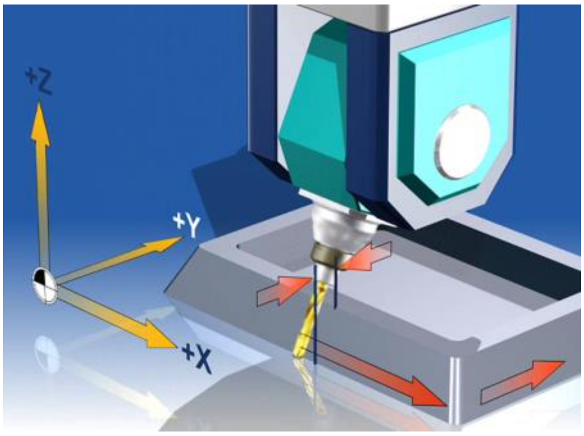
3D circumferential milling

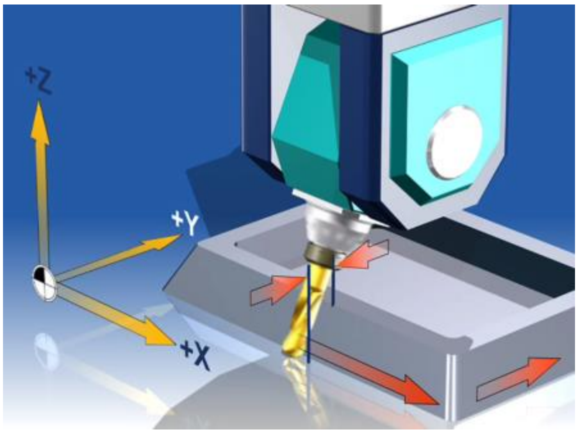
3D-Circumferential milling with CUT3DC

General explanation:

3D tool radius compensation with CUT3DC is generally used for circumferential milling of free form surfaces or pocket milling with inclined side walls.

In this 3D tool radius compensation, a deviation of the mill radius is compensated by infeed toward the normals of the surface to be machined. The plane, in which the milling tool face is located, remains unchanged if the insertion depth has remained the same.





For example, a milling tool with a smaller radius than a standard tool would not reach the pocket base, which is also the limitation surface. For automatic tool infeed, this limitation surface must be known to the control. (see page 21 "3D circumferential milling with limitation surfaces CUT3DCC/CUT3DCCD").

Programming

The following applies for programming:

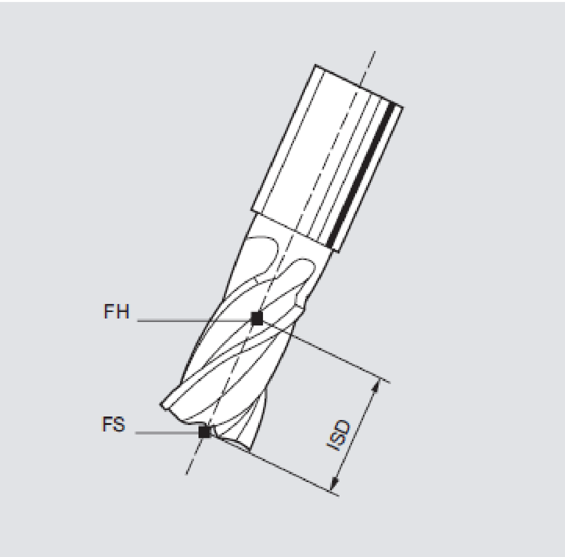
- The 3D radius compensation is only effective when 5-axis transformation “TRAORI” is selected.
- Tool orientation change is taken into account in tool radius compensation for cylindrical tools only.
- The commands are modally effective and written in the same group as CUT2D and CUT2DF.
- The same programming commands apply to 3D tool radius compensation as to 2D tool radius compensation. The left/ right offset is specified in the direction of motion using G41/G42.
  Intermediate blocks are permitted with 3D tool radius compensation.
- A circular block is always inserted at out corners. G450/G451 have no significance.
- The DISC command is not evaluated. The approach response is always controlled with NORM.
- The command is not deselected until the next movement in the current plane is performed.
  This always applies for G40 and is independent of the CUT command.

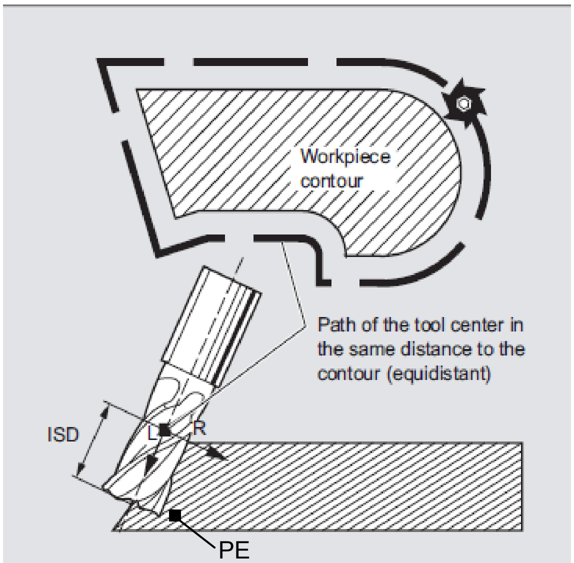
Notes

Insertion depth (ISD)

Program command ISD (insertion depth) is used to program the tool insertion depth for circumferential milling operations. This makes it possible to change the position of the machining point on the outer surface of the tool. ISD defines the distance between tool tip (FS) and the milling tool reference point (FH). The milling tool reference point (FH) is obtained by projecting the programmed machining point onto the tool axis. Insertion depth ISD is only evaluated when 3D tool radius compensation is active.

The type of milling used here is implemented by defining a path (guide line) and the corresponding orientation, whereby the tool path maintains the same distance to the contour (equidistant).
In this type of machining, the shape of the tool on the path is not relevant. Important is only the radius at the tool insertion point (PE).





NC command	Description
CUT3DC	3D tool radius compensation normal to path tangent and tool orientation. Used for 5-aixs circumferential milling e.g., for pocket milling with oblique side walls.
ISD	The distance (<value>) between the milling tool tip (FS) and the milling tool construction point (FH) are specified using the command ISD.

Notes

3D-Circumferential milling with CUT3DCC/CUT3DCCD

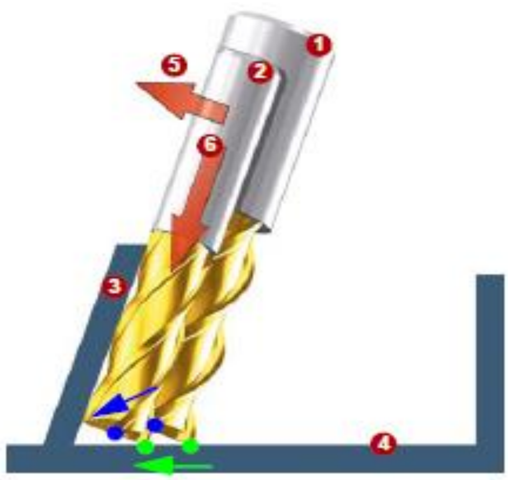
General explanation:

In 3D-circumferential milling with a continuous or constant change in tool orientation, the tool centre point path is frequently programmed for a defined standard tool. Because in practice suitable standard tools are often not available, a tool that does not deviate too much from a standard tool can be used.

For example, a milling tool with a smaller radius (**2**) than a standard tool (**1**) would not reach the pocket base (**4**), which is also the limitation surface. For automatic tool infeed, this limitation surface must be known to the control. A typical application for this function can be found in particular in structural components in the aviation industry.

Here the control system has to take into account not only that a compensation is required in the direction of the machining surface (**3**), but also that a deeper insertion in tool direction is required (**6**).

The control recognizes the fact, that not only radius compensation in the machining surface direction (**5**) has to take place , but also at the same time a axial correction along the tool axis must be preformed (**6**), so that the point of action **PE** is at the same level as the pocket floor (**4**). This results in a shift of the **TCP** in the direction of the pocket base (**4**).



1 Standard tool (specified for program)
2 Real tool (tool with smaller radius)
3 Machining surface, inner surface
4 Limitation surface, base of pocket
5 Correction to machining surface
6 Correction to limitation surface

TCP Tool centre point
PE Tool insertion point

NC command	Description
CUT3DCCD	Takes account of a limitation surface for a real differential tool that the programmed standard tool would define. The NC program defines the centre point path of a standard tool.
CUT3DCC	With the use of cylindrical tools takes account of a limitation surface that the programmed standard tool would have reached. The NC program defines the contour on the machining surface.

Notes

Tool types and parameters with CUT3DCC/CUT3DCCD

In general for circumferential milling with CUT3DC the shape of the tool on the path is not relevant only in the special case of CUT3DCC/CUT3DCCD with consideration of a limitation surface, tool dimensions of an end mill with corner rounding with reduced radius as compared with the standard tool (picture 1.10)is evaluated.

Example: standard tools with corner rounding

Address	Parameter	Explanation
121 or 131	$TC_DP1	End mill with corner rounding (Toroidal end mill)
R =	$TC_DP6	Nominal tool radius or diameter
r =	$TC_DP7	Nominal corner radius

Example: real tool with corner rounding

Address	Parameter	Explanation
121 or 131	$TC_DP1	End mill with corner rounding (Toroidal end mill)
R or Geo Ø	$TC_DP6	Nominal tool radius or diameter
r	$TC_DP7	Nominal corner radius
ΔR or ΔØ	$TC_DP15	Deviation of tool radius or Ø of the real tool compared with the standard tool (tool wear).
Δr	$TC_DP16	Deviation of corner rounding of the real tool compared with the standard tool (tool wear).
R’ or Ø’		Actual tool radius or diameter R’= $TC_DP6 + $TC_DP15 + OFFN (or TOFFR)
r’		Actual corner radius R’= $TC_DP7 + $TC_DP16
In our example $TC_DP15 + OFFN (or TOFFR) as well as $TC_DP16 is negative The tool type ($TC_DP1) is evaluated		

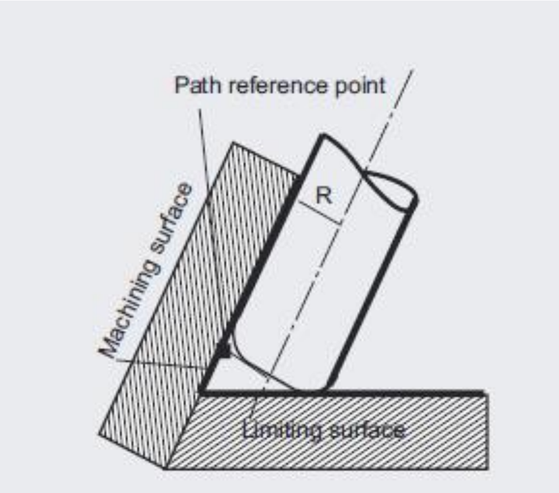
Using cylindrical tools with corner rounding

If cylindrical tools are used, infeed is only necessary if the machining surface and the limitation surface form an acute angle (less than 90 degrees).

If a toroidal endmill is used (endmill with rounded corners) tool infeed in the longitudinal direction is required for both acute and obtuse angles.

If **CUT3DCC** is active with a toroidal endmill the programmed path refers to a fictitious cylindrical mill with the same diameter. The resulting path reference point for a toroidal endmill is shown in picture below.

Notes



The angle between the machining and limitation surface may change from an acute to an obtuse angle and vice versa even within the same block.

The tool actually used, may be either larger or smaller than the standard tool. But the resulting corner radius must not be negative and the sign in front of the resulting tool radius must not change.

In **CUT3DCC** the NC parts program refers to the contour on the machining surface. As with 2D tool radius compensation, the effective tool radius is used, which is totaled from:

- the tool radius (tool parameter $TC_DP6)
- the wear value (tool parameter $TC_DP15)
- and a programmed tool offset OFFN or TOFFR.

The position of the limitation surface is determined by the difference between the following two values:

- Tool radius of standard tool (programmed tool radius)
- Tool radius of real tool (tool parameter $TC_DP6).

In **CUT3DCCD** the tool parameter $TC_DP6 specified does not affect the tool radius and the resulting compensation.

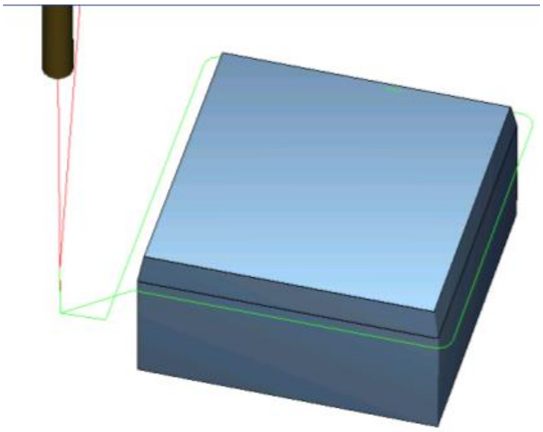
The compensation is the sum of:

- The wear value of the tool radius (tool parameter $TC_DP15) and
- the programmed tool offset OFFN or TOFFR.

Notes

Example 1: Circumferential milling with CUT3DC

A workpiece contour needs to be milled at the circumference with inclined walls of 10 degrees. The bottom edge of the work piece is programmed. A tool radius of 6 mm is to be compensated for in the program.



NC/WKS/DOKU_3D_RADIUS_COMPENSATION/CUT3DC

```
N10 G54 G40 G17
N20 CYCLE800(2,"TABLE",200000,57,0,0,0,0,0,0,0,0,0,1,,1)
N30 CYCLE800(2,"0",200000,57,0,0,0,0,0,0,0,0,0,1,,1)
N40 WORKPIECE(,"",,"BOX",112,0,-40,-80,0,0,105,105)
```

```
N50 T="CUTTER_10MM"
N60 M6
N70 S6000 M3
N80 M82 ; ATTENTION ONLY USE ON CHIRON
N90 H82 ; ATTENTION ONLY USE ON CHIRON
N100 TRAORI
N110 G54
N120 ORIVECT
N130 CUT3DC
N140 G645
N150 GROUP_BEGIN(0,"OUTSIDE_CONTOUR",0,0)
N160 G0 X-20 Y-20 D1
N170 G0 Z10
N180 G1 Z-10 F1000
N190 G41 X0 A3=1.7632698 B3=0 C3=10
N200 G1 Y105
N210 G1 A3=0 B3=-1.7632698 C3=10
N220 G1 X105
N230 G1 A3=-1.7632698 B3=0 C3=10
N240 G1 Y0
N250 G1 A3=0 B3=1.7632698 C3=10
N260 G1 X0
N270 G40 X-20 Y-20 A3=0 B3=0 C3=1
N280 G0 Z100
N290 GROUP_END(0,0)
```

```
N320 T="SF_D8_R08"
N330 M6
N340 S10000 M3
N350 TRAORI
N360 G54
N370 ORIVECT
N380 CUT3DC
N390 G645
N400 GROUP_BEGIN(0,"OUTSIDE_CONTOUR",0,0)
N410 G0 X-20 Y-20 D1
N420 G0 Z10
N430 G1 Z-10 F1000
N440 G41 X0 A3=1.7632698 B3=0 C3=10
N450 G1 Y105
N460 G1 A3=0 B3=-1.7632698 C3=10
N470 G1 X105
N480 G1 A3=-1.7632698 B3=0 C3=10
N490 G1 Y0
N500 G1 A3=0 B3=1.7632698 C3=10
N510 G1 X0
N520 G40 X-20 Y-20 A3=0 B3=0 C3=1
N530 G0 Z100
N540 GROUP_END(0,0)
N550 TRAFOOF
N560 CYCLE800(2,"TABLE",200000,57,0,0,0,0,0,0,0,0,0,1,,1)
N570 M30
```

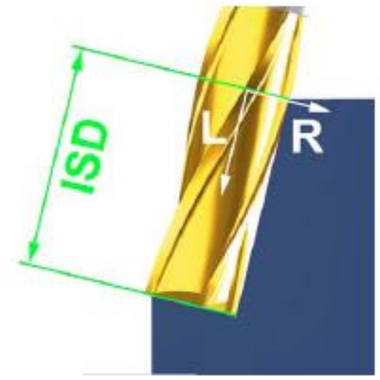
Note:

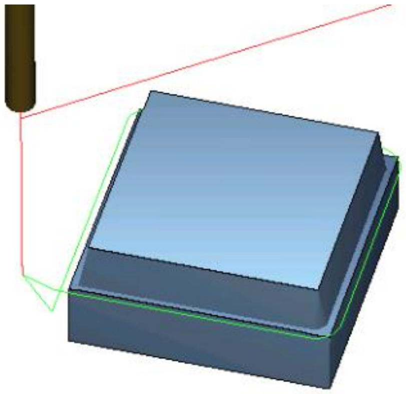
With **CUT3DC,** the tool radius compensation in the NC program refers to the contour on the machining surface (real tool).

Notes

Example 2: Circumferential milling with CUT3DC and ISD

A workpiece contour needs to be milled at the circumference with inclined walls of 10 degrees. The top edge of the work-piece is to be programmed with the engaged length of ISD=20 mm. A tool radius of 6 mm is to be compensated for in the program.





NC/WKS/M105_3D_TRC/EXAMPLE_2_CUT3DC_ISD

N100 CYCLE800(2,"TABLE",200000,57,0,0,0,0,0,0,0,0,0,1,,1)
N101 CYCLE800()
N102 G54
N103 WORKPIECE(,,"","BOX",112,0,-50,-80,0,0,100,100)
N104 T="CUTTER_12"
N105 M6

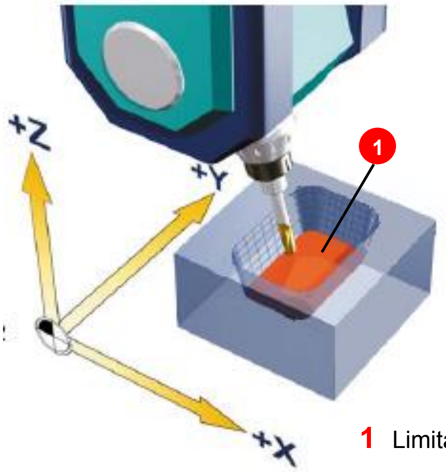
N106 S6000 M3
N107 CYCLE832(0.01,3,1)
N108 **TRAORI**; Activate 5-axis transformation
N109 **ORIWKS**; Orientation reference WKS
N110 **ORIVECT**; Great circle interpolation
N111 **CUT3DC**; Selection of circumferential 3D tool radius compensation
N112 **ISD=20**; Engaged length (insertion depth)
N113 G54 G0 X-20 Y-20 Z10
N114 G1 **Z0** F1000 M8
N115 **G41** X1.76 A3=1.7632698 B3=0 C3=10; Switch on TRC right of contour and approach
N116 G1 Y98.2367302
N117 G1 A3=0 B3=-1.7632698 C3=10
N118 G1 X98.2367302
N119 G1 A3=-1.7632698 B3=0 C3=10
N120 G1 Y1.7632698
N121 G1 A3=0 B3=1.7632698 C3=10
N122 G1 X-10
N123 **G40** X-20 Y-20 A3=0 B3=0 C3=1; Switch off TRC and exit
N124 G0 Z100
N125 **TRAFOOF**; Deactivate 5-axis transformation
N126 M30

Note:

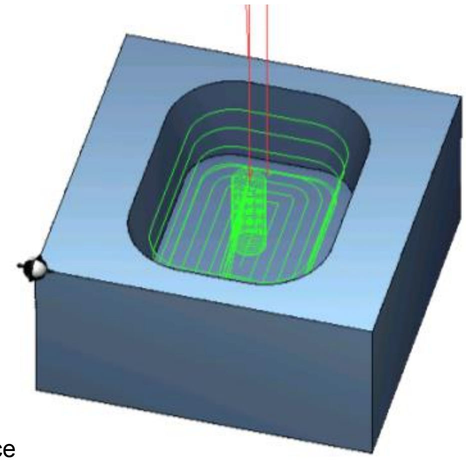
With **ISD** it is possible for example to programm the upper contour profile as a guide line and offset this guide line with the programmed value in **ISD** to the tool tip. This results into the lower profile.

Notes

Example 3: Circumferential pocket milling with CUT3DCC



1 Limitation surface



NC/WKS/DOKU_3D_RADIUS_COMPENSATION_CUT3DCC_A2

```
N10 R50=51.5;     POCKET WIDTH
N20 R51=71.5;     POCKET LENGTH
N30 R52=12;       POCKET ANGLE
N40 G54 G17 G40 G64
N50 CYCLE800(2,"TABLE",200000,57,0,0,0,0,0,0,0,0,0,1,,1)
N60 CYCLE800(2,"0",200000,57,0,0,0,0,0,0,0,0,0,1,,1)
N70 WORKPIECE(,"",,"BOX",112,0,-40,-80,0,0,105,105)
```

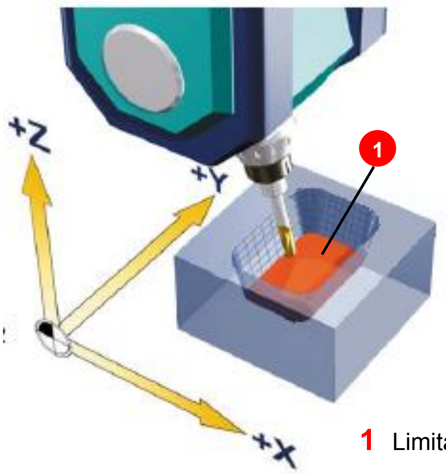
```
N80 T="CUTTER_10MM"
N90 M6
N100 S10000 M3
N110 G0 X52.5 Y52.5
N120 POCKET3(50,0,1,-10,R50,R51,15,52.5,52.5,0,10,0.1,0,1000,0.1,0,21,60,8,3,15,3.5,1,0,1,2,11100,11,110)
N130 M82 ; ATTENTION ONLY USE ROTARY AXIS CLAMPING ON CHIRON
N140 H82 ; ATTENTION ONLY USE ROTARY AXIS CLAMPING ON CHIRON
N150 TRAORI
N160 G54
N170 ORIWKS
N180 ORIVECT
N190 CUT3DCC
N200 G64
N210 FGROUP(XC,YC,ZC)
N220 TRANS X52.5 Y52.5
N230 G0 X0 Y0
N240 G0 Z10
N250 G1 Z-9 F1000
N260 G1 G41 X0 Y=-R51/2 Z-10 A2=R52 B2=0 C2=0 F1000
N270 G1 X=R50/2-15
N280 ORICONCCW
N290 G3 X=R50/2 Y=-R51/2+15 A2=0 B2=R52 C2=0 CR=15
NUT=R52*2
N300 ORIVECT
N310 G1 Y=R51/2-15
N320 ORICONCCW
N330 G3 Y=R51/2 X=R50/2-15 A2=-R52 B2=0 C2=0 CR=15
N340 ORIVECT
N350 G1 X=-R50/2+15
N360 ORICONCCW
N370 G3 X=-R50/2 Y=R51/2-15 A2=0 B2=-R52 C2=0 CR=15
NUT=R52*2
N380 ORIVECT
N390 G1 Y=-R51/2+15
N400 ORICONCCW
N410 G3 X=-R50/2+15 Y=-R51/2 A2=R52 B2=0 C2=0 CR=15
NUT=R52*2
N420 ORIVECT
N430 G1 X0
N440 G40 X0 Y0 Z-9 A2=0 B2=0 C2=0
N450 G0 Z50
N460 TRAFOOF
N470 TRANS
N480 CYCLE800(2,"TABLE",200000,57,0,0,0,0,0,0,0,0,0,1,,1)
N490 M30
```

Note:

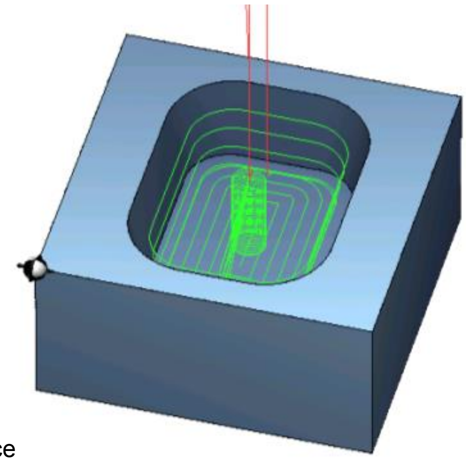
With **CUT3DCC,** the tool radius compensation in the NC program refers to the contour on the machining surface of a real differential tool with consideration of a limitation surface that the norm tool would have reached.

Notes

Example 4: 5-axis circumferential pocket milling with CUT3DCC



1 Limitation surface



NC/WKS/DOKU_3D_RADIUS_COMPENSATION_CUT3DCC_A3

```
N10 R50=51.5;       POCKET WIDTH
N20 R51=71.5;       POCKET LENGTH
N30 R52=12;         POCKET ANGLE
N40 R53=10;         POCKET DEPTH
;*************************************************************************************
N50 R54=R53*TAN(R52);    VECTOR CALCULATION
```

```
N60 G54 G17 G40 G64
N70 CYCLE800(2,"TISCH",200000,57,0,0,0,0,0,0,0,0,0,1,,1)
N80 CYCLE800(2,"0",200000,57,0,0,0,0,0,0,0,0,0,1,,1)
N90 WORKPIECE(,"",,"BOX",112,0,-40,-80,0,0,105,105)
N100 T="CUTTER_10MM"
N110 M6
N120 S10000 M3
N130 G0 X52.5 Y52.5
N140 POCKET3(50,0,1,-10,R50,R51,15,52.5,52.5,0,10,0.1,0,1000,0.1,0,21,60,8,3,15,3.5,1,0,1,2,11100,11,110)
N150 M82 ; ATTENTION ONLY USE ROTARY AXIS CLAMPING ON CHIRON
N160 H82 ; ATTENTION ONLY USE ROTARY AXIS CLAMPING ON CHIRON
N170 TRAORI
N180 G54
N190 ORIWKS
N200 ORIVECT
N210 CUT3DCC
N220 G64
N230 FGROUP(XC,YC,ZC)
N240 TRANS X52.5 Y52.5
N250 G0 X0 Y0
N260 G0 Z10
N270 G1 Z-9 F1000
N280 G1 G41 X0 Y=-R51/2 Z=-R53 A3=0 B3=-R54 C3=R53 F1000
N290 G1 X=R50/2-15
N300 ORICONCCW
N310 G3 X=R50/2 Y=-R51/2+15 A3=R54 B3=0 C3=R53 CR=15 GROOVE=R52*2
N320 ORIVECT
N330 G1 Y=R51/2-15
N340 ORICONCCW
N350 G3 Y=R51/2 X=R50/2-15 A3=0 B3=R54 C3=R53 CR=15 GROOVE=R52*2
N360 ORIVECT
N370 G1 X=-R50/2+15
N380 ORICONCCW
N390 G3 X=-R50/2 Y=R51/2-15 A3=-R54 B3=0 C3=R53 CR=15 GROOVE=R52*2
N400 ORIVECT
N410 G1 Y=-R51/2+15
N420 ORICONCCW
N430 G3 X=-R50/2+15 Y=-R51/2 A3=0 B3=-R54 C3=R53 CR=15
NUT=R52*2
N440 ORIVECT
N450 G1 X0
N460 G40 X0 Y0 Z-9 A3=0 B3=0 C3=R53
N470 G0 Z50
N480 TRAFOOF
N490 TRANS
N500 CYCLE800(2,"TABLE",200000,57,0,0,0,0,0,0,0,0,0,1,,1)
N510 M30
```

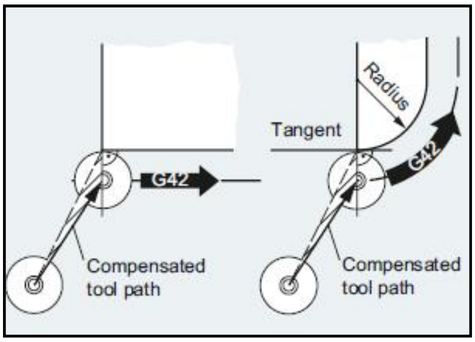
Notes

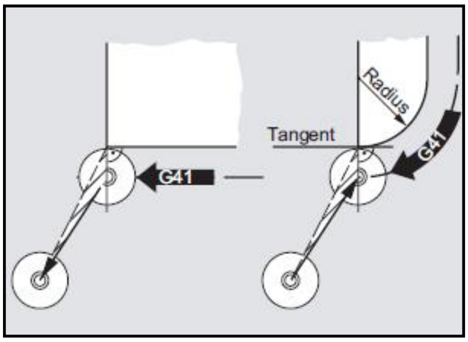
Important G-code functions for 3D-TRC

Contour approach and exit with NORM, KONT

Approach-/exit move with NORM

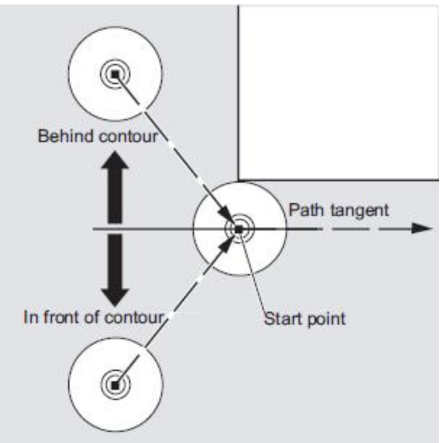
Activate direct approach/retraction on a straight line with tool radius compensation (G41/G42).
The tool is aligned perpendicular to the contour point.





1. Approach:
If NORM is activated, the tool will move directly to the compensated start position along a straight line (irrespective of the preset approach angle programmed for the travel movement) and is positioned perpendicular to the path tangent at the starting point.

2. Retract:
The tool is perpendicular to the last compensated path end point and then moves (irrespective of the preset approach angle programmed for the travel movement) directly in a straight line to the next uncompensated position, e.g. to the tool change point.



Approach/retract move with KONT

Activate approach/retraction with travel around the starting/end point according to the programmed corner behavior G450 or G451. Prior to the approach the tool can be located in front of or behind the contour. The path tangent at the starting point serves as a separation line:

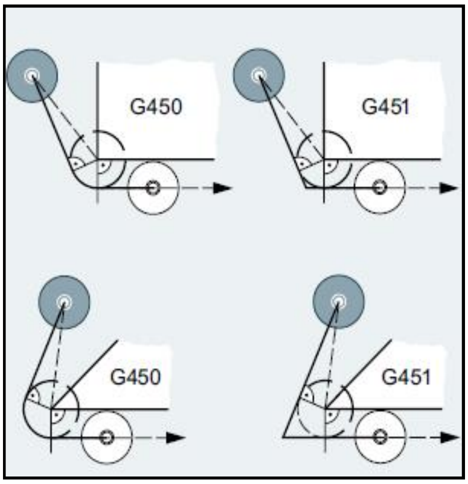
Notes

Accordingly, two scenarios need to be distinguished where approach/exit with KONT is concerned:

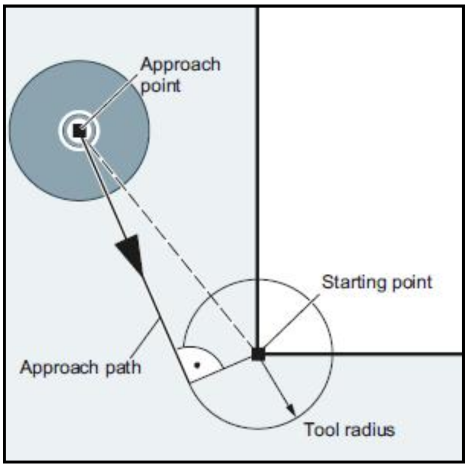
1. The tool is located in front of the contour.
   → The approach/retract strategy is the same as with NORM.

2. The tool is located behind the contour.
   - Approach:
     The tool travels around 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:



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.
- Retract:
  The same applies to retraction as to approach, but in the reverse order.

Notes

List of available tool radius compensations

NC commands	Description
CUT2D CUT2DF	***2 1/D-Circumferential milling*** • 2 1/2D compensation with compensation plane determined with G17 - G19 • 2 1/2D compensation with compensation plane determined by frame (e.g. TRANS, ROT). CUT2DF is the default setting.
CUT3DC	***3D-Circumferential milling (structural components)*** • Compensation perpendicular to path tangent and to tool orientation
CUT3DCC CUT3DCCD	***3D-Circumferential milling with limitation surface*** • NC program refers to the contour on the machining surface (real tool). • NC program refers to the tool centre path (norm tool).

CUT3DFS CUT3DFF CUT3DF	***3D-Face milling (freeform surfaces)*** • Constant orientation (3-axis). Tool faces in Z direction of the coordinate system defined with G17 - G19. Frames have no influence. • Constant orientation (3-axis). Tool in Z direction of the coordinate system currently defined with frame. • 5-axis with variable tool orientation for freeform surfaces with specification of surface normal vectors.
G450 G451 ORID ORIC	***Intersection procedure for 3D TRC (circumferential milling)*** • Transition circle (tool travels round workpiece corners on a circular path • Intersection of equidistant paths (tool backs off from the workpiece corner). • No orientation change in inserted blocks at external corners. Orientation movement is executed in linear blocks • Travel path extended with circles. The orientation change is executed in a circular interpolation around the corner

The compensations are activated by appending the appropriate command G41/G42, e.g.: CUT3DC G41. G40 deactivates the tool radius correction.

M105: END

Notes

A035

Sinumerik Operate

Program Handling Using External Storage

Module Description:

It is often a requirement to store machining programs externally to the controller.
The EES (**E**xecute from **E**xternal **S**torage) feature can then be used to execute the programs, from the external source, as if they were stored in the NC memory.
The external storage can be on the System CF card, USB memory, or a networked PC.

Examples reason for storing programs externally:

- The size of the program.
- The number of programs on the controller is at the limit.
- Centrally stored programs can be accessed by many networked machines.
- Ensures that the programs can be recovered in the event of data loss.

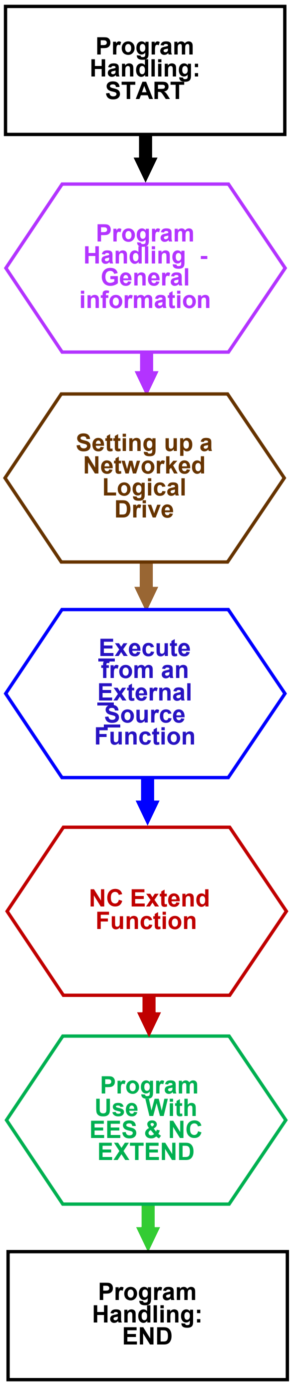
Module Objective:

Upon completion of this module you will know how to:

- Setup a networked logical drive
- Activate the Execute from an External Source command
- Activate the NC Extend memory.
- Use main programs and subprograms with the above features.

Content:

Sinumerik


Program Handling: START
Program Handling - General information
Setting up a Networked Logical Drive
Execute from an External Source Function
NC Extend Function
Program Use With EES & NC EXTEND
Program Handling: END

Notes

Program Handling - General information.

Logical Drives

The logical drive features allows storage of machining programs in addition to the standard NC memory.

.

The additional storage can be:

- On the controller (System CF card).
- A USB device.
- A networked computer.

By default, the USB features are already active.

Programs can be easily copied and pasted between the controller and a logical drive.

It is also possible to execute a program from a logical drive however, there are limitations with program jumps and block search, for example.

NC Extend, EES, and Global Part Program Memory.

Options are available which allow full program functionality for programs stored externally of the controller.

NC Extend: Releases 100MB of System CF card memory. (Option number: **6FC5800-0AP77-0YB0**)

Execute from External Source (EES): Suitable for connecting via Ethernet to an external PC for unlimited storage capacity. (Option number: **6FC5800-0AP75-0YB0**)

Global Part Program Memory:
The Global Part Program Memory feature behaves like standard NC memory and has no restrictions on program jumps, block search etc.
The feature can be activated for ***one*** of the following memory types:

- System CF card (NC Extend).
- USB.
- Networked computer (EES).

Standard NC memory allocation and availability of EES/NC Extend functions:

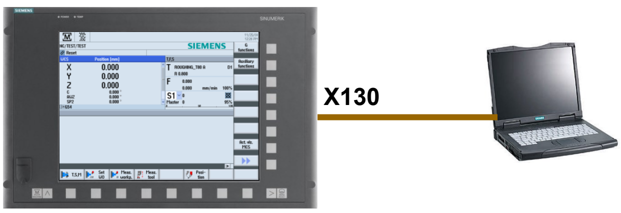
Setting up a Networked Logical Drive.

The following example shows how to connect to a Windows7® computer via the **X130** (Factory network) interface of the controller (Direct connection without DHCP)
X130 will be configured with an IP address of **192.168.100.1**.
The computer will be configured with an IP address of **192.168.100.2**
Both will have the subnet mask of **255.255.255.0**

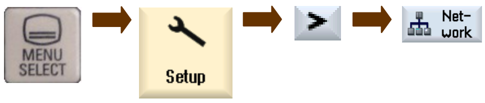
Note:
There are certain IP addresses that must be avoided as they are reserved for use by the system.
IP addresses in the **192.168.214.xxx** range must be avoided.
IP addresses in the **192.168.215.1 to 31** range must also be avoided.

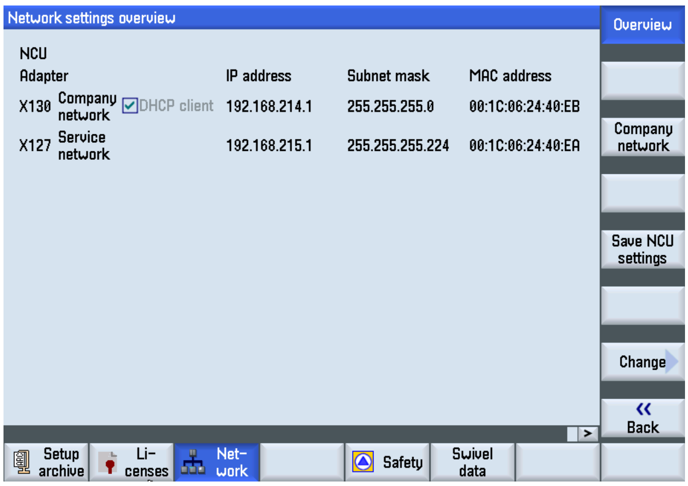
Configuring Interface X130:

Connect a suitable Ethernet cable between the computer and the X130 interface of the controller.



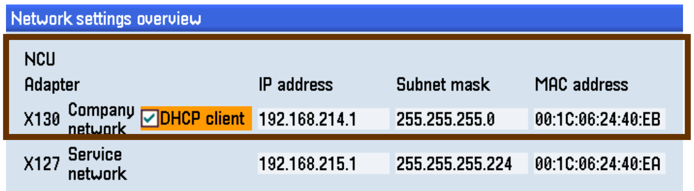
Locate the network settings screen:



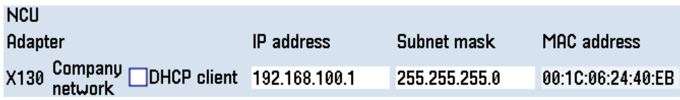


Notes

Setting the X130 IP Address:



Deselect the DHCP client option and enter the required IP address and subnet mask (192.168.100.1/255.255.255.0):

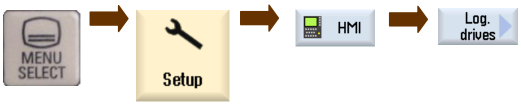


Changes will become effective after RESET (po)

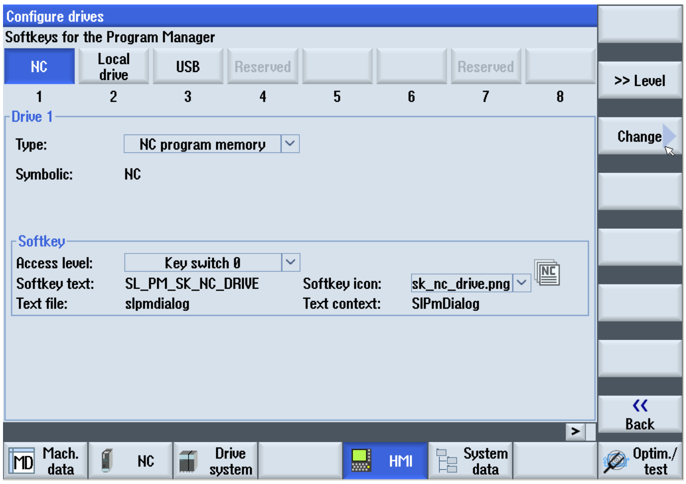
Note: The control needs to be switched off/on to activate the new settings

Setting up the Logical Drive:

Locate the logical drives set-up area:

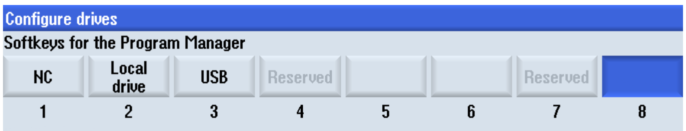


Default Logical drives screen:

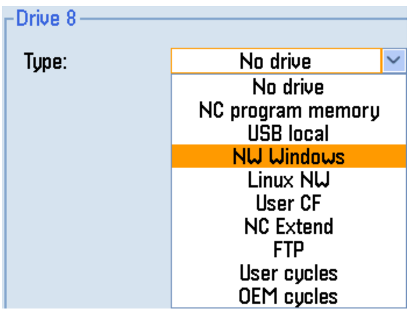


Select a free soft-key. Further soft-keys, 24 in total, are accessed via the ***>>Level*** soft-key.

Example - Soft-key 8:



Choose the ***NW Windows*** option:



For this example the following data is required (the external computer settings will be changed later):

Computer name: 192.168.100.2
(IP address of the PC)

Enabling name: Programs
(Name of the shared folder on the PC).

User name: AUDUSER
(Login name of the PC)

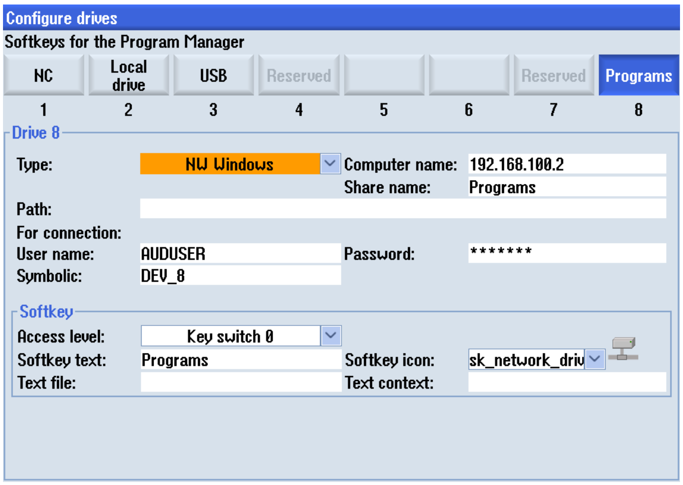
Password: SUNRISE
(Login password of the PC, for example)

Soft-key text: Programs
(the text that will appear on the soft-key in the Program manager area).

The remaining parameters are to be left unchanged for this example.

Notes

The relevant data has to be entered:



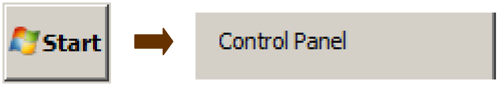
Drive set up

Computer Network Set-up.

The computer must now be given an IP address that is in the same subnet mask.
The IP address for this example is: **192.168.100.2** with a sub-net mask of **255.255.255.0.**

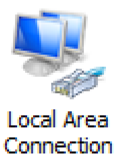
Locate and open the **Network and Sharing Center** of the computer to be connected to the controller.

This can be found in the **Control Panel:**

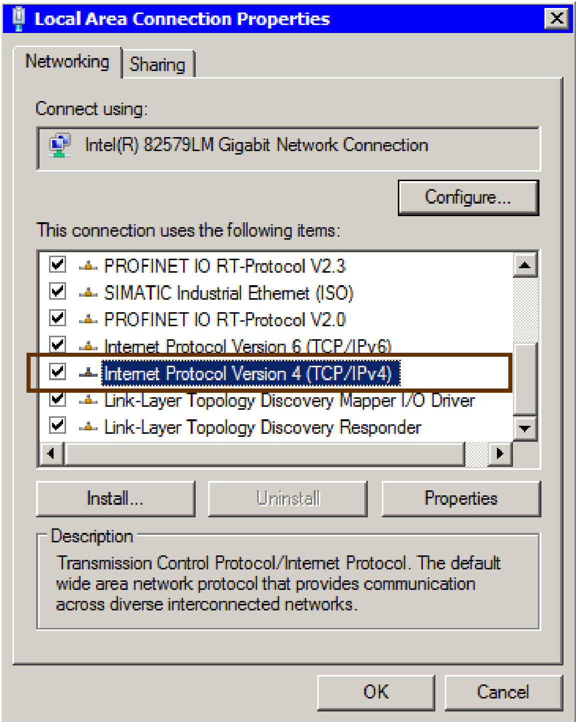


Control Panel Home

Manage wireless networks
Change adapter settings
Change advanced sharing settings



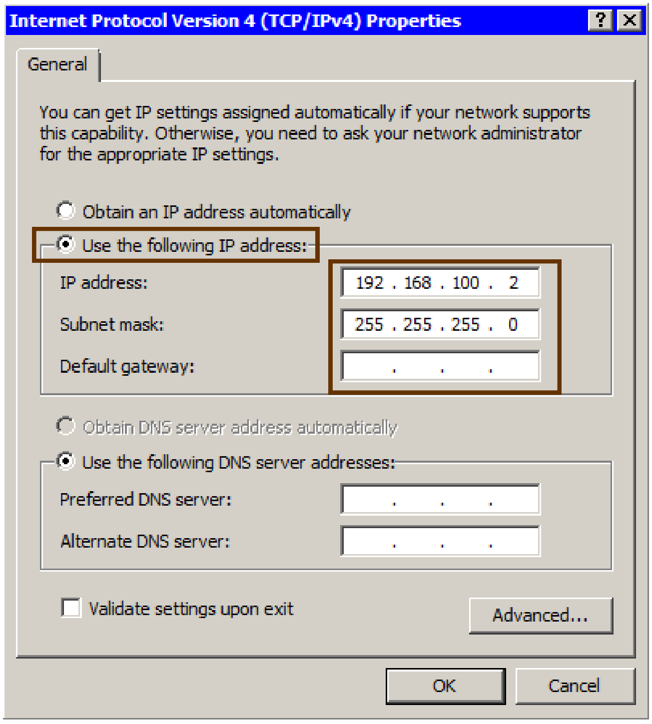
Double click the connection in use and then choose ***Properties.***



Double click the ***Internet Protocol Version 4(TCP/IPv4)*** option.

Notes

The IP address and the subnet mask can now be entered:



Computer- Shared Folder

A shared folder allows it to be accessed via the network.
The folder is required for the storage of programs which will be accessible from the controller.

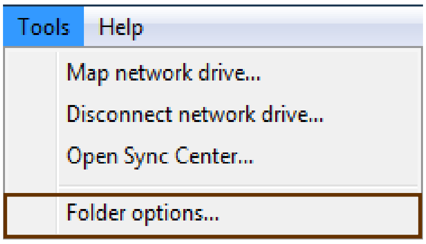
For reasons of security it is possible to restrict access to the shared folder by setting permissions.

The following example will show how to create a folder on the D drive and how to set up sharing and permissions.

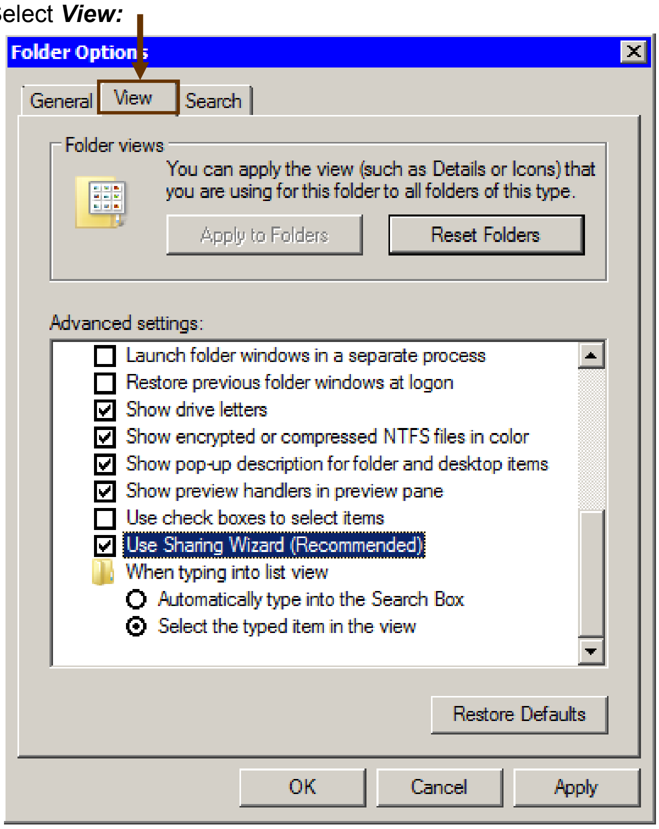
The option - ***Use sharing Wizard*** - can be deselected, via the ***Tools*** menu in Windows® Explorer, to ease setup:

Create a new folder:

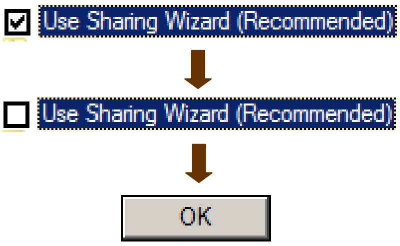
From Windows® Explorer, select the ***Tools*** menu, followed by ***Folder options:***



Select ***View:***



Deselect the ***Use Sharing Wizard*** option:



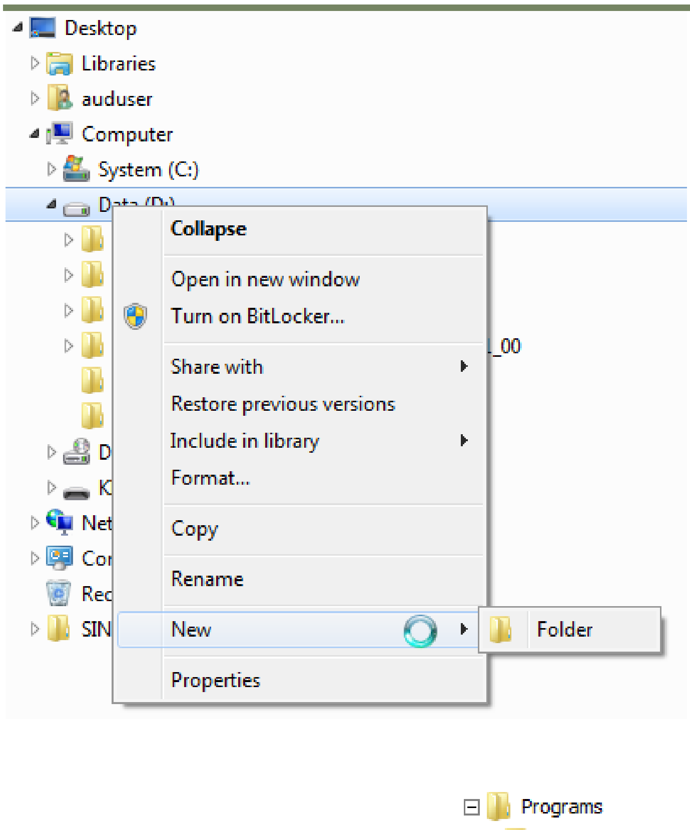
Creating a New Folder

Example:

In Windows® Explorer:

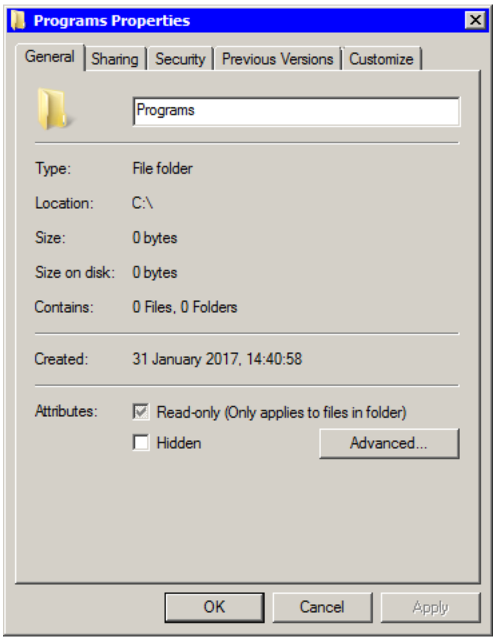
- Select the **D** drive.
- With the mouse pointer In the right hand window, right click and create a new folder.
- Give a suitable name to the folder.
  Example ***Programs.***

Notes

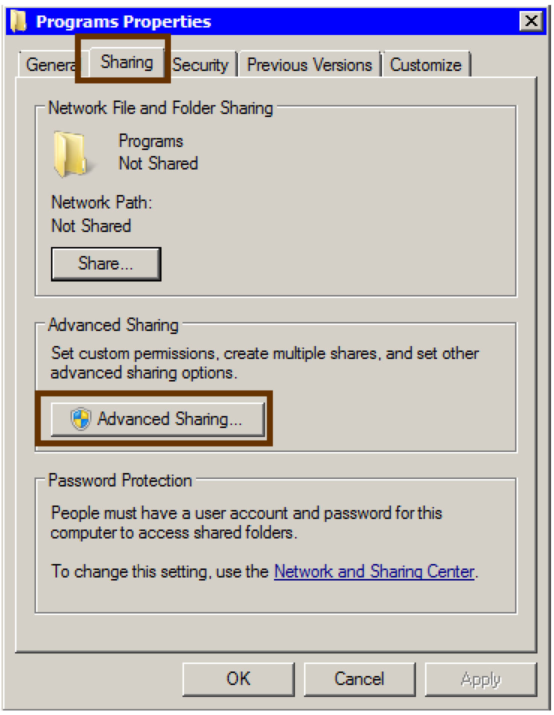


Sharing the New Folder

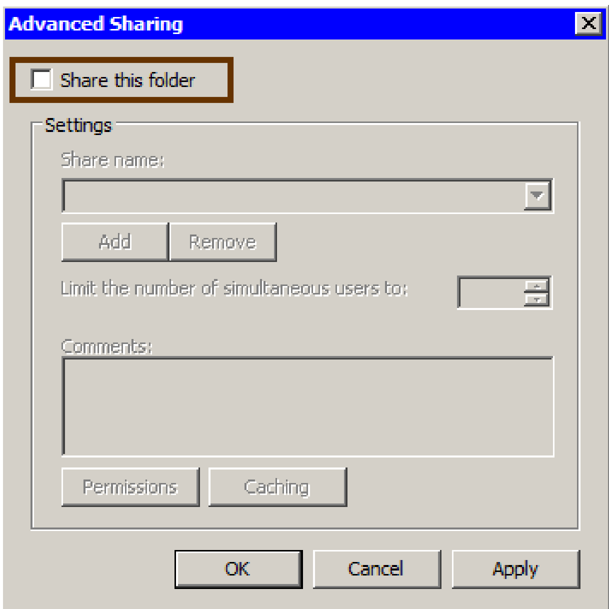
- Right click with the mouse pointer on the ***Programs*** folder.
- Select the ***Properties*** option from the list



- Select the ***Sharing*** tab.
- Select the ***Advanced sharing*** tab.

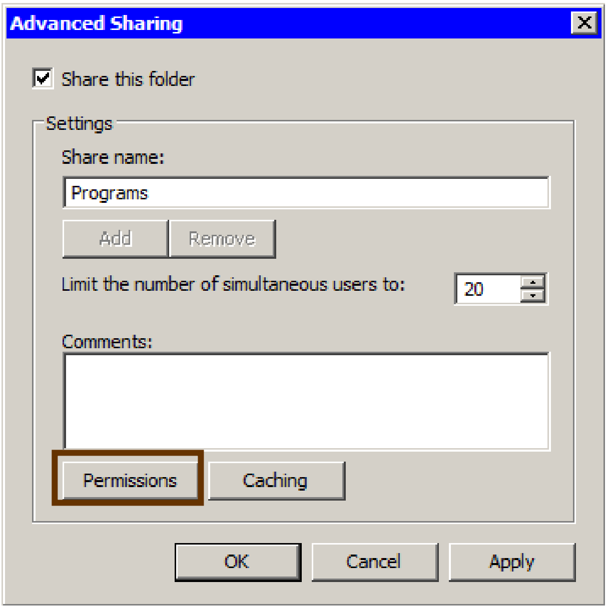


- Select ***Share this folder***.

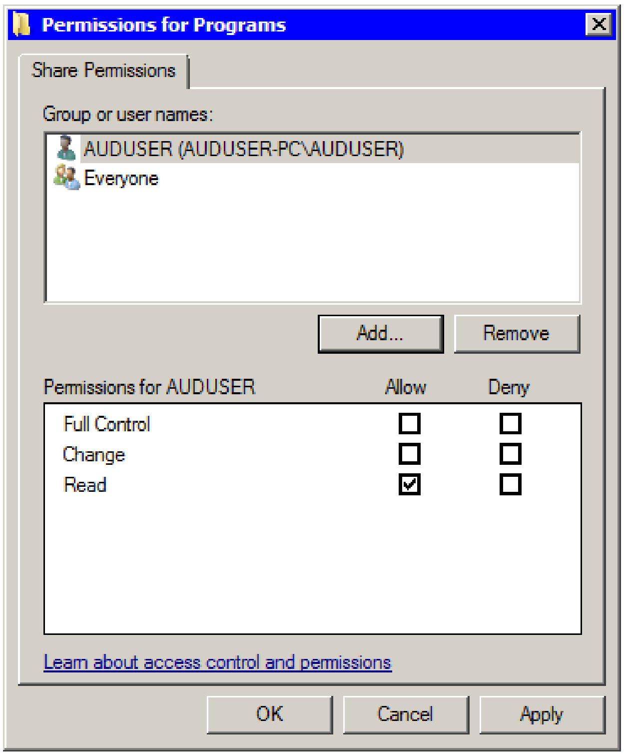


Notes

- Select *Permissions*.

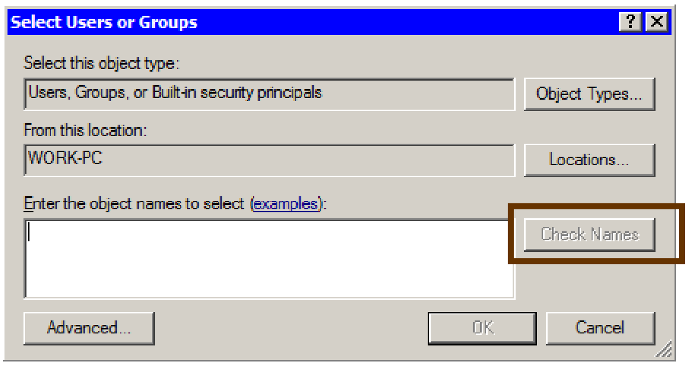


The default setting is for everyone to have read only access to the folder.



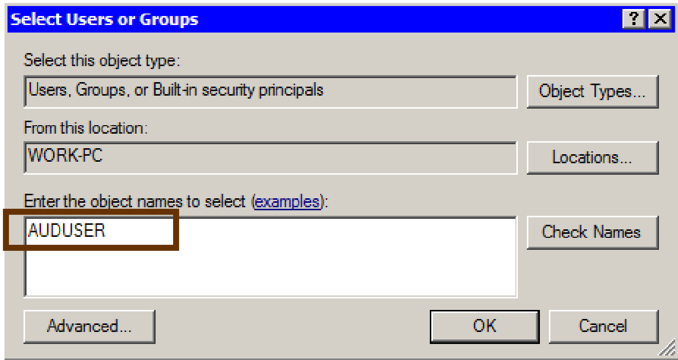
It is possible to restrict access to the shared folder by other users of the PC.
The default setting is for every user to have read only access to the shared folder.
If restrictions are required, the ***Everyone*** user can be removed or the permissions deselected.
Users can than be added via the ***Add*** button.

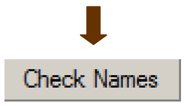
The required users of the computer can be searched for in the ***Check Names*** field.



In the following example, the user ***AUDUSER*** is going to be given permission to access the shared folder.

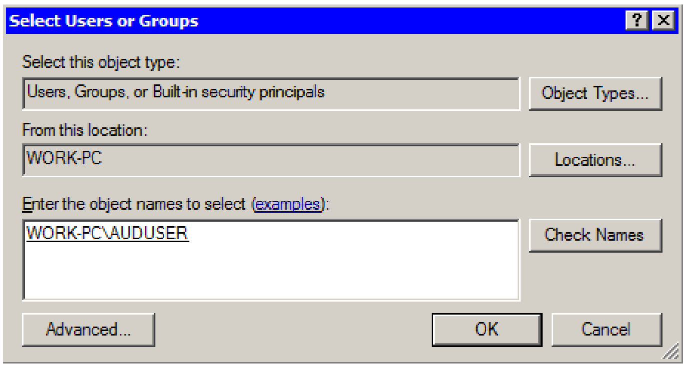
Enter ***AUDUSER*** in the **Check Names** field, then select the ***Check Names*** button.

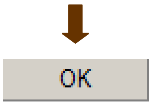




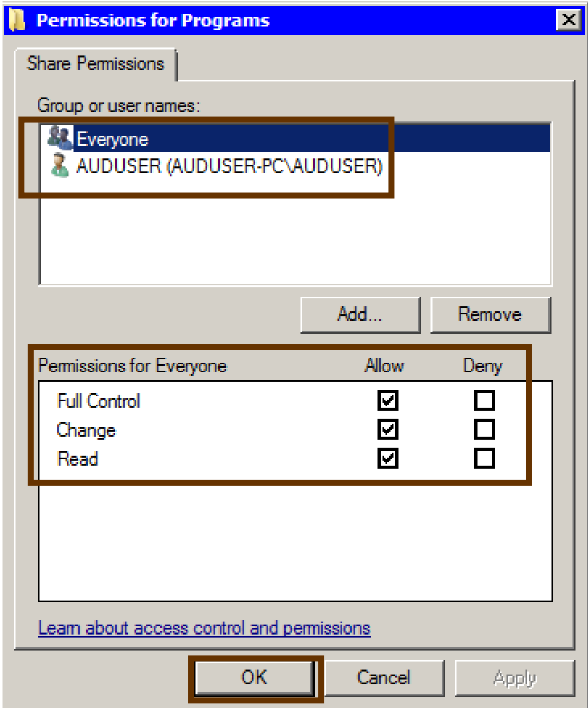
Notes

If the user exists, it will be displayed.

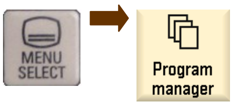


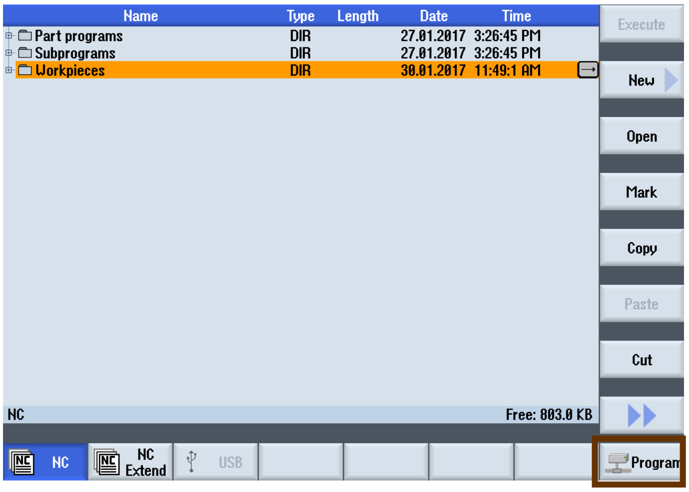


Example: AUDUSER has been granted full access to the shared “Programs” folder.
The Share permission for Everyone has been removed.

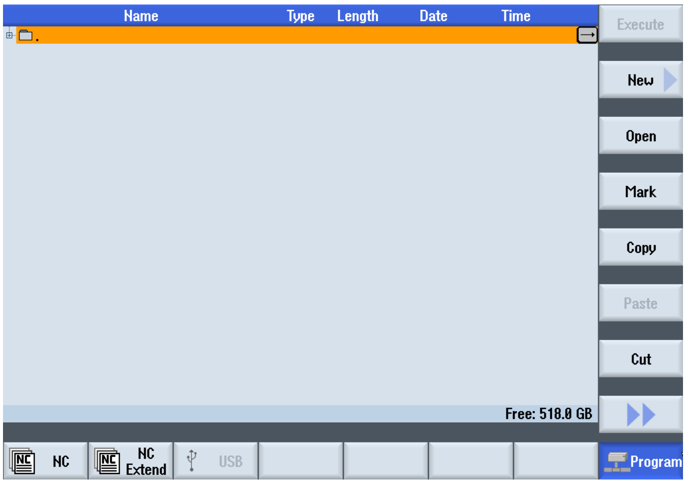


It should now be possible to access the shared folder from the ***Program manager*** area of the controller.





The shared folder being accessed from the controller.



Note: If the Soft-key for accessing the shared folder is grey, this indicates a problem with the network set up or possibly the security of the computer hosting the shared folder.
Firewall settings should be checked and temporarily disabled to see if this is causing problems.

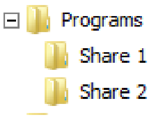
Notes

Accessing individual sub-folders.

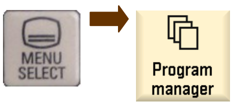
If a folder is created within the shared folder it will inherit the share properties of the host folder.

It is possible to directly access a subfolder by entering a path on the Logical drive set-up screen.

Example:
The ***Programs*** folder has been shared.
Two folders, ***Share 1*** & ***Share 2***, have been added within the shared folder.
There is no need to share these new folders.



Access the ***Program manager*** screens:

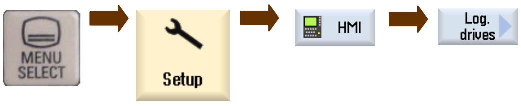


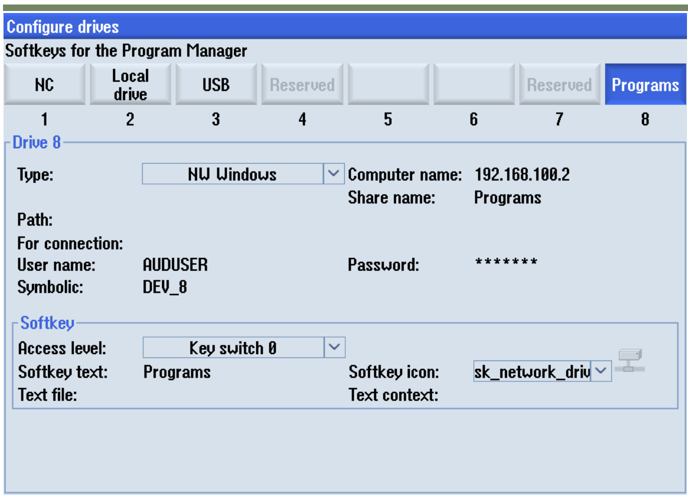
The new folders can be seen inside the original shared folder:

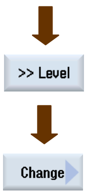
Accessing individual folders.

By entering a path in the logical drives set-up screen it is possible to access individual folders contained within the main shared folder.

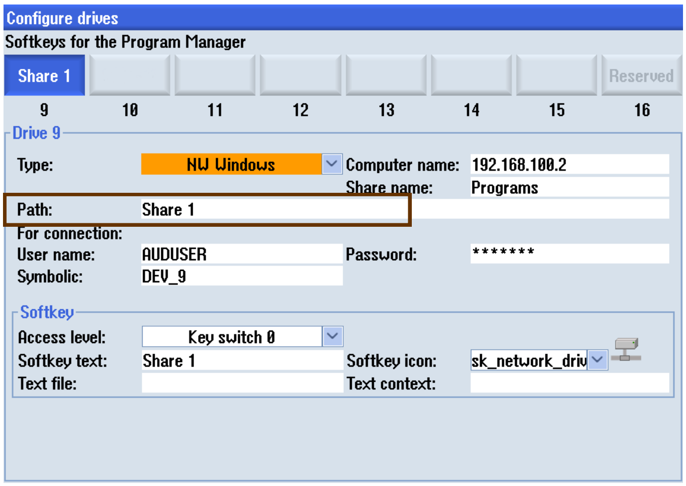
Example:
The following example shows two additional soft-keys which have been set-up to give access to one each of the new folders.





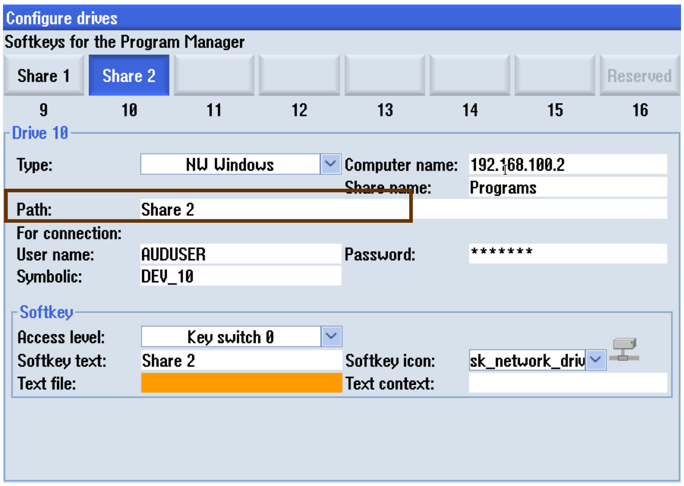


The path to **Share 1** has been included.

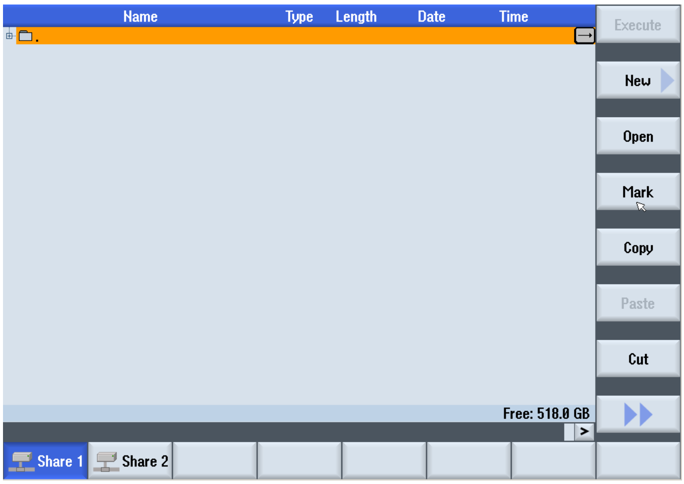


Notes

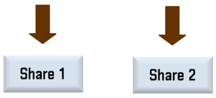
The path to **Share 2** has been included.



Access soft-keys 9 & 10 in the ***Program manager*** area:



The two sub-folders can now be accessed:



Execute from an External Source Function

The **EES** (**E**xecute from **E**xternal **S**ource) function allows a networked logical drive to be used with the **Global part program memory** setting. The size of available memory is limited only by the space available on the networked PC.
It is also possible to use the EES function on a USB memory device or user CF card.

When the option is used in conjunction with the **Global part program memory** setting, it behaves exactly like standard NC memory. i.e. there are no restrictions on block search, long program jumps etc., that are found with other methods of executing programs from an external source. It is also possible to edit the program in NC stop.

Activating the EES Function:

The option, **6FC5800-0AP75-0YB0**, needs to be set.

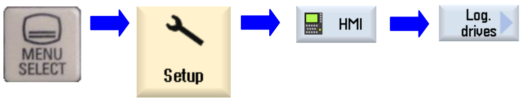
Activate the option with an NCK Reset.

Once the option is set and activated it can be decided which logical drive is to be used.
This can be:

- USB memory device
- Networked computer.

Setting the Global part program memory

Locate the ***Logical drives*** area:



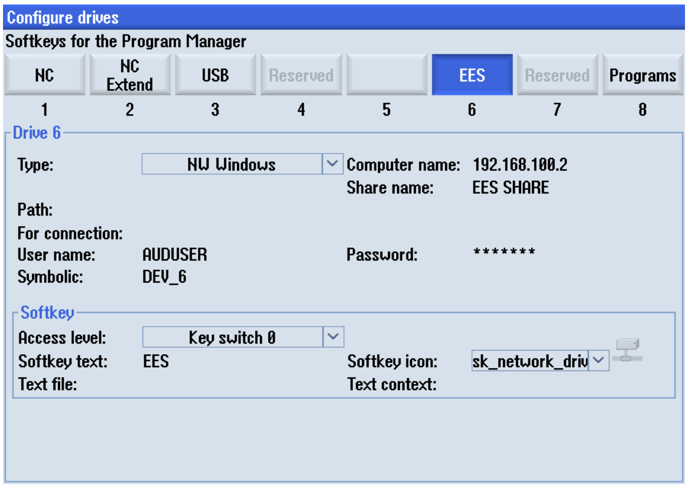
The global part program memory setting can be applied to either:

- A USB device
- The NC Extend memory
- A networked computer.

Notes

Example - Setting the Global part program memory for a networked computer.

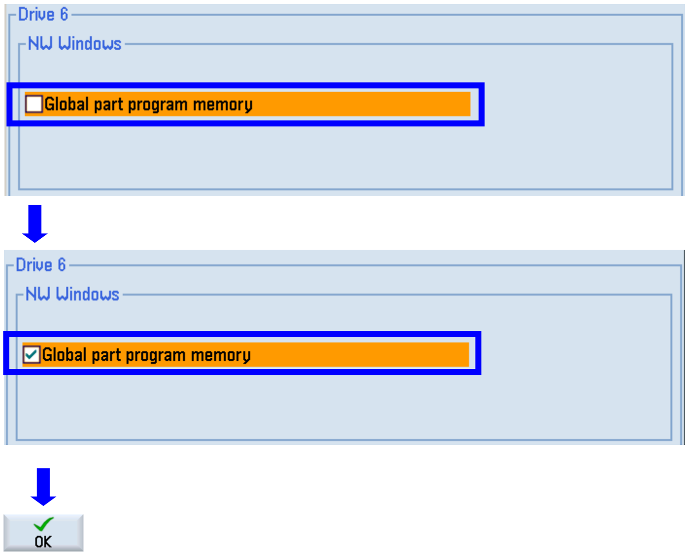
For the example a new networked logical drive (***EES SHARE***) has been assigned to soft-key 6:



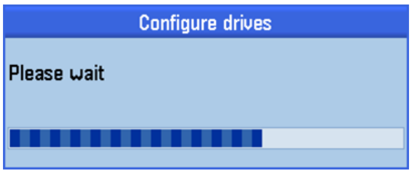
The ***EES SHARE*** folder has been created and shared on the networked computer:

From within the Logical drives area, select the ***EE***S logical drive, followed by the ***Change*** and then ***Details*** soft-keys.

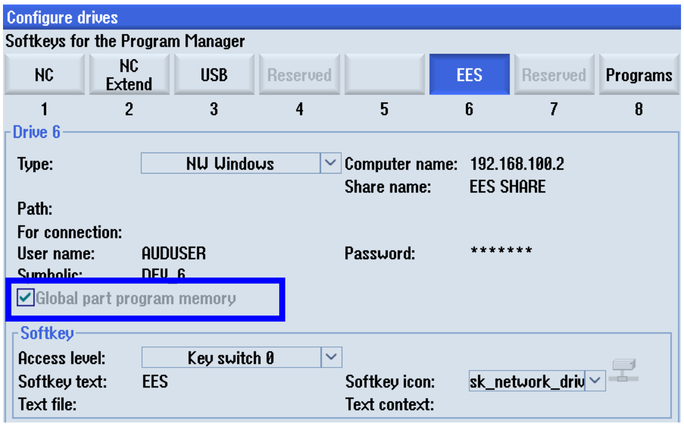
The Global part program memory can now be set.

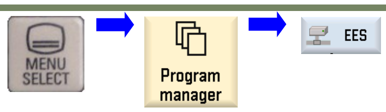


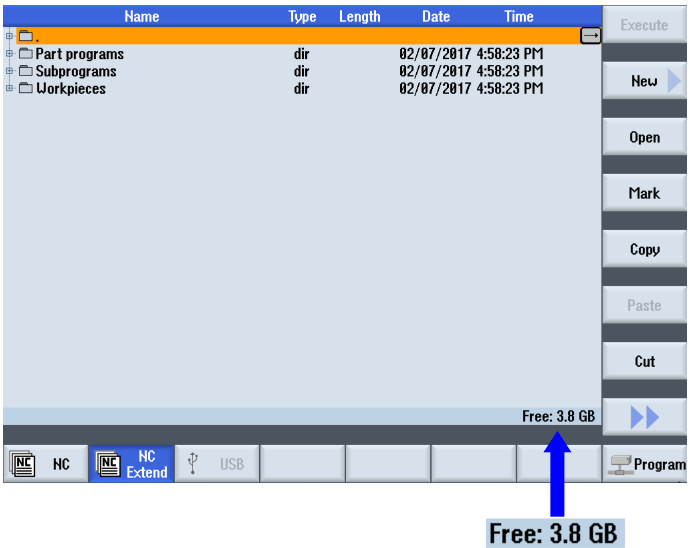
The EES SHARE Logical drive will now be activated for Global part program memory.



Confirmation the Global part program memory is active is seen on the Logical drive page for that particular logical drive:





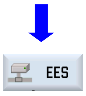


The available space, in this case the networked computer's hard disk, can be seen at the bottom right of the screen.

Notes

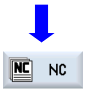
The result of activating the Global part program memory is the creation of three folders which match the default NC memory folders, also this drive whether a local drive, USB or a networked drive will become the “MASTER NC MEMORY”.

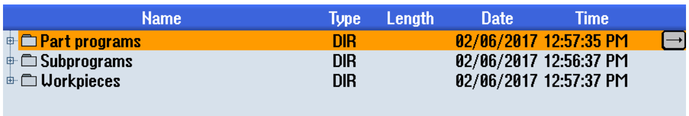
Navigate to the Program manager screen:



The folder structure that was created by the Global part program memory.

This is the same as the folder structure in the NC memory area.

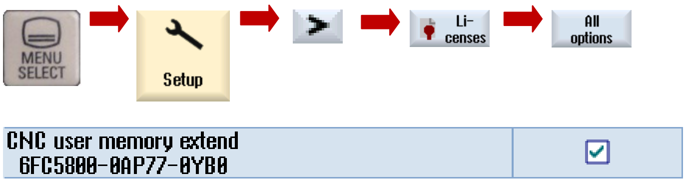




NC Extend Function.

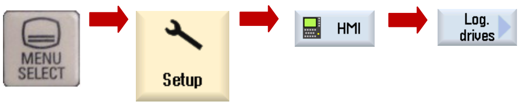
The ***NC Extend*** function releases 100MB of System CF card memory which, when used with the **Global part program memory** feature, behaves exactly like standard NC memory. i.e. there are no restrictions on block search, long program jumps etc. that are found with other methods of executing programs from external storage.

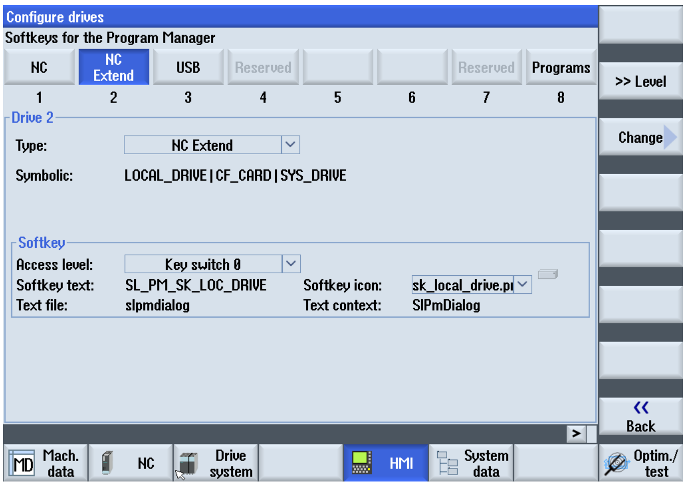
The option - **6FC5800-0AP77-0YB0 -** needs to be set.



Activate the option with an NCK reset:
After the reset it will be possible to the ***NC Extend*** soft-key in the Program Manager area

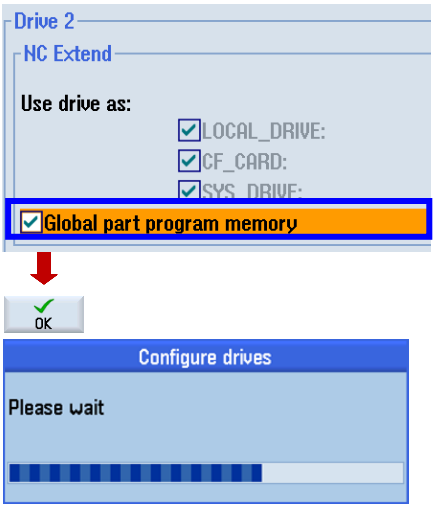
The settings for the NC Extend can be viewed in the Logical drives area.





The default settings, e.g. soft-key text can be changed if required.

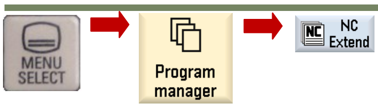
Setting the Global part program memory:



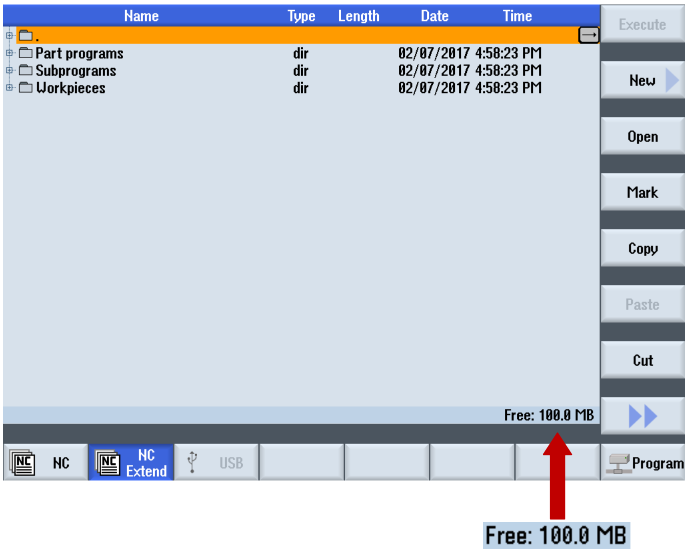
Only one Global Part Program Memory can be active, therefore, If it was already active on another logical drive it will be automatically deactivated for that drive.

The result of the Global part program memory being set is the creation of the folder structure on the NC Extend drive:

Notes



The ***NC Extend*** area has 100MB of additional memory for storage of part programs/sub-programs/workpieces.



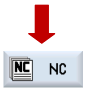
The available space, in this case on the System CF card, can be seen at the bottom right of the screen.

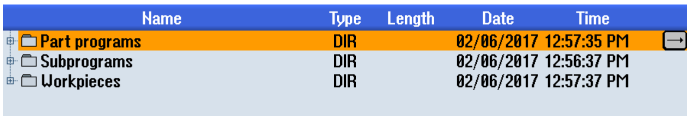
The result of activating the Global part program memory is the creation of three folders which match the default NC memory folders.

Navigate to the Program manager screen:

The folder structure that was created by the Global part program memory.

This is the same as the folder structure in the NC memory area.



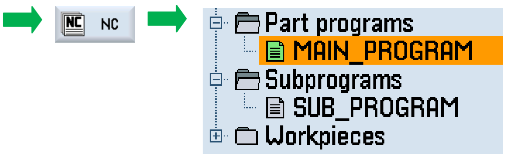


Program Use With EES & NC EXTEND

Part programs, subprograms etc. can now be stored on the external source and used in the same way as the ones in the NC memory.

If, for example, the programs and subprograms are in different memory areas it is necessary to call the subprogram in a certain way, depending on its and the part program's location.

Example 1: Main program and subprogram in the ***standard NC*** memory:



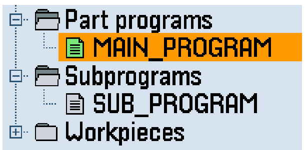
Example: MAIN_PROGRAM.MPF calling the SUB_PROGRAM.SPF.
In this instance, only the name of the subprogram needs to typed in order to execute it from the part program.

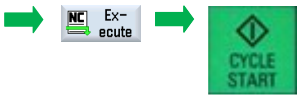
```
NC/MPF/MAIN_PROGRAM
SUB_PROGRAM¶
M30¶
```

The contents of the SUB_PROGRAM.SPF :
A program message (Sub_Program has been called) will be displayed for 10 seconds (G04 F10) when the program is executed.

```
NC/SPF/SUB_PROGRAM.SPF
MSG ("sub_program has ben called")¶
G04 F10¶
M17¶
```

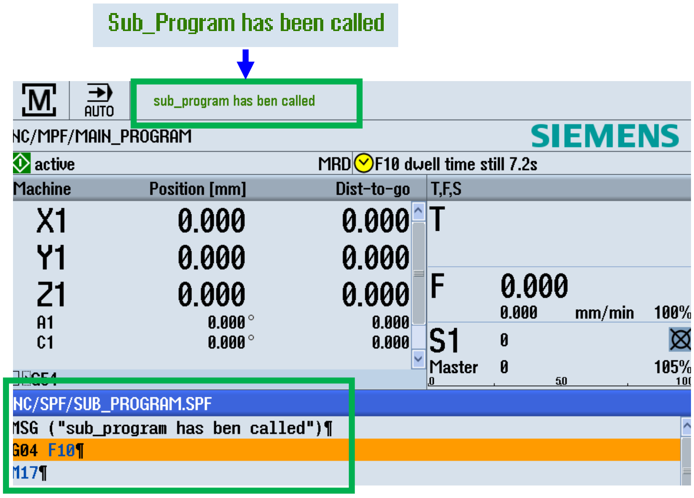
Execute the ***MAIN_PROGRAM.MPF*** program:





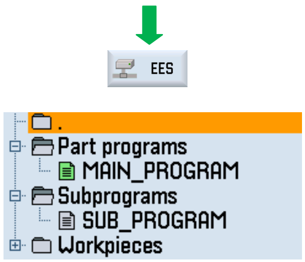
Notes

The ***SUB_PROGRAM.SPF*** will be called and will run for 10 seconds:



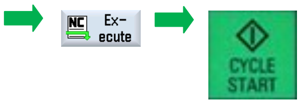
Example 2:

Main program in the ***EES*** memory, subprogram in the ***EES*** memory:



The call to the **SUB_PROGRAM.SPF** can be seen below:

```
EES/spf.dir/SUB_PROGRAM.SPF
MSG ("sub_program has ben called")¶
G04 F10¶
M17¶
```



The subprogram is called:

Sub_Program has been called

Note:

If a subprogram exists in both the NC and EES/NC Extend memory, the one in the NC memory will be called unless the path is specified.

A035 : END

Notes