Industrial Hydraulics | Electric Drives and Controls | Linear Motion Assembly Technologies | Pneumatics | Service Automation | Mobile Hydraulics

Rexroth Bosch Group

Rexroth PNC
Description of Functions

1070 073 871
Edition 02

Function description, V7.3

Title Rexroth PNC
Description of Functions

Type of Documentation Function description

Document Typecode **DOK-PNC***-NC*FUNCTION-FK02-EN-P**

Purpose of Documentation This manual contains information on the following:

- basic steps required for system start-up and
- the available functions of the PNC.

Record of Revisions

Description	Release Date	Notes
DOK-PNC***-NC*FUNCTION-FK01-EN-P	04.2002	Gültig ab V6.x
DOK-PNC***-NC*FUNCTION-FK02-EN-P	05.2003	Gültig ab V7.3

Validity The specified data is for product description purposes only and may not be deemed to be guaranteed unless expressly confirmed in the contract. All rights are reserved with respect to the content of this documentation and the availability of the product.

Published by Bosch Rexroth AG
Postfach 11 62
D-64701 Erbach
Berliner Straße 25
D-64711 Erbach
Tel.: +49 (0) 60 62/78-0
Fax: +49 (0) 60 62/78-4 28
Abt.: BRC/ESM11 (WE)

Contents

Page

1	**Safety Instructions**	**1–1**
1.1	**Intended use**	**1–1**
1.2	**Qualified personnel**	**1–2**
1.3	**Safety markings on products**	**1–3**
1.4	**Safety instructions in this manual**	**1–4**
1.5	**Safety instructions for the described product**	**1–5**
1.6	**Documentation, software release and trademarks**	**1–7**
2	**General**	**2–1**
3	**System**	**3–1**
3.1	**Direct automatic execution**	**3–1**
3.1.1	The PC control panel is used as a file server	3–2
3.1.2	Any remote Windows computer is used as a file server:	3–3
3.2	**Default state**	**3–6**
3.2.1	Assigning and activating	3–7
3.3	**Inch/metric switching**	**3–8**
3.3.1	Assigning	3–9
3.3.2	Unit of measure switching	3–10
3.4	**Rotary axis speed scaling**	**3–11**
3.4.1	Assigning	3–11
3.5	**Adjusting circle radiuses**	**3–12**
3.5.1	Assigning	3–13
4	**Drives (axes, spindles)**	**4–1**
4.1	**Linear axis**	**4–1**
4.1.1	Assigning	4–1
4.2	**Rotary axis**	**4–2**
4.2.1	Assigning	4–2
4.3	**Modulo axis (endless axis)**	**4–3**
4.3.1	Assigning	4–4
4.4	**Linear modulo axis**	**4–5**
4.4.1	Assigning	4–5
4.5	**Hirth axis**	**4–6**
4.5.1	Assigning	4–7
4.6	**Gantry axes**	**4–9**
4.6.1	Assigning	4–16
4.6.2	Activating	4–22
4.6.3	Deactivating	4–23

Contents

4.7	**Approaching the reference point**	**4–24**
4.7.1	Assigning	4–27
4.7.2	Activating	4–29
4.8	**Drive-controlled interpolation**	**4–31**
4.8.1	Assigning	4–32
4.8.2	Activating	4–33
4.8.3	Deactivating	4–33
4.9	**Axis transfer**	**4–34**
4.9.1	Assigning	4–41
4.9.2	Activating	4–42
4.9.3	Deactivating	4–50
4.10	**Spindle**	**4–51**
4.10.1	Assigning	4–52
4.11	**Spindle/C axis**	**4–53**
4.11.1	Assigning	4–55
4.11.2	Switching over from spindle mode to C axis mode.	4–56
4.11.3	Switching over from C axis mode to spindle mode.	4–57
4.12	**Spindle/turret axis with motor encoder**	**4–58**
4.12.1	Assigning	4–61
4.12.2	Synchronization after control unit start-up	4–62
4.12.3	Determining the turret axis reference point	4–63
4.12.4	Examples: Changing between spindle mode and turret mode	4–63
4.13	**Gear range switching with ”Spindle idling”**	**4–65**
4.13.1	Assigning	4–66
4.13.2	Activating/deactivating	4–66
4.14	**Functions for SERCOS handling**	**4–67**
4.14.1	SERCOS initialization	4–67
4.14.2	Assigning	4–68
4.14.3	Automatic parameter download	4–70
4.14.4	Changing parameters in the drive via SERCOS monitor	4–73
4.14.5	Logging the transmitted SERCOS parameters	4–74
4.14.6	Logging of SERCOS timing	4–75
4.14.7	Downloading SERCOS files in phase 3 ”on demand only”	4–76
5	**Coordinate systems**	**5–1**
5.1	**Basics and overview**	**5–1**
5.1.1	Axis and coordinate names	5–1
5.1.2	Physical and logical axes	5–2
5.2	**Coordinate identifiers**	**5–4**
5.2.1	Pseudo coordinates	5–4
5.2.2	Working range coordinates	5–6
5.3	**Coordinate systems of the control unit**	**5–9**
5.3.1	MCS machine coordinate system	5–10
5.3.2	ACS axis coordinate system	5–11
5.3.3	BCS basis workpiece coordinate system	5–11
5.3.4	PCS program coordinate system	5–14
5.3.5	TCS tool coordinate system	5–14

Contents

5.4	**NC functions for defining coordinate systems**	**5–15**
5.4.1	Shifting the machine zero point (axis zero shift)	5–16
5.4.2	Axis transformation: ACS <–> BCS	5–16
5.4.3	Shifting the workpiece zero point	5–17
5.4.4	Functions for manipulating the programmed contour	5–21
5.5	**Eulerian angle**	**5–22**
5.6	**Shifts**	**5–28**
5.6.1	Axis zero shifts (Gx53-Gx59)	5–30
5.6.2	Structure of axis zero shift tables	5–34
5.6.3	Activating axis ZS table	5–38
5.6.4	External axis zero shift (G160, G260, G360)	5–39
5.6.5	G138 workpiece position compensation	5–41
5.6.6	G168 an G268 program coordinate shift	5–41
5.6.7	G60 programmed contour shift	5–41
5.6.8	G352...G359 inclined plane	5–42
5.6.9	Structure of ”inclined plane” tables	5–45
5.6.10	Activating “inclined plane tables”	5–46
5.6.11	Assigning shifts	5–47
5.6.12	Activating shifts	5–49
5.6.13	Deactivating shifts	5–50
5.7	**Working range coordinate programming**	**5–51**
5.7.1	Position of the tool tip (TCP)	5–53
5.7.2	Tool orientation (vector and tensor)	5–54
5.7.3	Orientation movement of the tool	5–62
5.7.4	Assigning	5–74
5.7.5	Activating	5–75
5.7.6	Deactivating	5–76
5.8	**Axis transformations**	**5–77**
5.8.1	5 axis transformation	5–80
5.8.2	6 axis transformation	5–90
5.8.3	Polar coordinate transformation/end face machining	5–103
6	**Precision / Dynamics**	**6–1**
6.1	**Maximum axis step change**	**6–1**
6.1.1	Assigning	6–2
6.1.2	Activating	6–2
6.1.3	Deactivating	6–2
6.2	**Path acceleration limitation**	**6–3**
6.2.1	Assigning	6–3
6.3	**Precision programming**	**6–4**
6.3.1	Assigning	6–6
6.3.2	Activating	6–6
6.3.3	Deactivating	6–6
6.4	**Shape**	**6–7**
6.4.1	Assigning	6–9
6.4.2	Activating	6–10
6.4.3	Deactivating	6–10

Contents

6.5	**Path slope**	**6–11**
6.5.1	Activating	6–12
6.5.2	Deactivating	6–12
6.6	**Look-ahead function**	**6–13**
6.7	**Jerk-limited velocity profile generation**	**6–14**
6.7.1	Assigning	6–17
6.7.2	Activating	6–17
6.7.3	Deactivating	6–17
6.8	**Circle compensation at the quadrant transition**	**6–18**
6.8.1	Assigning	6–20
6.8.2	Activating/deactivating	6–22
6.9	**Dynamic limit values**	**6–23**
6.9.1	Assigning	6–24
7	**Operation**	**7–1**
7.1	**Control reset**	**7–1**
7.1.1	Activating	7–2
7.2	**Jogging in machine coordinates**	**7–3**
7.2.1	Assigning	7–5
7.2.2	Activating	7–5
7.3	**Jogging in workpiece coordinates**	**7–7**
7.3.1	Assigning	7–9
7.3.2	Activating	7–10
7.3.3	Deactivating	7–11
7.4	**Handwheel**	**7–12**
7.4.1	Assigning	7–14
7.4.2	Activating	7–15
7.4.3	Deactivating	7–15
7.5	**Online correction with handwheel (in workpiece coordinates)**	**7–16**
7.5.1	Assigning	7–21
7.5.2	Activating	7–22
7.5.3	Deactivating	7–22
7.6	**Override**	**7–23**
7.6.1	Assigning	7–25
7.6.2	Activating	7–26
7.6.3	Deactivating	7–27
7.7	**Automatic program reselection**	**7–28**
7.7.1	Activating/deactivating	7–28
7.8	**Canceling the distance to go**	**7–29**
7.8.1	Activating	7–29
7.9	**Remove and return to the contour**	**7–30**
7.9.1	Options for returning to contour	7–32
7.9.2	Assigning	7–36
7.9.3	Activating	7–36

Contents

8	**Tool compensations**	**8–1**
8.1	**Basics**	**8–1**
8.1.1	Tool reference points	8–1
8.1.2	Tool coordinate system	8–2
8.1.3	Tool coordinate system with active axis transformation	8–2
8.1.4	Assigning	8–9
8.1.5	Activating	8–9
8.1.6	Deactivating	8–9
8.2	**Geometry compensations**	**8–10**
8.2.1	Overview	8–10
8.2.2	Standard compensation	8–13
8.2.3	External compensation (1st external compensation)	8–14
8.2.4	General compensation (2nd external compensation)	8–17
8.2.5	Overview: Methods of tool compensation	8–24
8.2.6	Assigning	8–25
8.2.7	Activating	8–28
8.2.8	Deactivating	8–29
8.2.9	Structure of geometry compensation tables	8–30
8.2.10	Activating	8–31
8.3	**Compensation switchover**	**8–32**
8.3.1	Assigning	8–34
8.3.2	Activating	8–35
8.3.3	Deactivating	8–36
8.4	**Path compensation (cutter radius compensation)**	**8–37**
8.4.1	Assigning	8–39
8.4.2	Activating	8–39
8.4.3	Deactivating	8–40
9	**Safety / Monitoring**	**9–1**
9.1	**Electronic limit switches (software limit switches)**	**9–1**
9.1.1	Activating	9–2
9.2	**Torque reduction**	**9–3**
9.2.1	Assigning	9–4
9.2.2	Activating/deactivating	9–4
9.3	**Limit rapid travel**	**9–5**
9.3.1	Assigning	9–6
9.3.2	Activating/deactivating	9–6
9.4	**Axis position monitoring**	**9–7**
9.4.1	Activating	9–8
9.5	**Distance to end point**	**9–9**
9.5.1	Assigning	9–9
9.5.2	Deactivating	9–9
9.6	**Area control**	**9–10**
9.6.1	Assigning	9–13
9.6.2	Activating	9–13
9.6.3	Deactivating	9–13

Contents

10 Axis compensations 10–1

10.1 Leadscrew error compensation (LSEC) 10–1
10.1.1 Determining the compensation values of an axis 10–1
10.1.2 Entering compensation values 10–3
10.1.3 Activating 10–6
10.1.4 Deactivating 10–6

10.2 Cross compensation (CCOMP) 10–7
10.2.1 Assigning 10–8
10.2.2 Activating 10–13
10.2.3 Deactivating 10–14

11 Communication with the PLC 11–1

11.1 Auxiliary functions 11–1
11.1.1 Assigning 11–2
11.1.2 Activating 11–6

12 Technology / machining functions 12–1

12.1 Oscillating axis 12–1
12.1.1 Activating 12–2
12.1.2 Deactivating 12–2

12.2 Punching and nibbling 12–3
12.2.1 Assigning 12–5
12.2.2 Activating 12–6
12.2.3 Deactivating 12–6

12.3 Tangential tool guidance 12–7
12.3.1 Activating 12–8
12.3.2 Deactivating 12–9

12.4 Tangential tool orientation 12–10
12.4.1 Activating 12–11
12.4.2 Deactivating 12–11

12.5 Rounding of corners 12–12
12.5.1 Assigning 12–13
12.5.2 Activating 12–13
12.5.3 Deactivating 12–13

12.6 Switching NC blocks via high-speed signal 12–14
12.6.1 Assigning 12–17
12.6.2 Activating 12–17

12.7 Tapping 12–18
12.7.1 Assigning 12–23
12.7.2 Activating 12–24
12.7.3 Deactivating 12–25

12.8 Probe 12–26
12.8.1 Assigning 12–27

Contents

12.9	**Digitizing (recording measured values)**	**12–29**
12.9.1	Assigning	12–42
12.9.2	Activating	12–45
12.9.3	Deactivating	12–45
12.9.4	The recording file	12–46
12.10	**Axis distance control for the digitizing process**	**12–49**
12.10.1	Monitoring deviations	12–54
12.10.2	Assigning	12–57
12.10.3	Switching on/off	12–58
12.11	**Spline**	**12–59**
12.11.1	Spline with coefficient programming	12–63
12.11.2	Spline with point programming and tangential transitions (C1-continuous cubic splines)	12–67
12.11.3	Spline with point programming and continuous bend transitions (C2-continuous cubic splines)	12–73
12.11.4	Spline with control point programming (B spline)	12–77
12.11.5	Programming spline	12–79
12.11.6	Assigning	12–82
12.11.7	Activating	12–82
12.11.8	Deactivating	12–82
12.12	**Path-speed-dependent laser power control**	**12–83**
12.12.1	Assigning	12–86
12.12.2	Activating	12–86
12.12.3	Deactivating	12–88
A	**Appendix**	**A–1**
A.1	**Abbreviations**	**A–1**
A.2	**Memory demand for function extensions**	**A–2**
A.3	**Index**	**A–3**

Contents

Notes:

1 Safety Instructions

Please read this manual before commissioning the PNC or activating new functions. Store this manual in a place to which all users have access at any time.

1.1 Intended use

This manual contains all information required for the proper use of the control units. For reasons of clarity, however, it cannot contain each and every detail about each and all combinations of functions. Likewise, it is impossible to consider each and any aspect of integration or operation.

The PNC controls serve as

- activate feed drives, spindles and auxiliary axes of a machine tool via SERCOS interface for the purpose of guiding a processing tool along a programmed path to process a workpiece (CNC). Furthermore, I/O components are required for the integrated PLC which – in communication with the actual CNC – controls the machine processing cycles holistically and acts as a technical safety monitor.
- program contours and the processing technology (path feedrate, spindle speed, tool change) of a workpiece.

Any other application is deemed improper use!

The products described hereunder

- have been developed, manufactured, tested and documented in compliance with the safety standards. These products pose no danger to persons or property if they are used in accordance with the handling stipulations and safety notes prescribed for their configuration, mounting, and proper operation.
- comply with the requirements of
  - the EMC Directives (89/336/EEC, 93/68/EEC and 93/44/EEC)
  - the Low-Voltage Directive (73/23/EEC)
  - the harmonized standards EN 50081-2 and EN 50082-2
- are designed for operation in industrial environments, i.e.
  - no direct connection to public low-voltage power supply,
  - connection to the medium- or high-voltage system via a transformer.

  In residential environments, in trade and commerce as well as small enterprises class A equipment may only be used if the following warning is attached:

☞ **This is a Class A device. In a residential area, this device may cause radio interference. In such case, the user may be required to introduce suitable countermeasures, and to bear the cost of the same.**

The faultless, safe functioning of the product requires proper transport, storage, erection and installation as well as careful operation.

1.2 Qualified personnel

The requirements as to qualified personnel depend on the qualification profiles described by ZVEI (central association of the electrical industry) and VDMA (association of German machine and plant builders) in:
Weiterbildung in der Automatisierungstechnik
edited by: ZVEI and VDMA
MaschinenbauVerlag
Postfach 71 08 64
D-60498 Frankfurt.

The present manual is designed for

- NC project engineers and commissioning personnel.

These persons need special knowledge of

- the possible configurations of PNC and
- the possible adjustments of the PNC for use with a specific machine tool.

Programming, start and operation as well as the modification of programs or program parameters may only be performed by properly trained personnel! This personnel must be able to judge potential hazards arising from programming, program changes and in general from the mechanical, electrical, or electronic equipment.

Interventions in the hardware and software of our products, unless described otherwise in this manual, are reserved to our specialized personnel.
Tampering with the hardware or software, ignoring warning signs attached to the components, or non-compliance with the warning notes given in this manual may result in serious bodily injury or material damage.

Only electrotechnicians as recognized under IEV 826-09-01 (modified) who are familiar with the contents of this manual may install and service the products described.

Such personnel are

- those who, being well trained and experienced in their field and familiar with the relevant norms, are able to analyze the jobs being carried out and recognize any hazards which may have arisen.
- those who have acquired the same amount of expert knowledge through years of experience that would normally be acquired through formal technical training.

With regard to the foregoing, please note our comprehensive range of training courses. Please visit our website at
http://www.boschrexroth.com
for the latest information concerning training courses, teachware and training systems. Personal information is available from our Didactic Center Erbach,
Telephone: (+49) (0) 60 62 78-600.

1.3 Safety markings on products

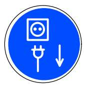
Warning of dangerous electrical voltage!

Warning of danger caused by batteries!

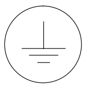
Electrostatically sensitive components!

Components sensitive to electrostatic discharge!

Warning of hazardous light emissions
(optical fibre cable emitters)



Disconnect mains power before opening!



Lug for connecting PE conductor only!

Connection of shield conductor only

1.4 Safety instructions in this manual

DANGEROUS ELECTRICAL VOLTAGE
This symbol is used to warn of a **dangerous electrical voltage.** The failure to observe the instructions in this manual in whole or in part may result in **personal injury**.

DANGER
This symbol is used wherever insufficient or lacking compliance with instructions may result in **personal injury**.

CAUTION
This symbol is used wherever insufficient or lacking compliance with instructions may result in **damage to equipment or data files**.

☞ This symbol is used to draw the user's attention to special circumstances.

★ This symbol is used to mark descriptions of tasks to be performed by the user.
If it is important that steps are taken in the correct sequence, action prompts may be shown with serial number instead of being marked "★".
Example:

1. Carry out the first step.
2. Carry out the second step.

1.5 Safety instructions for the described product

DANGER
Danger of life through inadequate EMERGENCY-STOP devices!
EMERGENCY-STOP devices must be active and within reach in all system modes. Releasing an EMERGENCY-STOP device must not result in an uncontrolled restart of the system!
First check the EMERGENCY-STOP circuit, then switch the system on!

DANGER
Incorrect or undesired axis movement!
First, new programs should be tested carefully without axis movement! For this purpose, the control offers the possibility of inhibiting axis movements and/or auxiliary function outputs by appropriate softkeys in the 'Automatic' mode.

DANGER
Incorrect or undesired control unit response!
Rexroth accepts no liability for damage resulting from the execution of an NC program, an individual NC block or the manual movement of axes!

Furthermore, Rexroth accepts no liability for consequential damage which could have been avoided by programming the PLC appropriately!

DANGER
Retrofits or modifications may adversely affect the safety of the products described!
The consequences may include severe injury, damage to equipment, or environmental hazards. Possible retrofits or modifications to the system using third-party equipment therefore have to be approved by Rexroth.

DANGEROUS ELECTRICAL VOLTAGE
Unless described otherwise, maintenance works must be performed on inactive systems! The system must be protected against unauthorized or accidental reclosing.

Measuring or test activities on the live system are reserved to qualified electrical personnel!

DANGER
Tool or axis movements!
Feed and spindle motors generate very powerful mechanical forces and can accelerate very quickly due to their high dynamics.

- **Always stay outside the danger area of an active machine tool!**
- **Never deactivate safety-relevant functions!**
- **Report any malfunction of the unit to your servicing and repairs department immediately!**

CAUTION
Use only spare parts approved by Rexroth!

CAUTION
Danger to the module!
All ESD protection measures must be observed when using the module! Prevent electrostatic discharges!

The following protective measures must be observed for modules and components sensitive to electrostatic discharge (ESD)!

- Personnel responsible for storage, transport, and handling must have training in ESD protection.
- ESD-sensitive components must be stored and transported in the prescribed protective packaging.
- ESD-sensitive components may only be handled at special ESD-workplaces.
- Personnel, working surfaces, as well as all equipment and tools which may come into contact with ESD-sensitive components must have the same potential (e.g. by grounding).
- Wear an approved grounding bracelet. The grounding bracelet must be connected with the working surface through a cable with an integrated 1 MΩ resistor.
- ESD-sensitive components may by no means come into contact with chargeable objects, including most plastic materials.
- When ESD-sensitive components are installed in or removed from equipment, the equipment must be de-energized.

1.6 Documentation, software release and trademarks

Documentation

This manual contains information on the following:

- basic steps required for system start-up and
- the available functions of the PNC.

Overview of available documentation	Part no.		
	German	English	French
PNC-R – Connectivity Manual for project engineering and maintenance	1070 073 704	1070 073 736	–
PNC-R – Software installation	1070 073 796	1070 073 797	–
PNC-P – Connectivity Manual	1070 073 880	1070 073 881	–
PNC-P – BF2xxT/BF3xxT Control Panel Connectivity Manual	1070 073 814	1070 073 824	–
PNC-P – Software installation	1070 073 882	1070 073 883	–
Description of functions	1070 073 870	1070 073 871	–
MACODA Operation and configuration of the machine param-eters	1070 073 705	1070 073 742	–
Operating instructions – Standard operator interface	1070 073 726	1070 073 739	1070 073 876
Operating instructions – Diagnostics Tools	1070 073 779	1070 073 780	–
Error Messages	1070 073 798	1070 073 799	–
PLC project planning manual, Software interfaces of the integrated PLC	1070 073 728	1070 073 741	–
iPCL system description and programming manual	1070 073 874	1070 073 875	–
ICL700 system description (PNC-R only), Program structure of the integrated PLC ICL700	1070 073 706	1070 073 737	–
DIN programming manual for programming to DIN 66025	1070 073 725	1070 073 738	–
CPL programming manual	1070 073 727	1070 073 740	1070 073 877
CPL Debugger Operating Instructions	1070 073 872	–	–
Tool Management – Parameterization	1070 073 782	1070 073 793	–
Software PLC Development environment for Windows NT	1070 073 783	1070 073 792	–
Measuring cycles for touch-trigger switching probes	1070 073 788	1070 073 789	–
Universal Milling Cycles	–	1070 073 795	–

Release

☞ **This manual refers to the following versions:**
Software release: V7.3

The current release number of the individual software modules can be viewed by selecting the 'Control-Diagnostics' softkey in the 'Diagnostics' operating mode.

The software version of Windows may be displayed as follows:

3. Click the right mouse button on the **My Computer** icon on your desktop.
4. Select **Properties**.

Trademarks

All trademarks of software installed on Rexroth products upon delivery are the property of the respective manufacturer.

Upon delivery, all installed software is copyright-protected. The software may only be reproduced with the approval of Rexroth or in accordance with the license agreement of the respective manufacturer.

MS-DOS® and Windows™ are registered trademarks of Microsoft Corporation.

PROFIBUS® is a registered trademark of the PROFIBUS Nutzerorganisation e.V. (user organization).

SERCOS interface™ is a registered trademark of the SERCOS interface Joint VDW/ZVEI Working Committee.

2 General

This manual is designed to provide support

- in the application/activation of various control functions.

This manual is focused on the necessary procedures for parameter assignment.
As a prerequisite, all hardware components and their cabling, i.e. CNC, drives, PLC and power supply modules must have been installed, integrated and tested for faults in accordance with the application documentation. In particular, this includes all safety-relevant devices, e.g. EMERGENCY STOP circuits, release contacts, limit switches, etc.!
In the ideal case, the software of all SERCOS drives has been commissioned. However, this is not absolutely necessary because the control unit is able to configure all connected drives by downloading the required parameters during start-up. In this case, properly adapted initialization files must be available in the control unit.
To adapt the initialization files, you should have the relevant documentation available (i.e. tables showing the SERCOS parameter settings required for every drive, drive documentation, etc.). For more detailed information on the SERCOS interface, see page 4–67.

For reasons of clarity, all functions of the PNC are grouped according to subject (see table of contents).

For every function, you will find information on

- its purpose or intended use,
- its conditions / restrictions of use, and
- modification options (e.g. by MACODA parameter setting, DIN/CPL instruction, file, or interface).

☞ **In part, a certain memory size is required in the NC for the application of functions. For more information, please refer to ”Memory demand for function extensions” in the annex.**

Because many functions involve more than one issue (MACODA, DIN, interface, etc.), they may be described in full or in part also in other PNC manuals.

☞ **Therefore, please also refer to other relevant manuals.**
The following documentation supplements this manual:

MACODA parameter:	”Configuration Parameters and MACODA Parameter Description”
G/M functions:	”DIN Programming instructions”
CPL instructions:	”CPL Programming Manual”
For interface signals:	”PLC Project Planning”

In addition, have the drive documentation ready.

General

To carry out the various process steps, you must be familiar with

- the PNC standard operator interface,
- the Windows graphical user interface of the PC operating panel, and
- the MACODA software tool.

With MACODA, you have write access to the NC configuration parameters if you have signed on as a ”machine setter” or an ”MTB” (DIAGNOSTICS ▶ SET ▶ SET USER). Therefore, you must be familiar with the operation of MACODA!

DANGER
Improper handling by inadequately trained or unskilled personnel may cause serious damage to the machine, loss of data or even personal injury!

Therefore, properly trained personnel only may start and operate drives or set or modify configuration parameters!
Such personnel must be able to recognize and avoid risks arising from changed parameter settings or generally involved in the mechanical, electrical or electronic equipment.

Rexroth accepts no liability for damage, including consequential damage, caused by incorrect or unskilled handling of equipment.

---

3 System

3.1 Direct automatic execution

Function:

Allows the execution of part programs stored on any networked computer (the" file server" hereinafter).
This allows you to use part programs that are too big for the part program memory of the control unit. Part programs are now limited in size and number only by the disk memory allocated by the file server.
In addition, this allows central administration of all part programs of the whole machine outfit, making the part programs available to various control units at any time without any distribution activities being required.

Through its integrated NFS client, the control unit can mount up to two externally stored directories to its internal file system by means of the directories "Mount" (/mnt/) and "floppy disk" (/mnt2/). This requires that the "PNC NFS Server" is installed and running on the PC control panel or directly on the file server. This application is part of the "SW OSA3 V7.x.x"PNC-software package supplied on CD.
For installation, please refer to manual "PNC-P Software Installation or "PNC-R Software Installation".

There are 2 different scenarios for working with the "Direct automatic" execution function:

- The PC control panel is used as a file server:
  which means that the "PNC NFS Server" application runs on the PC control panel.
  In this case, either the local drives of the PC control panel or, optionally, the network drives linked to the PC control panel may be used to run the "Direct automatic" execution function.
- Any remote Windows computer is used as a file server:
  which means that the "PNC NFS Server" application runs on any remote networked Windows computer.
  Also in this case, either the local drives of the file server, or, optionally, the network drives of any other networked server may be used to run the "Direct automatic" execution function.
  However, this requires more work in terms of configuration.

Relevant MACODA parameters (MP):

3080 00002: Buffer size required for execution through NFS.

3080 00001: Search path for subprograms.

3.1.1 The PC control panel is used as a file server

Precondition:

- The control unit and the PC control panel must be properly interconnected, configured and booted. The PNC operator interface must have been started.
- The ”PNC NFS Server” application must have been installed on the PC control panel. This application is part of the ”SW OSA3 V7.x.x” PNC software package supplied on CD and is usually installed together with the whole package.
  For details, please refer to manual ”PNC-P Software Installation” or “PNC-R Software Installation”.

1. If network drives are to be used, please start the Windows Explorer on the PC control panel first.
   In the left pane of the Explorer window, open the branches ”Network neighborhood” and ”Entire network” down to the required level. Here, all workgroups, computers and their shared drives are displayed.
   Select the drive and right-click ”Map network drive”.
   Select a letter for the local drive. The checkbox ”Reconnect at logon” must be selected.
   Click ”OK”.
2. Start the ”PNC NFS Server” application on the PC control panel. (START ▶ PROGRAMS ▶ BOSCH PC-PANEL ▶ PNC NFS SERVER).
3. Start the”PNC configuration” application on the PC control panel. (START ▶ PROGRAMS ▶ BOSCH PC-PANEL ▶ PNC CONFIGURATION).
4. Select T3SWCONFIG ▶ SELECT NFS-MOUNT-DIRECTORY.
5. Select the control unit of your choice.
6. Select the directories to be mounted.
   For more detailed information on this procedure, see ”PNC-P Software Installation” manual or “PNC-R Software Installation” manual.
7. Confirm the selected directories with OK.
   Confirm the info dialogue that follows with OK and wait until the main window of the configuration software is displayed again.
   Then, close the ”PNC Configuration” application.
8. Move the PNC operator interface to the foreground.
9. If required, adjust the buffer size with MP 3080 00002.
   The required buffer size depends on various factors (e.g., high network load, a long feed with very short traversing blocks) and is difficult to assess without knowing the local conditions. Therefore, you may have to experiment with different settings. If the machining process falters with the standard buffer size (8 K bytes), try to solve the problem by increasing the buffer size by 1 K byte increments.
10. If required, adjust the search path for sub-programs with MP 3080 00001.

Direct automatic execution

CAUTION
Danger of damage to the machine or the workpiece!
Do not initiate a control RESET during machining.

11. Initiate a control RESET.

3.1.2 Any remote Windows computer is used as a file server:

CAUTION
Improper connections or configurations may cause network failure!
Therefore, we recommend that all network configuration jobs are done by the (system) network administrator only.

Precondition:

- The control unit and the control panel PC must be properly interconnected, configured and booted. The Bosch operator interface is displayed.
- The TCP/IP addresses of the control unit and the PC control panel must be in the same subnet as the respective file server.
  See also the information on ”Configuring TCP/IP” in the ”Network settings” section of the manuals on software installation.
- The computer selected as the file server must be physically integrated into the same network as the control unit and ready for operation.

1. Please ensure that the TCP/IP protocol is installed on the file server and that the TCP/IP address of the file server is in the same subnet as the control unit and the PC control panel.
2. Please ensure that the file server can recognize the connected control unit. For this purpose, the TCP/IP address and the host name of the control unit must be entered in the ”hosts” file of the file server. When entering the host name (max. 8 digits), please note that it is case sensitive!
   In the case of a Windows 95/98 PC, the ”hosts” file is in the Windows system directory. In the case of an NT PC, it is in a subdirectory of the system directory (”...\system32\drivers\etc”).
   This file can be modified with an ASCII editor (e.g., ”Notepad.exe”).

Direct automatic execution

3. If you have not yet installed the ”PNC NFS Server” application on the file server, do it now. This application is part of the ”SW OSA3 V7.x.x”PNC software package supplied on CD.
   For information on its installation, please see the section on ”Setup to the control panel PC”, under ”Software installation” in the manuals on software installation.
   Use the setup option ”Customer Setup” and select only the ”NFS Server” component. When you are requested to enter the name of the target control unit, enter the host name of the control unit (e.g., ”pnc”. Please note that this entry is case sensitive!).
4. When the installation is finished, reboot the file server.
   The NFS server should be running. This is indicated by the corresponding icon on the taskbar. If the icon is not displayed, you can also start the NFS server manually. START ▶ PROGRAMS ▶ BOSCH PC-PANEL ▶ PNC NFS SERVER.
5. Specify the directories which the NFS server is to make available to the NC. Select
   START ▶ PROGRAMS ▶ BOSCH PC-PANEL ▶ EDIT EXPORTS.
   The ASCII file ”export.us” will be loaded and displayed.

   Example: contents of ”export.us”

   ```
   c:\typ3pcp\cycles  typ3osa
   c:\typ3pcp\cncfiles  typ3osa
   ```

   - Each directory to be exported must be entered on a separate line.
   - No directory to be exported must be a subdirectory of any other directory to be exported.
   - Enter at least 1 blank between the path of the desired directory and the host name of the control unit.
   - Save the modified file.

☞ **The NFS server can make more than 2 directories available. However, each control unit can mount no more than 2 directories at the most.**
In this way, several networked control units can be served by a single NFS server.

6. Right-click the NFS server icon on the taskbar and select ”Reload export.us”. This has the effect that changes made in the ”export.us” file are adopted by the NFS server.
   You can verify whether the changes made have been correctly adopted:
   Right-click the NFS server icon on the taskbar and select ”About”.
7. Move the PNC operator interface to the foreground.
8. Change to the ”User FEPROM” directory and load the ”hosts file to the editor.
   Make sure that the file contains a line with the TCP/IP address and the host name of the file server. If there is no such line, please add it.
   Format: `<IP address> <host name>`
   Example: `192.168.100.9  server11`

Direct automatic execution

9. Save the changed file as ”hosts” in the directory ”System (/etc)”. In order to make a backup of the changed file, copy it to the ”User FEPROM” directory.
10. Change to the ”User FEPROM” directory and load the ”startup” file to the editor.

Example: contents of ”startup”

```
NFSMOUNT server11:/c/typ3pcp/cycles /mnt rw c–
NFSMOUNT server11:/c/typ3pcp/cncfiles /mnt2 rw
```

Host name of the PC on which the NFS server is running. Must be specified in the ”hosts” PNC file including the TCP/IP address!

Directory which the control unit is to obtain from the NFS server. Please note that the syntax with regard to the signs ”:” and ”/” differs from Windows!.

- Each directory to be mounted must be entered on a separate line.
- Comment lines are indicated by the ”#” character at the beginning of the line.
- Enter at least 1 blank between the individual statements.
- A maximum of 2 directories can be mounted on each PNC control unit (2 mount points):
  Parameter ”/mnt”: the directory is addressed as ”Mount” (/mnt/) in the PNC file system.

  Parameter ”/mnt2”: the directory is addressed as ”Disk” (/mnt2/) in the PNC file system.
- ”rw”: access authorization. ”r”: read; ”w”: write/delete
- ”–c”: enable the cache. Do not use with removable disks!

11. Save the changed file in the ”User FEPROM” directory. In this way, it is saved permanently as a backup copy. To have a work copy of this changed file available, save it also as ”startup” in the root directory.
12. If required, adjust the buffer size with MP 3080 00002.
13. If required, adjust the search path for sub-programs with MP 3080 00001.

CAUTION
Danger of damage to the machine or the workpiece!
Do not initiate a control RESET during machining.

---

14. Initiate a control RESET.

3.2 Default state

Function:

Ensures that the active NC functions of the control unit are reset to a precisely defined state (mode). The function has a modal effect.
For each assigned channel, you can define what its NC functionality is to be like after the following "events":

- State after start-up
  (after power-up or RESET; ⇒ MP 7060 00010)
- State upon M30
  (after the end of a program; ⇒ MP 7060 00020)
- State upon a control reset.
  (AUTOMATIC ▶ CHANNEL CONTROL RESET;
  ⇒ MP 7060 00020)
- State upon a "System control reset".
  (DIAGNOSTICS ▶ RESET FUNCTIONS ▶ SYSTEM CONTROL RESET; ⇒ MP 7060 00020)

Start-up is always followed by a "System control reset".

☞ **For information on "Control reset" and "System control reset", please refer to page 7–1 ff.**

Relevant MACODA parameters (MP):

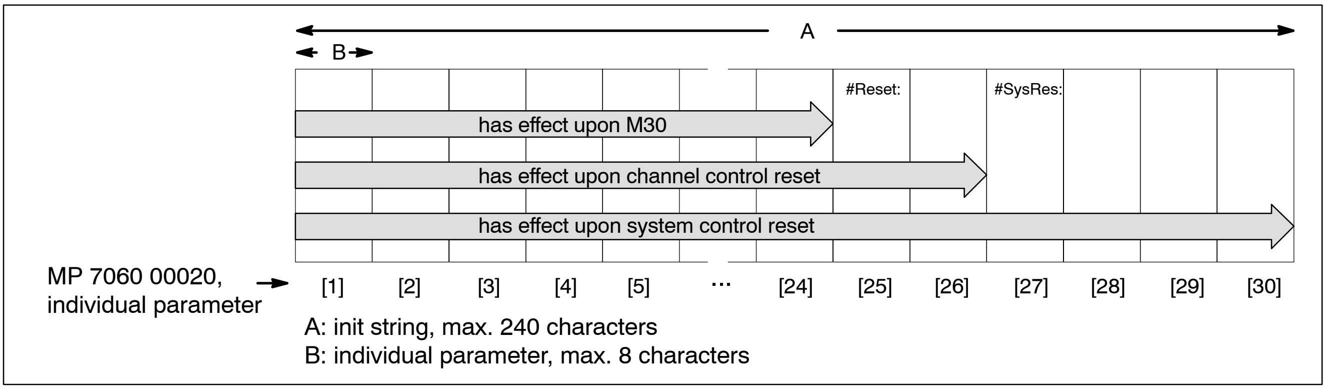
7060 00010: Default state upon start-up.
Serves to define the functions that are to be activated upon start-up, but not upon "System control reset", "Control reset", or "M30".

7060 00020: Default state upon a control reset.
Serves to define the functions that are to be activated upon "System control reset", "Control reset", or "M30".

The desired default state can be defined in the two channel-specific MACODA parameters by the "init string".
As a maximum, an init string may have 240 characters for a total of 30 individual parameters of 8 characters, each. It contains the syntax of all the NC functions that are to be activated upon the occurrence of a specific event.

While the whole init string of MP 7060 00010 is taken into account when the control unit is started up, the init string of MP 7060 00020 is interpreted by the PNC only up to a certain point. This point depends on the event that has occurred and is marked by the position of the keywords "#Reset:" and "#SysRes:" within the init string.

Default state



☞ **Auxiliary functions with compulsory acknowledgement that are entered in MP 7060 00010 may inhibit control unit start-up!**

3.2.1 Assigning and activating

1. Adjust MP 7060 00010, or 7060 00020.
   Please note: many NC functions are assigned to ”Groups”. These NC functions remain active once they have been selected until they are specifically deactivated or ”superseded” by another NC function of the same group being selected.

☞ **For information on groups, see ”Overview of G instructions” in the annex of the DIN Programming Instructions.**

CAUTION
Danger of damage to the machine or the workpiece!
Do not initiate a system control reset during machining.

---

2. Initiate a system control reset (DIAGNOSTICS ▶ RESET FUNCTIONS ▶ SYSTEM CONTROL RESET).

3.3 Inch/metric switching

For each operating mode, the control unit can switch between the two units of measure (inch or metric) used in the various operating modes:

- for the NC operating modes ”Manual” and ”Automatic”: via G70 (inch) or G71 (metric).
- for the NC operating modes ”Jog” and ”Traverse to reference point”: via the axis-related interface signal NC-I0.3, ”Inch incr. step”.
  In combination with ”Traverse to reference point”, the position display only is switched to inch.

Switching with G70/G71 while in NC operating modes ”Manual” or ”Automatic” has the effect that the control unit will interpret all subsequent translatory positions, speed and acceleration values programmed as inch or metric units of measure.
When the units of measure are switched, the control unit converts the positions and feedrate values currently displayed into the programmed unit of measure.
Depending on whether G70 or G71 is active on a channel, inch or metric values will be displayed with CPL functions.

In the NC operating modes ”Jog” or ”Traverse to reference point” , setting the interface signal NC-I0.3 has the effect that the 4 incremental steps (1000, 100, 10, 1 increments) permanently stored in the system or the incremental step as defined by the user (x increments; MP 1015 00002) will be interpreted by the control unit as inch units of measure with the weighting ”0.0001 * (MP 1015 00001) inch”. The effect is limited exclusively to synchronous and asynchronous linear axes. This switching of units of measure has no effect on rotary or endless axes.
Nor does it have an effect on any of the jogging speeds parameterized.

☞ **Because the incremental steps are used also in handwheel mode, switching the unit of measure by interface signal will also change the weighting of a handwheel increment.**

Restrictions:

- Switching the units of measure by G70/G71 has an effect neither on linear auxiliary aces (asynchronous linear axes) nor on rotary or endless axes (i.e., all axes programmed in degrees).
- Switching the units of measure by interface signal has no effect on rotary or endless axes (i.e., all axes programmed in degrees).
- Inch/metric switching by G70/G71 or by interface signal NC-I0.3 will not change the values of the geometry compensation or zero shift tables. These values will be automatically converted by the control unit exclusively for the purpose of displaying the current compensation values.
  Table input formats can be individually modified as follows:
  - via MP 6020 00021, ”Number of decimal positions for compensation values (metric)”,

Inch/metric switching

  - via MP 6020 00022, ”Number of decimal positions for compensation values (Inch)”, or
  - by using the toggle softkey INCH/METRIC in the table editor.

- Inch/metric switching by G70/G71 or by interface signal NC-I0.3 will not change the data made available to the PLC. All values transferred to the PLC will be in μm or 0.001 degrees.

- Spindle or diagnostics values displayed will not be affected by inch/ metric switching.

Relevant MACODA parameters (MP):

1015 00001:	Axis resolution.
1015 00002:	Variable increment step.
6020 00001:	Number of decimal places for feedrate display (metric).
6020 00002:	Number of decimal places for feedrate display (inch).
6020 00011:	Number of decimal places for axis display (metric).
6020 00012:	Number of decimal places for axis display (inch).

Relevant G functions:

G70:	Inch programming
G71:	Metric programming

Relevant interface signals:

NC-I0.3:	”Incremental step in inch” (axis interface). Switches the unit of measure to ”Inch” while in the NC operating modes ”Jog” or ”Traverse to reference point”.
NC-I1.0:	”Manual feed/ incremental bit 0” (axis interface).
NC-I1.1:	”Manual feed/ incremental bit 1” (axis interface).
NC-I1.2:	”Manual feed/ incremental bit 2” (axis interface).
NC-I1.3:	”Manual feed/ incremental bit 3” (axis interface).
NC-O6.1:	”G70 active” (channel interface). Displays whether G70 is active on the current channel.

Relevant CPL functions:

MCODS(...)	see ”Positions” under MCODS in the CPL Programming Manual

3.3.1 Assigning

1. Adjust MP 6020 00001 through MP 6020 00012, if required.

CAUTION
Danger of damage to the machine or the workpiece!
Do not initiate a system control reset during machining.

2. Initiate a system control reset (DIAGNOSTICS ▶ RESET FUNCTIONS ▶ SYSTEM CONTROL RESET).

3.3.2 Unit of measure switching

In the NC operating modes ”Manual” or ”Automatic”:

1. to switch to ”Inch”, program G70.
   to switch to ”Metric”, program G71.

In the NC operating modes ”Jog” and ”Traverse to reference point”:

Precondition:

- The axis-related interface signal NC-I0.3, ”Inch incr. step” must be generated.

1. NC-I0.3 must be set at the relevant axis interface in order to switch to ”Inch”. To switch back to ”Metric”, reset this signal.

3.4 Rotary axis speed scaling

Function:

Ensures that the control unit specifies the correct speeds of synchronous rotary or endless axes with both metric-based and inch-based programs.
This allows to execute existing ”country-specific” part programs without prior modification.

Restrictions:

- G595 must be active (= all synchronous axes of a channel are taken into account for calculating the path feedrate).
  In order to ensure that G595 is active upon a start-up or a reset, we recommend that ”G595” be included in MP 7060 00010 (start-up) and in MP 7060 00020 (control reset).
- This function has an effect on synchronous axes of the ”rotary” or ”endless” movement types only (⇒ MP 1003 00004).
- This function relates exclusively to speed, not to position.

Relevant MACODA parameters (MP):

7040 00110: Rotary axis speed scaling with G70/G71.

Relevant G functions:

G595: All axes on a channel are taken into account for calculating the feedrate.

3.4.1 Assigning

1. Adapt MP 7040 00110. Please note:
   - If the control unit is installed at a location where only metric part programs are used (G71 is active; e.g., in Europe):
     7040 00110 [1]: 1
     7040 00110 [2]: 1
   - If the control unit is installed at a location where only inch-based part programs are used (G70 is active; e.g., in the USA):
     7040 00110 [1]: 2
     7040 00110 [2]: 2
   - If the control unit is installed at a location where both metric-based and inch-based part programs are used:
     7040 00110 [1]: 2
     7040 00110 [2]: 1

CAUTION
Danger of damage to the machine or the workpiece!
Do not initiate a system control reset during machining.

2. Initiate a system control reset (DIAGNOSTICS ▶ RESET FUNCTIONS ▶ SYSTEM CONTROL RESET).

3.5 Adjusting circle radiuses

Function:

The PNC can avoid runtime errors, which may be caused by inexact statement/calculation of parameters when programming a circular contour segment with G2/G3.
This concerns the programming of circles both by means of their radiuses and end points (radius-based programming) and by means of center points and end points (center-point-based programming).

For radius-based programming:

The parameters that the control unit needs are the radius, the end point and the sense of rotation of the circle. The current position is always taken to be the starting position of the circular movement.
The only problems that may occur with the internal calculation of the circular path relate to semicircles. For semicircles, the radius to be entered must be exactly half the distance between the starting position and the end point.
If the programmed radius is smaller than this (e.g., due to inexact entry or calculation during data preprocessing), a runtime error will occur unless the radius is corrected.
To avoid a runtime error in this case you may define a tolerance zone (⇒ MP 705000030). Within this tolerance zone, the PNC will automatically enlarge any radius that is too small and bring it up to the required value (half the distance between the starting position and the end point).

For center-point-based programming:

The parameters that the control unit needs are the center point, the end point and the sense of rotation of the circle. The current position is always taken to be the starting position of the circular movement.
The control unit will use the ”starting point/center point” and ”center point/end point” distances to calculate the required circle radius. If the two radiuses calculated are different (e.g., due to inexact entry of the circle center point), the PNC can compensate this discrepancy by shifting the circle center point correspondingly.
A shift is carried out in this case only if the following is true:

MP 7050 00010 < discrepancy < MP 7050 00020

If the discrepancy is smaller than MP 705000010, no compensation will be carried out. In this case, the control unit will use the programmed data only for the movement. This may cause a jump in the contour at the end of the circular segment.
Any discrepancy that is larger than MP 7050 00010 will cause a runtime error (323).

Relevant MACODA parameters (MP):

7050 00010: Radius accuracy for center-point programming.

7050 00020: Radius tolerance for center-point programming.

7050 00030: Radius tolerance for radius-based programming.

Adjusting circle radiuses

Relevant G functions:

G2/G3: Circular interpolation enabled.

G202/G203: Helical-N interpolation enabled.

3.5.1 Assigning

This function is always active with circular interpolation. It applies the current settings of the relevant MACODA parameters.

1. Adjust MP 7050 00030 as required for compensations relating to radius-based programming.
2. Adjust MP 7050 00010 and MP 7050 00020 as required for compensations relating to center-point-based programming.

CAUTION
Danger of damage to the machine or the workpiece!
Do not initiate a system control reset during machining.

3. Initiate a system control reset (DIAGNOSTICS ▶ RESET FUNCTIONS ▶ SYSTEM CONTROL RESET).

Adjusting circle radiuses

Notes:

4 Drives (axes, spindles)

4.1 Linear axis

Function:
Assigns a connected SERCOS drive as linear axis.

A linear axis is a machine axis which
- moves only along a straight path,
- is usually operated in position control mode,
- can be traversed only within a finite range, and
- is programmed with ”mm” or ”inch” units of measure.

A distinction is made between ”synchronous” and ”asynchronous” linear axes:
- Synchronous linear axes can be traversed in interpolation with other synchronous axes assigned to the same channel. Therefore, they can be used for generating tool paths in a plane or in 3D. The number of synchronous axes required for this depends on the layout of the machine tool (e.g., for lathes, 2 synchronous linear axes, for milling machines, 3 synchronous linear axes are required as a minimum).
- No interpolation is possible among asynchronous linear axes. In general, they are used to carry out simple positioning, feeder or handling tasks involving linear motions.

Relevant MACODA parameters (MP):

1001 00001: Drive function type.
1003 00001: Physical axis designation.
1003 00002: Channel assignment.
1003 00004: Axis movement type.

4.1.1 Assigning

Precondition:
- The SERCOS parameters have been correctly set in the drive or will be transferred by the control unit to the drive during SERCOS initialization (addressing, timing, operating mode, weightings, etc.; for information on the SERCOS interface, see page 4–67 ff).

1. In MP 1001 00001, assign value ”1” to the respective drive.
2. In MP 1003 00001, enter the physical axis designation.
3. In MP 1003 00004, set the axis movement type to ”1”.
4. In MP 1003 00002, specify whether the axis is a machining (i.e. synchronous) axis or an auxiliary (asynchronous) axis.
   If it is a machining axis, specify the channel to which it is to be assigned.
5. Parametrize the linear axis as required in terms of speed, dynamics, etc.

6. Ensure that the axis is correctly included in the PLC program in respect of interface signals.

CAUTION
Danger of damage to the machine or the workpiece!
Do not initiate a control RESET during machining.

---

7. Initiate a control reset.

4.2 Rotary axis

Function:
Assigns a connected SERCOS drive as rotary axis.

A rotary axis is a machine axis which
- carries out rotational movements only,
- is usually operated in position control mode,
- can be traversed only within a finite range, and
- is programmed in the unit of measure of "degrees".

Relevant MACODA parameters (MP):
1001 00001: Drive function type.
1003 00001: Physical axis designation.
1003 00002: Channel assignment.
1003 00004: Axis movement type.

4.2.1 Assigning

Precondition:
- The SERCOS parameters have been correctly set in the drive or will be transferred by the control unit to the drive during SERCOS initialization (addressing, timing, operating mode, weightings, etc.; for information on SERCOS, see page 4–67 ff.).
  For a rotary axis, the type of weighting, e.g., must be set to "rotary" in SERCOS parameter S-0-0076 (type of weighting for position data).

1. In MP 1001 00001, assign value "1" to the respective drive.
2. In MP 1003 00001, enter the physical axis designation.
3. In MP 1003 00004, set the axis movement type to "3".
4. In MP 1003 00002, specify whether the axis is a machining (i.e. synchronous) axis or an auxiliary (asynchronous) axis.
   If it is a machining axis, specify the channel to which it is to be assigned.
5. Parameterize the axis as required in terms of speed, dynamics, etc.
6. Ensure that the axis is correctly included in the PLC program in respect of interface signals.

CAUTION
Danger of damage to the machine or the workpiece!
Do not initiate a control reset during machining.

7. Initiate a control reset.

4.3 Modulo axis (endless axis)

Function:

Assigns a connected SERCOS drive as a modulo axis (rotary axis with modulo calculation, endless axis).

A modulo axis is a machine axis which

- carries out rotational movements only,
- is usually operated in position control mode,
- can be traversed endlessly in both directions of rotation,
- is programmed in the unit of measure of ”degrees”.

By means of the modulo value (also referred to as modulo factor), the current position of the axis upon reaching its set position is re-transformed into the position range between 0 and the modulo value.
The modulo value is stored in the drive (S-0-0103) and is automatically output by the PNC during start-up.

There is a positioning logic available with variable parameter setting for modulo axes. By setting the parameters correspondingly, you can define the strategy to be used by the modulo axis for approaching its target position.

Restrictions:

- While G91 is active (incremental data input), modulo calculation in the control unit is suppressed in operating modes ”Automatic” and ”Manual data input” with the exception of the output setpoint and its display.

Relevant MACODA parameters (MP):

1001 00001:	Drive function type.
1003 00001:	Physical axis designation.
1003 00002:	Channel assignment.
1003 00004:	Axis movement type.
1003 00005:	Positioning logic.
1003 00050:	Switching the positioning logic of endless axes using G151.

Relevant G functions and special functions:

G151:	Changing the positioning logic
G150:	Resetting the positioning logic to the setting in MP 1003 00005.

Modulo axis (endless axis)

G189:	Absolute data input 2
DC(..)	Changing the positioning logic to ”shortest path” for the current block.
ACP/ACN(..)	Changing the positioning logic to ”sign logic” for the current block.

Relevant interface signals:

NC-O21.0: ”G189 active” (channel interface).

4.3.1 Assigning

Precondition:

- The SERCOS parameters have been correctly set in the drive or will be transferred by the control unit to the drive during SERCOS initialization (addressing, timing, operating mode, weightings, etc.; for information on SERCOS, see page 4–67 ff.).
  For a modulo axis, the type of weighting, e.g., must be set to ”rotary” and the processing format to ”Modulo format” in SERCOS parameter S-0-0076 (type of weighting for position data). In addition, the modulo value must be parameterized in S-0-0103.

1. In MP 1001 00001, assign value ”1” to the respective drive.
2. In MP 1003 00001, enter the physical axis designation.
3. In MP 1003 00004, set the axis movement type to ”2”.
4. In MP 1003 00002, specify whether the axis is a machining (i.e. synchronous) axis or an auxiliary (asynchronous) axis.
   If it is a machining axis, specify the channel to which it is to be assigned.
5. Set the desired positioning logic in MP 1003 00005.
6. Parameterize the axis as required in terms of speed, dynamics, etc.
7. Ensure that the axis is correctly included in the PLC program in respect of interface signals.

CAUTION
Danger of damage to the machine or the workpiece!
Do not initiate a control reset during machining.

8. Initiate a control reset.

4.4 Linear modulo axis

Function:

Assigns a linear axis as an endless axis. When the linear axis has reached the modulo value, its setpoint is automatically set to ”0”. This prevents a computational overflow of axis values and allows programmable zerosetting of the program coordinate system to the current axis position with G105.
Theoretically, the axis can thus be traversed endlessly in one direction. Therefore, this function is well-suited for conveyor-type applications, e.g., uncoiling wire from a reel, spring coiling, etc.

Restrictions:

- The respective drive must be able to carry out the modulo calculation autonomously. At the drive end, the modulo value can usually be parameterized with S-0-0103.
- Position values may be programmed in a traversing block only if they are smaller than/equal to the modulo value.
- Negative position values may be programmed only if they are smaller than the modulo value.
- A control reset will reset the program zero point to the axis zero point.
- Measuring (G175/G275) in the context of linear modulo axis operations may be performed only with traversing movements in positive direction.

Relevant MACODA parameters (MP):

1003 00004: Axis movement type.

Relevant G functions:

G105: Zerosetting of modulo axis.

4.4.1 Assigning

Precondition:

- The SERCOS parameter ”Modulo value” (S-0-0103) has been correctly set in the drive or will be transferred by the control unit to the drive during SERCOS initialization (see MACODA manual).

1. In MP 1001 00001, assign value ”1” to the respective drive.
2. Assign the axis as a synchronous or an asynchronous axis to the system and parameterize it as required in terms of speed, dynamics, etc.
   The axis movement type must be set to ”Linear axis with modulo calculation”. To this end, set MP 1003 00004 to ”4” for this axis.
3. Ensure that the axis is correctly included in the PLC program in respect of interface signals.

CAUTION
Danger of damage to the machine or the workpiece!
Do not initiate a control reset during machining.

---

4. Initiate a control reset.

4.5 Hirth axis

Function:

Assigns a rotary axis or an endless axis as a Hirth axis. Hirth axes can approach only those positions that are located on a clearly defined grid (Hirth grid; ⇒ MP 1003 00055).
Therefore, this function is well-suited for the axes of tool changers, or for axes with a serration.
Data input may be set optionally to degrees or places (⇒ MP 1003 00056).

With **place programming**, enter the target position without any decimal positions.
Place 1 is always equivalent to the 0° position. All places are numbered consecutively in dependence on MP 1003 00055 and are shown on the position display in the form of position numbers. Negative position numbers, position number 0, or position numbers higher than those listed in MP 1003 00055 can be programmed only while G91 is active (incremental data input).

If the Hirth axis is to be **programmed in degrees**, automatic end-point correction is possible (⇒ MP 1003 00057). In this process, the NC corrects a programmed position automatically by shifting it to the nearest Hirth grid position. Without end-point correction, the NC will generate an error message in these cases.

Restrictions:

- No axis coupling possible
- No handwheel mode permitted
- No synchronous Hirth axis may be included in the group of axes that are selected with the ”Inclined Plane” function.
- The reference point and all contour and zero shifts of a Hirth axis must be located on the Hirth grid.
- After ”Feed hold”, ”Feed inhibit”, ”Emergency stop”, or ”Control reset” in ”Incremental jogging” mode (incremental step is active), the Hirth axis can be brought to a standstill also between permitted positions. See also Section 4.5.1 page 4–7.

Relevant MACODA parameters (MP):

1003 00055: Places on circle for Hirth axis
1003 00056: Place programming for Hirth axis
1003 00057: End-point correction for Hirth axes

Relevant interface signals:

NC-I2.5: ”Traverse to next grid position” (axis interface).
If this signal is set to high, the traversing movement is started if ”Jog” mode is active and the Hirth axis is currently not being jogged manually.
The NC sets the traversing direction automatically through the current state of the axis-related interface signal ”Traversing direction –” (NC-O14.4).

Hirth axis

NC-O16.5: ”Axis in grid position” (axis interface).
Before an axis is reference, this signal is always 0.
Once an axis is referenced, this signal is set to high only if the axis is in the ”In-position window”.

4.5.1 Assigning

Precondition:

- The axis is already available in the system as a rotary or an endless axis with its speed, dynamics and, if required, its positioning logic correctly parameterized in accordance with its operational requirements.
- The necessary axis-related interface signals are evaluated or generated within the framework of the PLC program.

1. Specify the required Hirth grid in MP 1003 00055.
2. Select the desired programming option in MP 1003 00056.
3. Activate end-point compensation in MP 1003 00057, if required.

CAUTION
Danger of damage to the machine or the workpiece!
Do not initiate a control reset during machining.

---

4. Initiate a control reset.

State: Hirth axis not on the Hirth grid...

...after ”Feed hold” or ”Feed inhibit”:

1. cancel ”Feed hold” or ”Feed inhibit”.
   ⇒ The programmed end position is approached immediately on the Hirth grid.

...after ”Emergency stop” or ”Control reset”:

1. Set the manual feed to ”Rapid”, ”Fast”, ”Medium”, or ”Slow”.
2. Activate jog mode.
3. Briefly jog the Hirth axis (via axis-related interface signals ”Manual+” / ”Manual–”; for information on ”Jogging”, see page 7–3).
   ⇒ Depending on the jogging direction (+/–), the NC traverses to the nearest position on the Hirth grid.

or

1. Activate jog mode.
2. Set the axis-related interface signal ”Traverse to next grid position”.
   ⇒ The NC traverses to the nearest position on the Hirth grid.
   The NC sets the traversing direction automatically through the current state of the axis-related interface signal ”Traversing direction –” (NC-O14.4).

Hirth axis

...after ”incremental jogging” mode (incremental step is active):

1. Set the manual feed to ”Rapid”, ”Fast”, ”Medium”, or ”Slow”.
2. Briefly jog the Hirth axis (via axis-related interface signals ”Manual+” / ”Manual–”; for information on ”Jogging”, see page 7–3).

   ⇒ Depending on the jogging direction (+/–), the NC traverses to the nearest position on the Hirth grid.

or

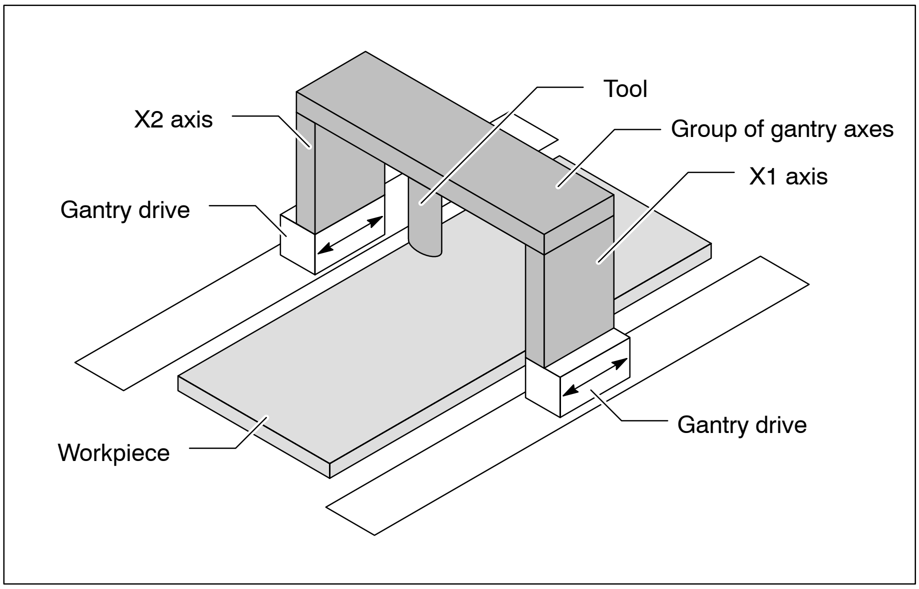
1. Set the axis-related interface signal ”Traverse to next grid position”.

   ⇒ The NC traverses to the nearest position on the Hirth grid.
   The NC sets the traversing direction automatically through the current state of the axis-related interface signal ”Traversing direction –” (NC-O14.4).

4.6 Gantry axes

Function:

Gantry axes are pairs of axes with solid mechanical coupling. They are traversed in parallel by separate drives. Coupling is active all the time.



The commissioning of gantry axes is subject to certain restrictions or, resp., may require some additional preparatory action (see Section 4.6.1).

Preconditions

- **Interface**: the drives must be digital drives as per SERCOS.
- **Drive data**: the drive data of solidly coupled axes must be identical so as to ensure that their behavior is identical also in asynchronous operation (operating modes ”Jog” and ”Traverse to reference point”). All drive data relating to axis dynamics (e.g., loop gain (KV), feed forward etc.) must be kept identical for all gantry axes of a group. This must be taken into account also for programming. This data may be changed only while the axes are at rest, because not all the drives would be able to recognize and execute a switching operation simultaneously (in the same SERCOS clock pulse).
- **Measuring system**: for gantry axes, identical absolute or cyclic/absolute measuring systems are required.

☞ **This description refers to cyclic/absolute measuring systems.**

Absolute encoders do not require any special drive referencing action to be taken during commissioning because the exact drive positions are known upon start-up.

- **Reference point switch**: for all gantry axes, there is a common reference point switch (cam) which is tripped by the master axis and which relays the signal ”Reference point switch recognized” simultaneously to all gantry drives.
- **Constellation of axis types**: in any one group of gantry axes, the following constellations of axis types are permitted:

Master axis	**Slave axis**
linear	linear
linear modulo	linear modulo
endless	endless
rotary	rotary

For modulo axes, the modulo values of master and slave axes must be identical.
The master axis must be a synchronous axis, the slave axis may be an asynchronous axis.

Restrictions

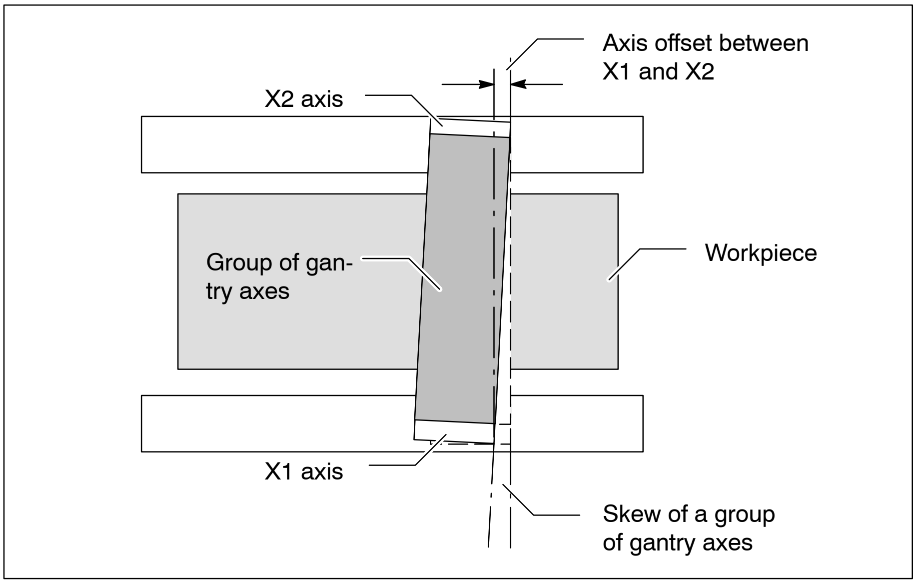
- After starting up the control unit, the coupled axes must be properly **aligned** by the mechanical coupling so as to permit parallel traversing of the axes (distance between the axes = constant).
- The signal flow as described in Sect. 4.6.1, “Assigning gantry axes” (para. 7., see page 4–19 ff.) must be strictly complied with in the referencing process.
- After every change of motor and/or measuring systems, a new commissioning must be carried out.
- After every change of software in the drives, a new commissioning must be carried out, because the working memory stored in the drive is lost.
- In referencing parameter S-0-0147, bit 7 (position after drive-controlled referencing) may be set to “1“ only if the correct reference offsets have been accepted by the drive.
- The master axis of a gantry axis group must never be a slave of an axis coupling at the same time.
- The manufacturer-specific SERCOS parameters (P-x-xxxx) mentioned in the context of the commissioning are relevant to Servodyn drives only. Equivalent parameters must be used in the case of other drives. The functionality must be ensured by the drive manufacturer.

Correcting a skew of a group of gantry axes

Before a group of gantry axes can be traversed for machining, any displacement of master and slave axes must be corrected. A distinction is made between

- permissible skews (correction by referencing the drives) and
- unpermissible skews (can be corrected only by mechanical shifting or axis jogging).

Gantry axes



☞ **A skew of the gantry axes must be corrected before the control unit is started if the skew prohibits the traversing of the gantry axes.**

A skew of gantry axes may have various causes:

- **The axes have not been referenced yet:** if the commissioning has been carried out correctly and the reference point is properly approached (S-0-0147, bit 7 = 1), the group of gantry axes is aligned in the referencing process.
  - ★ Carry out the “Approach reference point” function for the respective group of gantry axes.

- **Commissioning parameters missing: after** control unit has been started, but **before** referencing, the values provided by the encoders and the actual axis positions are not interrelated. In this condition, the control unit cannot recognize a skew, which persists until referencing is carried out. Until then, the commissioning distance (difference between the two encoder positions within one revolution) determined during start-up has to be maintained when the group of axes traverse. No compensation of the skew is possible at this stage.

★ Carry out the commissioning in accordance with Section 4.6.1.

- **Incorrect reference point offset:** if cyclic/absolute motor encoders are used, first the reference points and the reference point offsets of the gantry axes must be identified in order to be able to identify and remove the skew. Incorrect entries of the offsets in the SERCOS file may cause an unintentional skew of the axis group.
  - ★ Carry out the commissioning in accordance with Sect. 4.6.1 and determine the correct reference point offsets.

- **Axis jamming (static skew):** the actual position values of an axis are falling short of its selected setpoints. The skew can only be corrected when the jam has been removed!
  When the axis jam has been removed mechanically, the drives can be restarted.

  After the setpoint values have been released, jumps are applied to the drives.
  A persisting skew will thus be corrected with a jerk. If this jerk leads to a servo error, the loop gain factor of the respective axes must be lowered until the skew can be corrected without causing a servo error. This may be done, e.g., by means of a CPL program where the loop gain factor of the gantry axes can be changed.
  Once the axes have been released, the loop gain factor must be reset to its original value.

  ☞ **In the event of an axis jam, the drive must be shut down and the axis must be set free mechanically.**

Monitoring functions for gantry axis groups

There are various monitoring functions available to monitor and control the behavior of gantry axes:

- **Monitoring of following distance:** MACODA parameter 1003 00060 (tolerance = max. following error for coupled axes) allows to monitor the following distance during commissioning and machining: For this purpose, the control unit uses the overtravel information of master and slave from the cyclical SERCOS parameter S-0-0189. If S-0-0189 is not contained in the cyclical SERCOS telegram, the following distance is calculated from the position actual values of the gantry axes.

  When the maximum permissible following distance between master and slave is exceeded the group of axes is decelerated to a stop and a coupling error is generated.

  - **Reaction during commissioning**: As long as no reference points have been approached, the commissioning distance is taken as a reference for the actual position deviation. The actual position deviation must not differ from the commissioning distance by more than the tolerance range. Otherwise this will cause an error. If an error occurs, the axis group will be decelerated to a standstill.
  - **Reaction during operation**: If the deviation of the actual position values of slave and master axes exceeds a specified limit (tolerance), the axis group will be decelerated to a standstill (if a traversing movement is active) and a coupling error will occur.

Error analysis:

In the event of an unpermissible skew, the interface signal NC-O 16.6 “Coupling lag exceeded” will be set for the axis/axes involved. When the following distance is no longer exceeded, the signal will be deactivated.
This error state in the NC can be suppressed by setting the axis signal NC-I 4.5 for the master axis. In this event, monitoring of the coupling lag has to be programmed in the PLC.

☞ **Even if an error occurs, the axis coupling is not disbanded.**

- **Torque/current monitoring:**
  Torque/current monitoring (standstill (zero-speed) monitoring) works only in the **stationary state**, i.e. all gantry axes are ”in position” and there is no traversing command from the NC pending.
  Active axis torques are obtained from the SERCOS telegram.
  Torque monitoring outputs a reaction if the **sum total** of the torques of all drives of a gantry axis group exceeds the **max. total static torque** specified in MP 1003 00062.

  The static torque total is a qualitative measure of the torque bias. The quantitative characteristics must be specified by the manufacturer. They must be determined by various series of tests.

☞ **Evaluating the torque characteristic of the drives in motion would not be meaningful because of the heavy variations in driving torques that are likely to occur while the drives are in motion.**

SERCOS setting

To enable torque monitoring, the actual torque values (S-0-0084) of the gantry drives must be transferred with the cyclical SERCOS telegrams (entry in SERCOS file ”p2<xxx>.scs”). See also Sect. 4.6.1.

Error analysis

If the torque/current monitoring function indicates an excessive skew, first the **cause of the error** must be removed before the skew is corrected by referencing! Any referencing attempt while the cause of the error continues to exist will cause a repeated reaction of the monitoring function.

If the torque total is **exceeded**:

- axis signal NC-O 16.7 is set for the master axis. This signal will be reset as soon as the torque total no longer exceeds the limit.
- the NC outputs an error message, which is canceled upon a control reset.
  This error message can be suppressed by setting axis signal (NC-I 4.6).

  Output signal NC-O 16.7 is independent of NC-I 4.6!

Gantry axes

Relevant MACODA parameters (MP):

1003 00060: Maximum following distance for coupled axes
Defines for every **axis**
- whether, with axis coupling activated, following error monitoring is to be activated or deactivated
- the max. permissible following error of a slave axis.

1003 00061: Index of the leading gantry axis
Defines for every **axis**
- whether it belongs to a gantry group
- whether it is a leading axis.

1003 00062: Max. total static torque of a gantry axis group
Defines for each **leading axis** of a gantry group
- whether static torque monitoring of the respective axis group is to be activated or not
- the monitoring limit value.

Relevant G functions:

All G functions are permitted that allow the meaningful programming of a gantry group.

Relevant SERCOS parameters:

S-x-0041: Referencing speed

S-x-0042: Referencing acceleration

S-0-0147: Referencing parameter

S-x-0150/151: "Reference dimension offset" with motor encoder. If the reference dimension offset (SERCOS parameter Parameter S-x-0150 with motor encoder, S-x-0151 with external encoder) of the master axis is changed, the reference dimension offset of all slaves must be adjusted to ensure that the difference of the reference offsets between the slaves and the master is still the same.

S-0-0177/178: "Absolute dimension offset" with motor encoder. Distance between the machine datum and the zero point of the absolute measuring motor encoder.

S-0-0189: Following distance for the following distance monitoring, if included in the cyclical telegram.

S-0-0084: Torque actual value for torque monitoring.

S-0-0016: Recording of the torque actual value S-0-0084 and the following distance S-0-0189 in the cyclical telegram in the SERCOS files (p2*.scs) for the respective gantry axes.

☞ **With the exception of the reference dimension offset, S-x-0150/S-x-0151/S-0-0177/S-0-0178 (depending on the encoder type), all values must be identical.**

Gantry axes

Relevant Servodyn- SERCOS parameters:

P-0-0504: “State of reference point switch”. Marker in the critical range (yes/no)

☞ **Servodyn-SERCOS parameters (P-x-xxxx) are used for gantry axes commissioning. If other drives are used, the availability of equivalent functions must be checked and the parameter numbers assigned to these functions must be determined!**

Relevant interface signals:

NC-I 2.0-2.3 “Mode selection” (here: approach reference point)

NC-I 3.0 “Mode selection by PLC” (here: approach reference point)

Relevant axis interface signals:

NC-O 16.6 ”Coupling lag exceeded”:
This output signal is assigned to each axis of the gantry group. The signal is constantly updated, i.e. it does not remain active statically when an error that occurred has been removed.

Slave lag monitoring:
Low: slave follows within the permitted coupling lag
High: slave is exceeding the permitted lag.

NC-O 16.7 ”Standstill torque exceeded”:
This output signal is assigned to each axis of the gantry group. The signal is constantly updated, i.e. it does not remain active statically when an error that occurred has been removed.

In a gantry group, the NC-O 16.7 signal is relevant only to the master axis. The signals for the slave axes are not evaluated.
Low: the torque total of all axes is within the permitted range.
High: the torque total of all axes has exceeded the permitted range.
(NC-O 16.7 is independent of NC-I 4.6!)

NC-I 4.5 ”Coupling lag: NC error OFF”:
This input signal is assigned to each axis of the gantry group although it is relevant to the master axis only. The signals for the slave axes are not evaluated.
Low: if an error occurs, error handling/message by the NC.
High: if an error occurs, no error handling/error message by the NC.

NC-I 4.6 ”Standstill torque: NC error OFF”.
This input signal is assigned to each axis of the gantry group although it is relevant to the master axis only.
If this signal is set for the master axis, error analysis within the NC is suppressed if the static torque limit is exceeded.
Low: if an error occurs, error handling/message by the NC.
High: if an error occurs, no error handling/ error message by the NC.

NC-I 7.5 ”Drive inhibit, general”

NC-I 7.6 ”Drive off, general”

NC-I 7.7 ”Feed inhibit, general”.
Before referencing can be started, the PLC must inhibit **all** the gantry axes of the group.

NC-I 0.0-0.1 ”Manual +/–”.
In the ”Approach reference point” mode, the PLC sets the jog signals for all gantry axes.

☞ **Notes on interface signals:**

- The axis interface must be processed by the PLC both in respect of the master axis and in respect of the **gantry-**slave axes (the interface of a gantry-slave axis shows the same performance as the interface of a slave axis coupled by programming with a G function).
- The following drive-related interface signals (NC inputs) must always be kept identical within a group of coupled axes:
  – drive inhibit
  – drive off
  – feed inhibit

  This must be ensured by the PLC.

4.6.1 Assigning

Preconditions

The following conditions must be satisfied in order to assign gantry axes:

- After start-up, the master and the slave axes traverse simultaneously if all the axes have been assigned to the gantry group in MP 1003 00061. This applies to jog mode, approach the reference point, ..
- All gantry axes have **one** common reference point switch, which they recognize **simultaneously** during referencing. Therefore, the drives complete their traversing motions all at the same time.
- The reference offset can be set **independently** for the master axis only.
- Setting the reference offset of a slave independently is possible only during commissioning.

Gantry axes

- After commissioning, the reference dimension offset of the master (SERCOS parameter S-x-0150 with motor encoder, S-x-0151 with external encoder) can be changed.
  Simultaneously, the reference offset of all the slaves must be adjusted so as to ensure that the difference between the reference offsets (= commissioning distance) of the slaves and that of the master is identical with the commissioning distance identified.
- Servodyn-SERCOS parameters (P-x-xxxx) are used for gantry axes commissioning. If other drives are used, the availability of equivalent functions must be checked and which parameter number is used to activate it!

CAUTION
Dangerous skew of the gantry axis group!

As long as the reference offsets of the axes are not correctly entered and accepted, and SERCOS parameter P-0-0504 (cam position status (Reference point status)) is not saved in the drive, the reference point must not be approached. Otherwise, an offset occurs that may cause a dangerous skew of the gantry axis group! Prior to the first commissioning of the gantry drives, referencing parameter S-0-0147 must be set to bit 7 = 0 to allow the drive to be in a random position after referencing. In addition, parameter P-0-0504, cam position status (reference point switch) must be set to “0” (=init).

Sequence

The following description (items 1. through 9.) illustrates the sequence of a gantry axis group commissioning:

1. Correcting an **unpermissible skew** of a gantry axis group

   The process of assigning a gantry axis group may be different with various systems.
   Because the drives involved have cyclic absolute measuring systems, the focus in assigning is on the determination of a common reference point for all the gantry axes and on aligning them among each other.
   Before the referencing operation may be carried out, the key issue may be to get the axes out of an unpermissible skew.

   There are two ways to align the axes (depending on the machine tool design):

   - In the case of **gantry groups on heavy-duty portal machines**, individual gantry axes must be aligned in parallel with each other either mechanically or by jogging. However, this requires that the gantry coupling of the axes must be disbanded in MP 1003 00061 (all parameter values = 0) to allow jogging the master and the slave axes **individually** after start-up.
     In this case, carry out the following sequence:
     - Start up the control unit and the drives.

Gantry axes

  - Set MP 1003 00061 to the effect that no axis is assigned to a gantry group.
  - Start up the control unit and the drives again.
  - Jog the individual axes into a position where they are in parallel with each other and where a remaining skew can be corrected by subsequent referencing.
  - Re-assign the axes to the gantry group in MP 1003 00061. Continue with step 2. of this sequence.
- In the case of **relatively rigid gantry groups**, it is usually sufficient to shut down the drives and the gantry group will align **itself**. In this case, carry out the following sequence:
  - Shut down the machine.
  - Wait till the torque bias forces have pulled the gantry group into a parallel starting position, thus reducing the skew. Continue with step 2.

2. Start up the control unit and the drives.
3. The following **SERCOS parameters** must be set:
   - in parameter S-0-0147 (referencing parameter), set bit 7 = 0, to allow the drive to be in a random position after referencing.
   - set parameter P-0-0504 (cam position status) to “0” (= init).
4. The following **MACODA parameters** and, if applicable, **SERCOS parameters** must be set:
   - MP 1001 00061:
     Index of the leading gantry axis (not zero-based!)
     Defines for every **axis** whether it belongs to a gantry group or not. Enter the axis index (= axis number) of the leading axis for all axes of a gantry group, including the leading axis.
   - Setting the MACODA parameter for **torque monitoring** of the gantry axes (if desired):
     - MP 1003 00062:
       determine the max. total static torque (unsigned amounts!) of all the gantry axes (only the amount entered for the master will be taken into account). This parameter defines for every axis, provided it is the leading axis of a gantry group, whether static torque monitoring of the gantry group is to be activated or not. If this parameter is set to “0”, torque monitoring of the group is deactivated. Any value> “0” means: the maximum torque total of all group axis is monitored while the axes are at rest.
   - If torque monitoring is provided in MACODA, the actual torque value must be included in the cyclic telegram in the SERCOS files of the gantry axes.
     Example: Y axis
     In SERCOS file p2lin00y.scs:
     S-0-0016 = (...,S-0-0084 ,...)

Gantry axes

- Setting the MACODA parameter for **position actual value monitoring** of the gantry axes (if desired):
  - MP 1003 00060:
    maximum following error for coupled axis.
    Each slave is assigned a maximum following error (=tolerance) relative to the master. If this value = “0”, the maximum permissible following error is “0” for reasons of safety!

5. For the **parameter settings to be adopted**, restart the control unit and the drives.
6. Identifying the **commissioning distance**:
   After starting up the control unit, the drive encoders first provide the **actual position values** determined by one encoder revolution.
   Normally, the actual position values of the gantry axes differ, because the way their encoders are mounted is not exactly identical.
   The difference in the actual position values between the slave axis/axes and the master axis is defined as the commissioning distance.

☞ **As long as the axes have not been referenced, the commissioning distance between the master axis and the slave axis is maintained at its identified value by an appropriate setpoint input.**

7. Gantry axis **referencing** during commissioning:
   Precondition:
   - The referencing parameters of the gantry axes (S-x-0041: Speed and S-x-0042 acceleration) must be set so as to ensure that no damage to the mechanical system can occur in the course of the referencing process even if mistakes have been made during commissioning (e.g., incorrect offset data). Monitoring functions (torque/current monitoring), e.g., may be used to cancel movements before any damage occurs.
   - Referencing parameter: S-0-0147 bit 7 = 0.
     This setting must be maintained until the correct reference offset is entered in the SERCOS files and accepted by the drives after a SERCOS or control unit start-up.
   - Before referencing can be started, the PLC must inhibit all the gantry axes of the group by setting the ”Feed inhibit” interface signal.
     Next, the PLC sets the mode ”Approach to reference point” for all gantry axes at the interface as well as the jog signals for these axes.
     If the travel commands for all gantry axes are active at the interface, the PLC sets the ”Feed inhibit” signal to low. Now the drives approach to the reference points.

☞ **Observance of the above signal sequence is mandatory!**

Gantry axes

The **reference dimension offset** is determined both for the master and the slave(s). Because encoder mounting is never exactly identical (commissioning distance), this commissioning distance must also be taken into account for the reference offsets of the drives.

Carrying out “Drive-controlled referencing”:

- The command “Drive-controlled referencing” is set and released when the necessary control and status signals are assigned real-time bits by the master through the service channel.
- The drive switches to internal position control, cancels the position actual values status (S-0-0403), and ignores the cyclic set-point input.
- Taking the approach direction specified in the referencing parameter S-0-0147 into account, the drive accelerates with the referencing acceleration stated in S-0-0042 to the referencing speed specified in S-0-0041.
  Upon the signal change at the reference point switch, which is specified in S-0-0147, the drive recognizes the absolute encoder position within one revolution.

  Please note for gantry axes as well as for other axes:
  The positional distance between the “reference point switch and encoder marker” should be in the range of > $^{1}/_{4}$ encoder revolution or < $^{3}/_{4}$ encoder revolution.
  If the positional distance is in the critical range (< $^{1}/_{4}$ encoder revolution or > $^{3}/_{4}$ encoder revolution), this is signaled to the drive by the ”Save working memory” command (S-0-0264). This is to prevent a grid jump if further referencing processes are carried out.

☞ **See also the Servodyn-D with SERCOS interface manual.**

- The drive decelerates with the value from S-0-0042 to standstill. Simultaneously, the drive calculates the actual position on the basis of the encoder position, the reference dimension position actual value 1, and the reference dimension offset 1, and sets the position actual values status (S-0-0403).
  If required, the axis positions relative to the machine zero point must be set out.

Example: referencing a master and a slave drive

During a referencing process, drives recognize the reference switches
- Encoder value (reference switch) of the master axis: 8.000
- Encoder value (reference switch) of the slave axis: 3.000

The commissioning distance is 8.000 - 3.000 = 5.000

After recognizing the reference switches, the drives decelerate the referencing movement down to a standstill.
- Encoder value (standstill) of the master axis: 10.000
- Encoder value (standstill) of the slave axis: 5.000

Setting out the axes provides the actual position 1 of the axes relative to the machine zero point: 20.000

Gantry axes

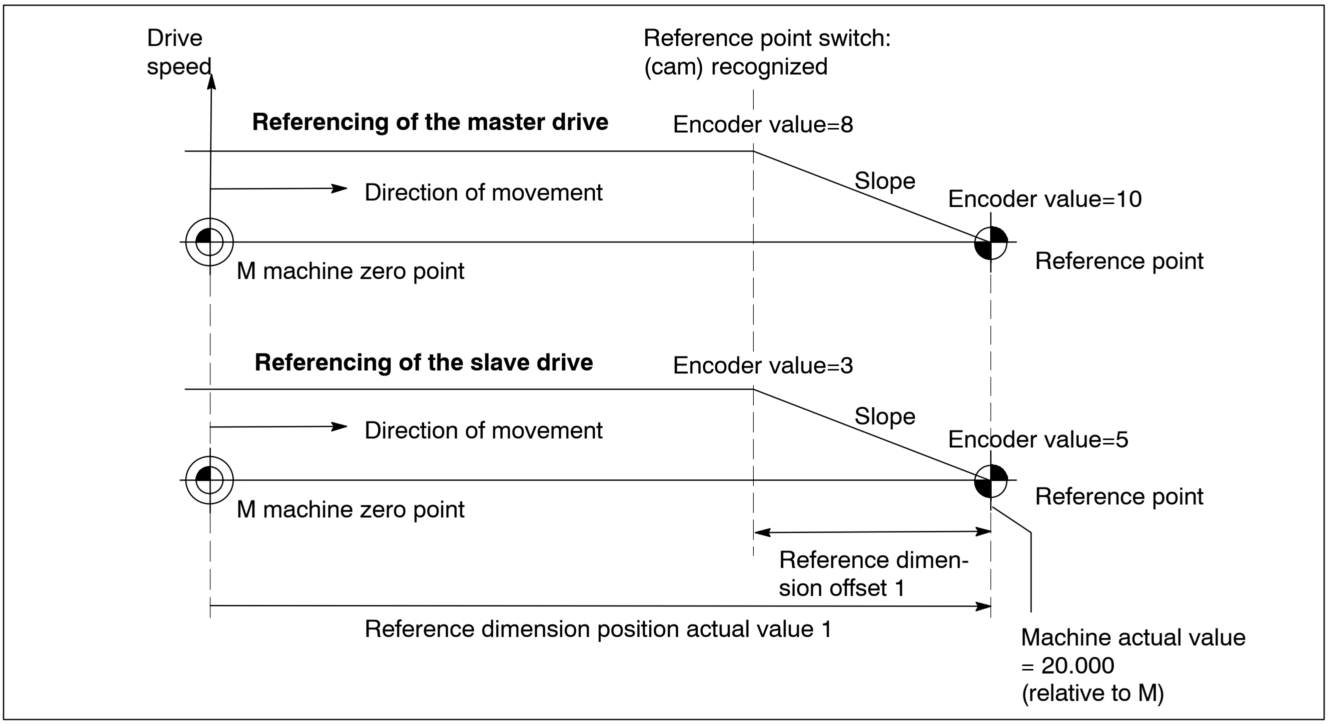
Because the gantry axes are aligned, they both have the same position relative to the machine zero point!

This leads to the following results:
- reference dimension offset 1 of the master axis
20.000–10.000=: **10.000**
- reference dimension offset 1 of the slave axis
20.000– 5.000=: **15.000**

The reference dimension offset 1 (S-X-0150) of both drives must be entered in the relevant SERCOS files.

The resulting positions of the reference point switches relative to the machine zero point are:
Master: encoder value + offset: 8 + 10 = 18
Slave: encoder value + offset: 3 + 15 = 18

After acceptance of the offset, the NC provides the value 18 as the reference point switch positions of both axes.



8. **Setting the SERCOS parameters**
   - Enter the determined **reference dimension offsets** of the axes in SERCOS file **p3*.scs**.
   - When referencing during commissioning is finished, the SERCOS **referencing parameter** for approaching the reference point must be set again (S-0-0147, bit 7 = 1) to ensure that the gantry group is aligned after referencing.
   - Parameter P-0-0504, **Cam position status** (reference point switch) is saved either with the command ”Save working memory” (S-0-0264), or it must be saved by the commissioning engineer.

☞ **Please note for the SERCOS file p3*.scs of every gantry axis: With the exception of the reference dimension offset, S-x-0150, S-x-0151, S-0-0177, S-0-0178 (depending on the encoder type), all values must be identical.**

9. Adopting the parameters by restart.

CAUTION
As long as the reference offsets of the axes are not correctly entered and accepted, and SERCOS parameter P-0-0504 (cam position status (Reference point status)) is not saved in the drive, the reference point must not be approached (S-0-0147 bit 7 =0)!

CAUTION
Damage to the mechanical system of the gantry group!

The referencing parameters of the gantry axes (S-X-0041: speed and S-X-0042 acceleration) must be set so as to ensure that no damage to the mechanical system can occur in the course of the referencing process even if mistakes have been made during commissioning (e.g., incorrect offset data).

Provide proper monitoring (e.g., torque/current monitoring) to be able to cancel movements before any damage occurs!

4.6.2 Activating

Precondition:

- All the tasks required to assign the gantry drives have been performed.

1. Start the system. After system start-up, the gantry coupling is **active**.
2. **Reference** the gantry group
   - if a “skew” (due to missing reference points, etc.) exists,
   - if the system must be re-commissioned after drives, encoders, or drive software have been changed, and
   - after every start-up.
3. To perform traversing movements, program only the master axis!
4. Jogging coupled axes: The master axis only can be jogged. All other axes coupled to this master traverse synchronously.

4.6.3 Deactivating

A gantry group cannot be deactivated by programming.

The only way to deactivate the gantry axes is through MP 1001 00061 (index of the leading gantry axis).
Every axis that belongs to a gantry group is assigned the parameter “0”. This will disband the group. When the system is restarted, this gantry group does not exist any more.

4.7 Approaching the reference point

Basics:

The PNC has to know the absolute position of all connected drives at any time. Otherwise, the proper and in particular safe control of traversing movements with regard to the maximum traversing range of the machine and the use of safety-relevant functions (position/area control, software limit switches, etc.) is impossible.

For this purpose, the control unit requires information from the drive whether its actual position values are related to the reference point of the axis:

- If that is the case, either an absolute encoder is being used or the reference point has already been approached, if an incremental encoder is used. ”Approaching the reference point” is not necessary in this event.
- If that is not the case, ”Approaching the reference point” has not yet been initiated if an incremental encoder is used. ”Approaching the reference point” is necessary in this event.

The drive keeps this information available and up-to-date in S-0-0403 (bit 0).

The logic for approaching the reference point (searching the marker, etc.), which used to be integrated in the NC, as well as the measuring system input no longer constitute part of PNC but are part of the SERCOS drive today.
That is why not the PNC but the drive has to be configured to the used measuring system via SERCOS parameter.

☞ **If the automatic parameter download for the drives is activated (see page 4–70 ff.), the SERCOS parameters in the SERCOS files of PNC have to be adapted!**

Present day SERCOS drives are capable of performing ”Drive-controlled referencing”, when given the command.
Having received the command ”Drive-controlled referencing” (S-0-0148), the drive is uncoupled from the position setpoint values of the NC and generates its own position values for referencing.
For this purpose, it uses the parameters S-0-0147 (referencing parameter), S-0-0041 (referencing speed) and S-0-0042 (referencing acceleration).

☞ **For the axis potentiometer to have effect on these drive-controlled movements too, it has to be transmitted to the drive with SERCOS parameter S-0-0108 (Feedrate override):**

- **from the PLC via the service channel in 500 ms increments or**
- **in the cyclical telegram (MDT). For this purpose, S-0-0108 has to be entered in the configuration list MDT (S-0-0024).**

Approaching the reference point

At the same time, it informs the NC that its actual position values are no longer related to the reference point of the axis (S-0-0403: bit 0=0). Once the drive has successfully carried out ”Drive-controlled referencing”, bit 0 in S-0-0403 changes back to logical ”1”.
The PNC acknowledges this information by setting the interface signal NC-O15.7 ”Reference point is known” at the respective axis interface.

This is shown in the axis display by a sign added to the position value. The following signs may be used:
* referenced synchronous axis
$ non-referenced synchronous axis
# referenced auxiliary axis
@ non-referenced auxiliary axis

Function:

The PNC can initiate a required ”Approaching the reference point” in different ways:

Case 1: manually via the standard operator interface

Case 2: via PLC through the digital interface

Case 3: via NC syntax through the G function G374

Case 4: via NC syntax through the G function G74

In cases 1 through 3

- the interface signal NC-O15.6 ”Reference point was reached” is reset at the respective axis interface.
- the command ”Drive-controlled referencing” (S-0-0148) is transmitted to the relevant drives. How the drives react to this command depends on the drive configuration.
- a positive acknowledgment of the drives involved (bit 0=1 in S-0-0403) is required.
  Once it arrives, the NC sets the interface signals NC-O15.6 ”Reference point was reached” and NC-O15.7 ”Reference point is known” at the respective axis interface.

☞ **If a drive does not report a positive acknowledgment after an adequate period of time, ”Approaching the reference point” has to be cancelled using control reset.**
In this event, the interface signals NC-O15.6 and NC-O15.7 stay reset at the respective axis interface.

Approaching the reference point

Case 4

As opposed to cases 1 to 3, case 4 (G74) is no *real* ”Approaching the reference point” but exclusively a linear traversing movement in rapid feed to the reference point coordinates of programmed axes.

The positions to be approached are determined by the PNC during start-up directly from the connected drives.
Depending on the measuring system used, the following SERCOS parameters are output for this purpose:

Motor encoder, incremental:	S-0-0052 (reference dimension position actual value 1)
External motor encoder, incremental:	S-0-0054 (reference dimensionposition actual value 2)
Motor encoder, absolute:	same as for motor encoder, incremental
External encoder, absolute:	same as for external encoder, incremental

☞ **If the automatic parameter download for the drives is activated (see page 4–70 ff.), the SERCOS parameters in the SERCOS files of PNC have to be adapted!**

Relevant G functions:

G 74: Approach reference point coordinates.
It is possible to program in the G74 block which axes are supposed to traverse.
Example: N.. G74 X0 Y0 Z0 B0

Although numerical values following the axis addresses have to be programmed, they are of no significance for the actual traversing movement.
Auxiliary and additional functions may be programmed within the same block.
Please note that the channel-related interface signal NC-O20.0 ”Rapid traverse active” is not output during G74!

CAUTION
Danger of damage to the machine or the workpiece!
Possibly active compensations will remain unconsidered in the course of the positioning process via G74.
In addition, G74 is not executed for referenced axes.

Approaching the reference point

G 374: Approach reference point.
It is possible to program in the G374 block for which axes ”Approaching the reference point” is to be initiated.
If several axes have been programmed in a G374 block, they will approach their reference point independent of each other (no continuous-path operation).
Example: N.. G374 X0 Y0 Z0 B0

Please note that although numerical values following the axis addresses have to be programmed, they are of no further significance.
Auxiliary and additional functions may be programmed within the same block.

Relevant interface signals:

NC-O15.6 ”Reference point was reached” (axis interface).
Only has effect in case of ”Approaching the reference point” via operator interface, PLC and G374.
The signal has not been set if
– ”Approaching the reference point” has not taken place
yet
– a referencing operation is still active, or
– the last referencing operation has been cancelled.
It is set again only when ”Approaching the reference point” has been executed successfully.

NC-O15.7 ”Reference point is known” (axis interface).
Has effect in case of ”Approaching the reference point” via operator interface, PLC and G374.
The signal has not been set if the drive does not know the absolute axis position.
The signal has been set if the drive knows the absolute axis position.

Relevant MACODA parameters (MP):

1005 00002: Rapid traverse rates of axes.
Is only relevant in case of ”Approaching the reference point” via G74.

4.7.1 Assigning

☞ **If the automatic parameter download for the drives is activated (see page 4–70 ff.), the SERCOS parameters in the SERCOS files of PNC have to be adapted!**
This applies to all SERCOS files transmitted to the drives in phases 2 and 3.
If the automatic parameter download is deactivated, SERCOS parameters have to be adapted in the drive itself. You may use the SERCOS monitor (see page 4–73) for this purpose.

Information on the parameters listed below is provided in the parameter manual of the drive documentation.

in connection with G74

The following parameters have to be adapted, if necessary, depending on the measuring system used because they are used to establish the setpoint positions of the axes programmed in the G74 block:

- when using an incremental motor encoder:
  S-0-0052 (reference dimension position actual value 1)
- when using an incremental external encoder:
  S-0-0054 (reference dimension position actual value 2)
- when using an absolute motor encoder:
  S-0-0052 (reference dimension position actual value 1)
- when using an absolute external encoder:
  S-0-0054 (reference dimension position actual value 2)

All parameters listed are included in the SERCOS files for phase 3.

in connection with ”Approaching the reference point” via operator interface, PLC or G374

- S-0-0026 (configuration list signal status word).
  Has to include SERCOS parameter S-0-0403.
  Included in the SERCOS files for phase 2.
- S-0-0041 (referencing speed in m/min).
  Included in the SERCOS files for phase 3.
- S-0-0042 (referencing acceleration in m/s^2).
  Included in the SERCOS files for phase 3.
- S-0-0147 (referencing parameter).
  Required coding for Servodyn drives: 0b10000100
  Included in the SERCOS files for phase 3.
- S-0-0407 (reference enable).
  Required coding for Servodyn drives: 0b00000001
  Included in the SERCOS files for phase 3.

Approaching the reference point

4.7.2 Activating

Preconditions:

- The required SERCOS parameters have to be assigned correctly in accordance with Chapter 4.7.1.
- SERCOS communication has been established with the respective drive and is working properly.
  In this context, refer to the description of the following signals:
  NC-I8.x / 9.x: ”Drive inhibit” (axis interface)
  NC-I10.x / 11.x: ”Drive off” (axis interface)
  NC-O14.0: ”SERCOS system ready” (axis interface)
  NC-O14.1: ”Drive ready” (axis interface)
  NC-O14.2: ”Drive under control” (axis interface)
  NC-O15.0: ”Axis inhibited (test)” (axis interface)
- Axes for which ”Approaching the reference point” is to be initiated have to be at standstill.
- The axis interface signals ”Handwheel assignment bit x” (NC-I2.0 and NC-I2.1) must not be set.
- The axis interface signal ”Feed inhibit” (NC-I12.x / NC-I13.x) must not be set.

manually via operator interface

Preconditions:

- The signal ”Mode selection by PLC” (NC-I3.0) has not been set at the channel interface.

☞ **”Approaching the reference point” of auxiliary axes is not possible via the operator interface.**

1. At the operating panel ”Approaching the reference point”, select:
   MANUAL ▶ REFERENCE POINT
2. Initiate ”Approaching the reference point” of the desired synchronous axes by using the respective jog key (axis interface signal ”Manual+” or ”Manual–”).

Use the axis display to check whether ”Approaching the reference point” has been carried out successfully. the additional sign following the position value of the axes involved has to change from ”$” to ”*”.

via PLC

1. The signal ”Mode selection by PLC” (NC-I3.0) is set at the respective channel interface.
2. Operating mode ”Manual approaching the reference point” is activated.
   In this context, refer to the description of the following signals:
   NC-I2.0: ”Mode selection bit 0” (channel interface:
   NC-I2.1: ”Mode selection bit 1” (channel interface)

NC-I2.2: ”Mode selection bit 2” (channel interface)
NC-I2.3: ”Mode selection bit 3” (channel interface)
Required coding: 0b0010.
For auxiliary axes:
NC-I1.6: ”Mode selection bit 0” (axis interface)
NC-I1.7: ”Mode selection bit 1” (axis interface)
Required coding: 0b10.

3. The respective axis interface signals ”Manual+” or ”Manual–” are set.

By means of the axis-related interface signals NC-O15.6 ”Reference point was reached” and NC-O15.7 ”Reference point is known”, the PLC can check whether ”Approaching the reference point” has been carried out successfully: both signals must be set.
If a drive does not give a positive acknowledgment after an adequate period of time, the PLC may cancel ”Approaching the reference point” by control reset:
jointly for all synchronous axes:
– positive edge at the channel-related interface signal input NC-I1.1
  for auxiliary axes:
– positive edge at the axis-related interface signal inputs NC-I3.1

via G374

1. Program G374. For syntax, see ”Relevant G functions”.
   Both synchronous and asynchronous axes may be programmed.

2. Initiate cycle start for execution.

Use the axis display to check whether ”Approaching the reference point” has been carried out successfully. The additional sign following the position value of the axis involved has to
– change from ”$” to ”*” (in case of synchronous axes) or
– from ”@” to ”#” (in case of asynchronous axes).

via G74

1. Program G74. For syntax, see ”Relevant G functions”.
   Only synchronous axes can be programmed in the G74 block!

2. Initiate cycle start for execution.

The programmed axes traverse to their reference point.

The additional sign following the position value of the axes involved does not change in the process.

4.8 Drive-controlled interpolation

Function:

With this function, data can be provided to a machining axis under its physical axis address through any channel, although it continues to be permanently assigned to its master channel (⇒ MP 100300002).
Position, speed and acceleration values may be specified.

For this purpose, the ”Drive-controlled interpolation” is first activated for the respective axis using G522. Through G522, the control unit switches over the current operating mode of the drive via S-0-0134 ”Master control word” (bit 8 and 9, coding the same as P-0-0127).
Please note:

if prior to switching over the drive is in

- main operating mode, it is switched to secondary mode 2.
- secondary operating mode 1, it is switched to secondary mode 3.

The drive function ”Interpolation in the drive” has to be parametrized for secondary operating mode 2 as well as for secondary operating mode 3. Parametrization is performed in the drive parameters S-0-0034 (for secondary operating mode 2) and S-0-0035 (for secondary operating mode 3) via bit 4 to bit 7.

Restrictions:

- The ”Drive-controlled interpolation” of an axis can only be switched on and off on its master channel (⇒ MP 1003 00002).
- At a given moment of time, an axis can only receive position data from one channel. Otherwise, an error message will be generated.
- Position specifications from the master channel (G520) are not permitted.
- If a channel initiates new position data although the last data specified by the same channel has not been completely interpolated yet, the old position data will be replaced by the new one.
- During a drive-controlled interpolation, the override function of the control unit (see page 7–23) has no influence on the corresponding axis.
  If influencing the axis via override is required for this period of time all the same, the PLC has to write the desired override value directly into drive parameter S-0-0108 (Feedrate override).
- If ”Feed hold” is initiated on the master channel during the drive-controlled interpolation, the traversing movement of the axis is halted but it can be continued again via ”Cycle start”.
- If ”Control reset” is initiated on the master channel during the drive-controlled interpolation, the drive-controlled interpolation is cancelled. The control unit takes over the interpolation again (corresponds to G521).
  An entry of G521 into the init string (see page 3–6) is not required.
- During the drive-controlled interpolation, a selective reset for this axis remains without effect.

Relevant MACODA parameters (MP):

1003 00002: Channel assignment.

7040 00010: unit of feedrate programmed in metric.

Relevant G functions:

G522: Switching over the synchronous axis to "Drive-controlled interpolation" on the current channel.
Example:
G522 X1 Z1 (switching over X and Z axis)

G521: Switching off an active "Drive-controlled interpolation" (if applicable) for all axes involved on the current channel.

G520: Specifying the position value for an axis which is currently switched to "Drive-controlled interpolation".
Example:
G520 X100 (X axis is supposed to interpolate to position X100).

G523: Specifying the speed value for an axis which is currently switched to "Drive-controlled interpolation". The value is stored in the drive under S-0-0259 and interpreted with the weighting from S-0-0044.
Example:
G523 X1000 (setting the positioning speed of the X axis to 1000 mm/min).
The unit is dependent on MP 704000010 and has to conform to the parametrized weighting from S-0-0044!

G524: Specifying the acceleration value for an axis which is currently switched to "Drive-controlled interpolation". The value is stored in the drive under S-0-0260 and interpreted with the weighting from S-0-0160.
Example:
G524 X3 (setting the acceleration of the X axis to 3 m/s^2).

☞ **The G functions G520, G523 and G524 only act block by block, whereas G521 and G522 form a group and deselect each other mutually.**

Relevant interface signals:

NC-O17.7: "Drive-controlled interpolation" (axis interface).
High signal: the interpolation takes place in the drive
Low signal: the interpolation takes place in the PNC

4.8.1 Assigning

☞ **For information on the drive parameters, please refer to your drive documentation.**
For output/writing of drive parameters, see page 4–73.

1. Make sure that secondary operating modes 2 and 3 of the drive have been parametrized correctly.
   The drive function ”Interpolation in the drive” has to be parametrized for secondary operating mode 2 as well as for secondary operating mode 3.
   Parametrization is performed in the drive parameters S-0-0034 (for secondary operating mode 2) and S-0-0035 (for secondary operating mode 3) via bit 4 to bit 7.
2. Make sure when speed data is specified (G523) that the feedrate unit set by the NC (⇒ MP 7040 00010) is interpreted correctly by the drive.
   The speed data is stored in the drive under S-0-0259 and interpreted with the weighting from S-0-0044.
3. Make sure that the acceleration data specified (G524) is interpreted correctly by the drive.
   The acceleration data is stored in the drive under S-0-0260 and interpreted with the weighting from S-0-0160.

4.8.2 Activating

1. Program the axis addresses in the G522 block for which the ”Drive-controlled interpolation” is to be activated (e.g. X1 Y1; the number after the axis name is required for programming but is not subject to any further evaluation).

Axes with active ”Drive-controlled interpolation” can now be addressed regarding position (G520), feed speed (G523) and acceleration (G524) from the remaining channels of the control unit under their physical axis names (⇒ MP 1003 00001).

☞ **Position specifications from the master channel (G520) are not permitted.**

4.8.3 Deactivating

1. Program G521.
   All axes on the channel for which ”Drive-controlled interpolation” has been activated, are switched back to ”NC-controlled interpolation” (main operating mode or secondary operating mode 1 are selected again on the drive).

4.9 Axis transfer

Function:

On one channel, the axes which are in one and the same machining-technological relationship are called **axis group**. They constitute an interpolation group. For certain machining tasks, it is necessary to change the **axis assignment**, i.e. to remove axes from one axis group and integrate them into the axis group of a different channel temporarily. In this context, it is possible to change the axis designations within an axis group.

The following **axis transfer functions**are available:

- Changing of axes between axis groups, i.e. a synchronous axis remains a synchronous axis
- removing axes from an axis group, i.e. a synchronous axis becomes an asynchronous axis
- Taking over axes into an axis group, i.e. an asynchronous axis becomes a synchronous axis
- Renaming the axes within an axis group
- Changing the axis classification (functional relevance)

Required **terms**:

- Taking over an axis: axis is **integrated** into the axis group
- Borrowing an axis: axis is **transferred**, i.e. channel borrows axis from an inactive channel
- Removing an axis: axis is **released** from present axis group
- For the meaning of the terms drive, channel, axes, axis group, auxiliary axes, synchronous and asynchronous axes, physical and logical axes, physical and logical axis index, spindle, please refer to Section 5.1.2.

Changing of axes between axis groups

When changing axes between two channels, a distinction is made between two cases:

- **The channel of the "source axis group" is not active.**
  If the channel from where the axis originates is not active, an axis can be integrated into a second axis group at any time, i.e. the axis is "borrowed" to the second channel. Once it has been released from there, it is automatically integrated into the source axis group again.

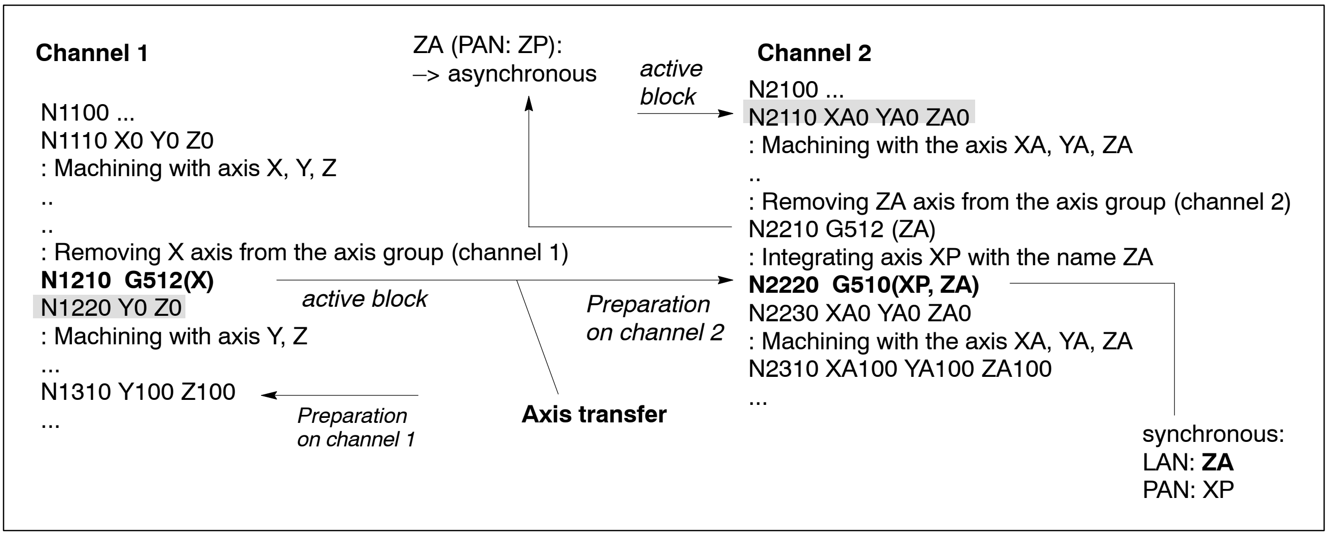
  If a program or manual input is selected in the source axis group while the axis is being borrowed, an error message will be displayed.

  With this kind of axis transfer, it is possible, e.g., to start a user reset program in a channel after a system control reset, which borrows all system axes from other channels too and moves them to a basic position .

Axis transfer

- **The channel of the "source axis group" is active.**
  When an axis is transferred from an active channel (axis group 1), the axis is first removed and in a second step integrated into a different channel (axis group 2).
  To make sure that the axis is taken over **without waiting time** in block preparation, the axis must already have been removed from axis group 1. This is the case, if the corresponding NC block in axis group 1 was already active at the time when the axis was transferred to axis group 2.

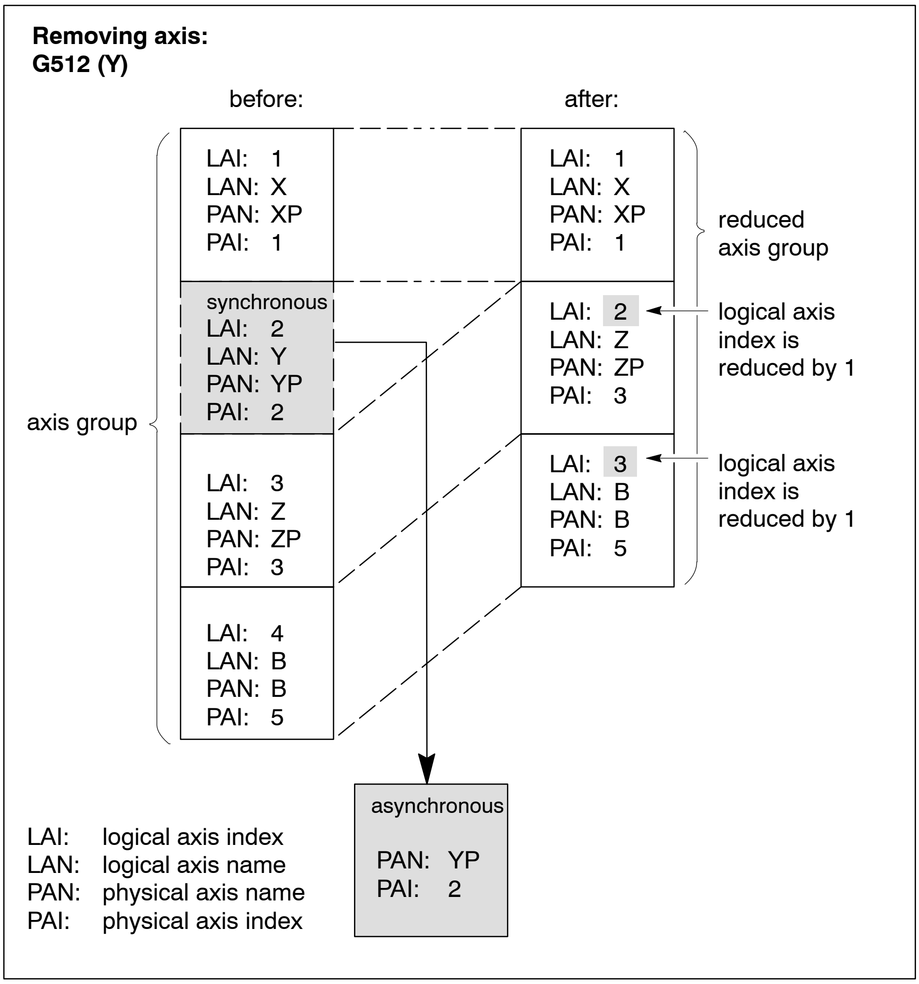
  Example:
  The X axis of channel 1 is transferred to channel 2 (see figure below). In channel 1, preparation has already reached block N1310, and block N1220 is active. Release of the X axis is thus completed.
  At this point in time, block N2110 is active on the 2nd channel. Preparation is at block N2220 and is prepared to take over the physical axis XP (previously X from channel 1).
  The axis XP is assigned the name ZA in the process. Since this name is already familiar to channel 2, there will be no waiting times until the axis is taken over.



Removing axes from an axis group

Using G512, an axis is removed from an axis group Thus a synchronous channel axis becomes an **asynchronous** axis.
Block preparation is not stopped during this action .

Example: Channel axis Y is released; it is then programmable as asynchronous axis YP on all channels. Channel axes Z and B are assigned new logical axis numbers. The Y axis disappears on a channel axis display. Programming of the logical axis Y results in an error message.



Axis transfer

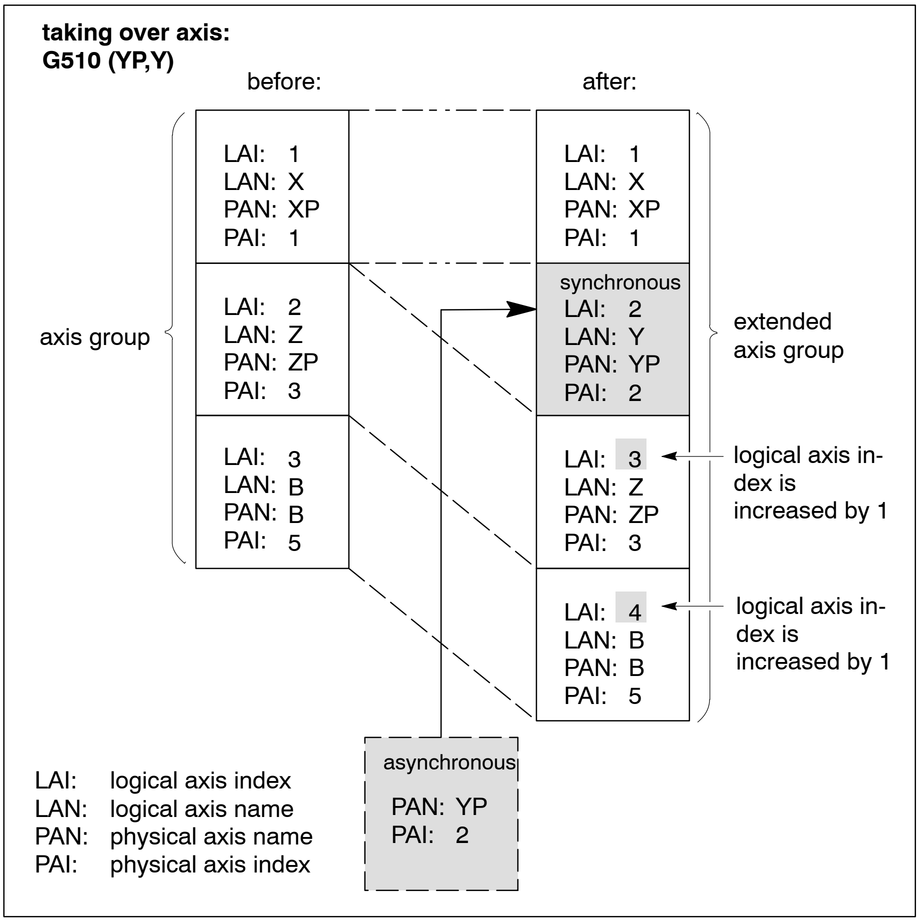
Taking over axes into an axis group

An asynchronous axis is accepted into an axis group via G510 or G511, thereby becoming a synchronous channel axis:

- In case of G510, the axis has to be standing still, otherwise there will be an error message and block preparation will be stopped.
- In case of G511, standstill of the axis is waited for implicitly.
- Optionally, a new logical axis name can be entered, which is used to address the axis on the channel. This logical name has to be predefined in one of the two channel-specific MACODA parameters 7010 00010 “Logical axis designation” or 7010 00020 “Optional axis designation”.

Example:

In the figure below, the physical axis YP is integrated into an axis group. It can then no longer be addressed as asynchronous axis YP on other channels. On the own channel, it can be programmed either with the logical or the physical axis name. The Y axis additionally appears on a channel axis display.
Channel axes Z and B are assigned new logical axis indexes.



An axis taken over is assigned the axis classification **999**, if it does not belong the default configuration of the channel. In all other cases, it receives its default classification from MACODA.

Axis transfer

Renaming the axes within an axis group

- When an axis is taken over into an axis group using G510 or G511, a new logical name can be entered optionally.
- The logical axis name can be modified directly with G515.

☞ **This logical name has to be predefined in MP 7010 00010 ”Logical axis designation“ or MP 7010 00020 “Optional axis designation”.**

Changing the axis classification

Depending on the plane selection (G17, G18, ...), a **functional relevance** (axis classification) is assigned to all logical axes of a channel via MP 7010 00030 or G21.

- Main axis, secondary axis, feed axis
- Meaning of the interpolation parameters (I, J, K)

The axis classification can be influenced by an axis transfer:

- If an axis is **removed** from an axis group which is located within a main plane, there will be an implicit change to G16 “No plane”. G16 can be used in the part program.
  It is possible to change the axis classification in the part program using G21 “Change axis classification” (see example below).
- If an axis is **taken over** into an axis group, it is at first assigned a neutral classification. It can be overwritten by programming G21 (see example below).
  When an axis is taken over, the order of the logical axes may be changed. If the functional relevance is changed in the process, a change to G16 “No plane“ takes place implicitly.

Example: G21 “Change axis classification”

`N100 G17 X0 Y0 Z0`	Default axis classification: X=1, Y=2, Z=3
`N200 G512(Y)`	Y is removed from axis group. Implicit switchover to G16. Circular interpolation not possible anymore.
`N210 G511(YA)`	The axis YA is adopted into the axis group. It is assigned a neutral axis classification.
N220 G21 YA2	YA is assigned the axis classification 2
N230 G17	Switching over to X/YA plane
N240 G2 X... YA...	Circular interpolation is possible

Axis transfer

Axis display at axis transfer

The axis display may change at axis transfer.

The reasons may be (see also examples in Section 4.9.2, page 4–42):

- The order of the current axis names and their axis assignment to the channel has changed.
- Axes have been removed from the axis group or adopted into the axis group.
- The names of the logical axes may have changed at transfer, if required.

The order of the axis display is determined in MP 6005 00020 (1.. 16). Without changes in the order, it is possible that the transfer causes the axis names to jump in the display.

The respective axis is displayed only if it is a synchronous channel axis or an asynchronous axis. If the negative value is entered in the parameter, it is always displayed.

Restrictions in axis transfer

- The functions G510 through G513 must be programmed in the NC block prior to an axis.
- Plane selection G17/18/19/20: functional relevance
- Drill axis switching G78:
  If G78is active and the drill axis is removed from the axis group, an implicit switchover to G79 takes place.
- Axis coupling G581:
  Removing an axis coupled with G581 from an axis group results in a runtime error.
- Tangential tool guidance G131:
  Removing an axis of rotation from an axis group when G131 is active results in a runtime error.
- If required, the zero shift tables have to be adapted. For details, see Section 5.6.2 page 5–35 ff.
- External zero shifts can be specified for logical axes or for system axes.
  Since the sequence of the logical axes can change in programs with axis transfer, it is recommended to specify the zero shift for system axes.
  Otherwise, it must be ensured that the order in which logical axes are entered in the PLC corresponds to the order on the channel. The order on the channel can be determined at the bit interface.
- An axis can be jogged/reference, if
  - the axis is assigned to a channel with the operating mode “Jog” or “Approaching the reference point“ , or
  - the axis is not assigned to any channel and the operating mode “Jog” or ”Approaching the reference point” is entered.

Relevant MACODA parameters (MP):

1003 00001:	Physical axis designations. Determines the axis names for max. 64 physical axes (actual axes on the machine).
1003 00002:	Default axis assignment to the channels.
7010 00010:	Logical axis designations. The coordinate axes of the machine or workpiece or of the program coordinate system are referred to as logical axes. They are limited to 8 axes per channel. In older MACODA versions without the MP 7010 00010, the physical axis designations are used instead.
7010 00020:	Optional axis designations. If unknown logical axis designations are used when axes are transferred or renamed, these have to be preset here.
6005 00022:	Display priority for axes and working range coordinates of the channel which are not involved in any axis transformation or when no axis transformation is active. 6005 00022 has priority over 6005 00020.
6005 00020:	Display priority for asynchronous axes and for axes not pertaining to the channel via signs: **+** : axis is only displayed if it pertains to the channel **–** : axis is also displayed if it has been assigned to a different channel.
6005 00021:	Display priority for working range coordinates of different channels or for working range coordinates of the own channel if they are **not** supposed to be displayed at the very top.

☞ **Within a channel, the order of display of all working range coordinates among each other is fixed. It is only possible to shift the entire unit within the display.**

Relevant G functions:

G510 (..)	Taking over axis Error message when the axis has not yet been released in its current axis group.
G511 (..)	Taking over axis with waiting until the axis is released.
G512 (..)	Removing an axis from an axis group
G513	Taking over default axis setting from MACODA.
G515 (..)	Assigning new logical axis names. This logical name has to be predefined in MP 7010 00010 “Logical axis designation” or MP 7010 00020 “Optional axis designation”.
G516 (..)	Removing logical axis names in the calling axis group again

Axis transfer

G16 Deactivating the plane selection.
Circular interpolation not possible anymore. Main or second axes can be removed from the axis group.

G21 (..) Changing axis classification

Relevant CPL functions

SD(20..): Provides the number of synchronous axes of the calling channel.

SD(21..): Provides the number of synchronous axes of a channel.

SD(22..): Provides the logical axis number of a physical axis if the physical axis is an axis of the calling channel.

SD(23..): Provides the physical axis number of a logical axis of the calling channel.

SD(24..): Provides the physical axis number of a logical axis.

SD(25..): Provides the channel of a physical axis at an active point in time.

MCODS(43, ...): Assignment axis – channel (determining the synchronous axes of a channel)

FXC(..): Direct access to axis zero shift values

SCS-, APOS-
and SPOS Different commands used to program the physical axis index (1 .. 16) or the physical axis names

CPOS; AXO;
WPOS; MPOS;
PPOS; PROBE;
COF; DPC;
SCL: Different commands used to program the logical or physical axis index

Relevant interface signals:

NC-O 19.0 – NC-O 19.3:
The channel number of the pertaining channel is coded in these axis output signals.
0 = Axis is not assigned to a channel, e.g. when it is operated as synchronous axis

4.9.1 Assigning

From a machining-technological point of view, axes can be transferred between different axis groups on the machine.

1. Assign the names in MP 1003 00001 and MP 7010 00010.
2. Logical axis designations which are not known, have to be preset in MP 7010 00020.
3. Determine default axis assignment to the channels in MP 1003 00002.

4. Determine display priority and working range coordinates in MP 6005 00020, 6005 00021 and 6005 00022.
5. Link axis output signals NC-O 19.0 – NC-O 19.3 in the PLC to provide information on the axis assignment (synchronous/asynchronous).
6. If required, adapt zero shift tables and external zero shift with regard to axis transfer.
7. Perform control unit start-up to adopt axis assignment.

4.9.2 Activating

Precondition:
With a view to program flow, the programs operating on the different channels have to be synchronized with regard to the axis transfer.
For example, the axis to be transferred has to be released before it can be taken over by the other channel. If necessary, the program of one channel has to wait for the program on the other channel to reach the transfer status.

★ The axis transfer in the program execution is activated using the functions described under “Relevant G functions”.

Example 1 :
The axis is temporarily transferred to a 2nd channel

At time t1, machining takes place with the logical axes X, Y, Z on the 1st channel and the axes U, B on the 2nd channel.
The zero shift table Tab1 is active on the 1st channel and table Tab2 with columns U and B is active on the 2nd channel.
In block N220, it is necessary to wait until the YP axis is released. This occurs at the active time of block N1210. From here on, machining is continued on the 2nd channel. YP is to be addressed as 2nd logical axis; the respective column of the zero shift table Tab 2 is effective.

At time t2, the axis transfer has been completed. YP is now displayed as 2nd logical axis in the axis display of the 2nd channel.
In block N1310, the YP axis is integrated into the first channel again. It has to be ensured in the program flow that the YP axis has been released by the 2nd channel at this point in time. If that is not the case, a runtime error will occur.

Channel 1	machining takes place on both channels simultaneously	**Channel 2**
N1100 G22 V Tab1	*Time t1*	N2100 G22 V Tab2
N1110 X0 Y0 Z0	←——————→	N2110 U0 B0
; Machining with axis X, Y, Z		*; Machining with axis U,B*
..		..
..		...
; Removing X axis from the axis group (channel 1)		*; Waiting until YP axis is released*
N1210 G512(X)	*Time t2*	N2220 G511 (YP)
N1220 Y0 Z0	←——————→	N2230
; Machining with axis Y, Z		N2230 U0 YP0 B0
...		*; Machining with axis U, YP, B*
N1310 G510(YP,Y)		...
...		N2310 G512(YP)

Axis transfer

Example 1

Physical axis index and physical axis name

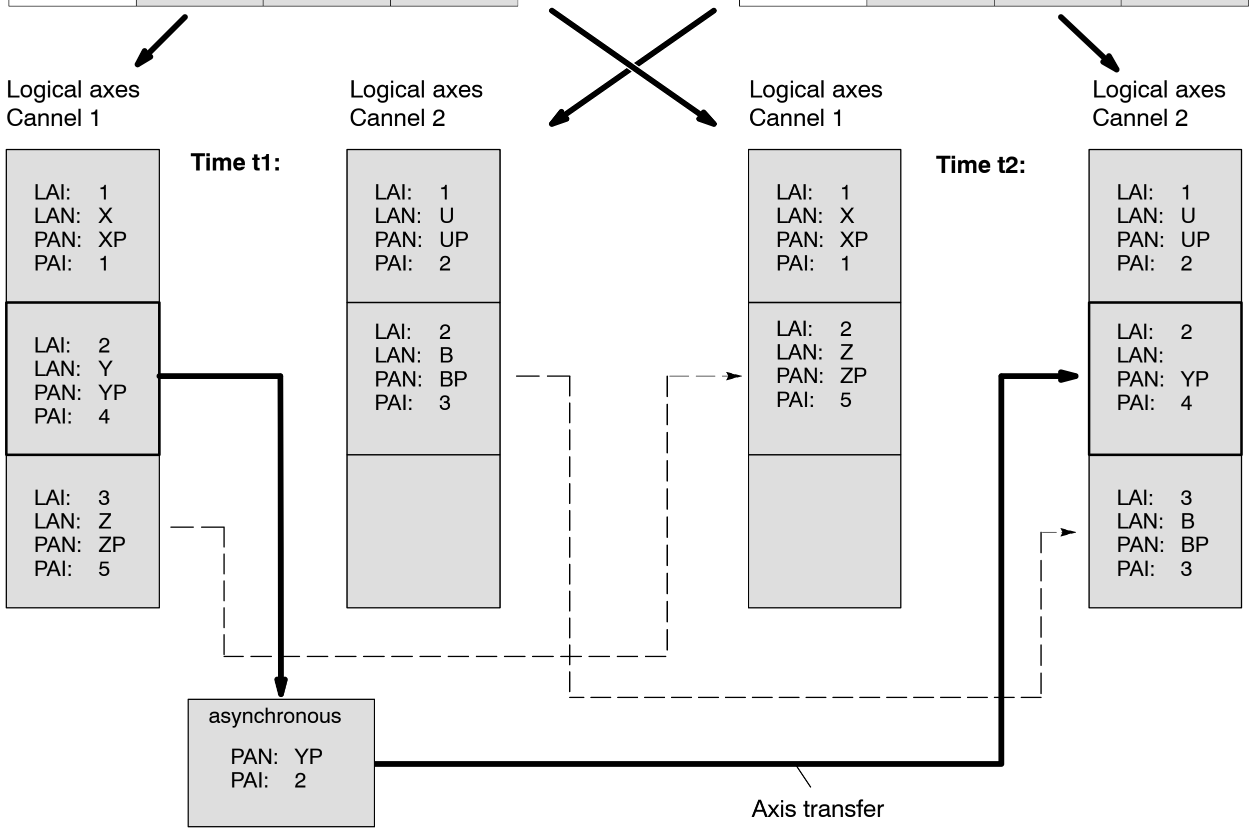
No. 1 XP	No. 2 UP	No. 3 BP	No. 4 YP	No. 5 ZP

Zero shift Tab1

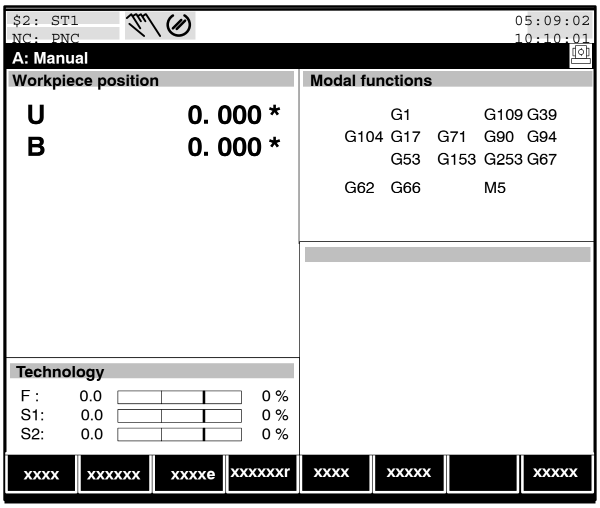
Tab1	X	Y	Z
G54	54.01	54.02	54.04
G55	55.01	55.02	55.04
...			

Zero shift Tab2

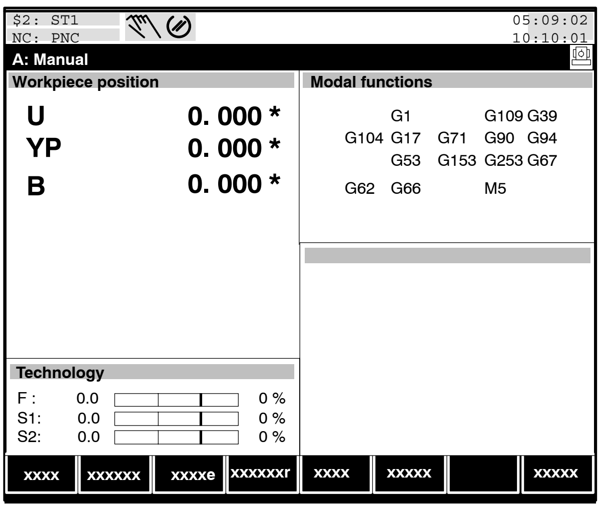
Tab2	U	B	YP
G54	54.01	54.02	54.04
G55	55.01	55.02	55.04
...			



Display: Axes of **channel 2** (at time t1)



Display: Axes of **channel 2** (at time t2)



Axis transfer

Example 2 :
Axis transfer in case of 2 real machining channels

At time t1, machining takes place with the logical axes X, Y, Z on the 1st channel and the axes U, V, W on the 2nd channel. The zero shift table Tab1 is active on the 1st channel and table Tab2 on the 2nd channel.

The Y and V axes are released in blocks N1210 and N2210. Both axes were involved in plane selection. That is why a switchover to G16 took place on both channels.
In block N1220, the ZP axis is assigned the new logical name Y.
In blocks N1230 and N2220, it is necessary to wait until the requested axis are released.
In block N1240, the axis classification and the plane are set again. X, Y and Z are shown in the axis display of the 1st channel, where Y is the axis ZP and Z the axis VP.

The order of the axis display can be determined in MACODA.

machining takes place on both channels simultaneously

Time t1: Channel 1 "; Machining with X, Y, Z" ⟷ Channel 2 "; Machining with axis U, V, W"

Time t2: Channel 2 "; Machining with axis U, V, W" ⟷ Channel 1 "; Machining with X, Y, Z"

Channel 1

```
N1100 G22 V Tab1
N1110 X0 Y0 Z0
; Machining with X, Y, Z
...
; Removing Y from axis group
N1210 G512 (Y)
; ZP is assigned the name Y
N1220 G515 (ZP,Y)
; Waiting until VP is released
; VP is assigned the name Z
N1230 G511 (VP,Z)
; Axis classification; on
N1240 G21 X1 Y2 Z3 G17
N1250 X0 Y0 Z0
N1260 G2 ...
; Machining with X, Y, Z
...
```

Channel 2

```
N2100 G22 V Tab2
N2110 U0 V0 W0
; Machining with axis U, V, W
...
; Removing V from axis group
N2210 G512 (V)
; Waiting until YP axis is released
; YP is assigned the name V
N2220 G511 (YP,V)
N2230
N2240 U0 V0 W0
; Machining with axis U, V, W
...
```

Axis transfer

Example 2

Physical axis index and physical axis names

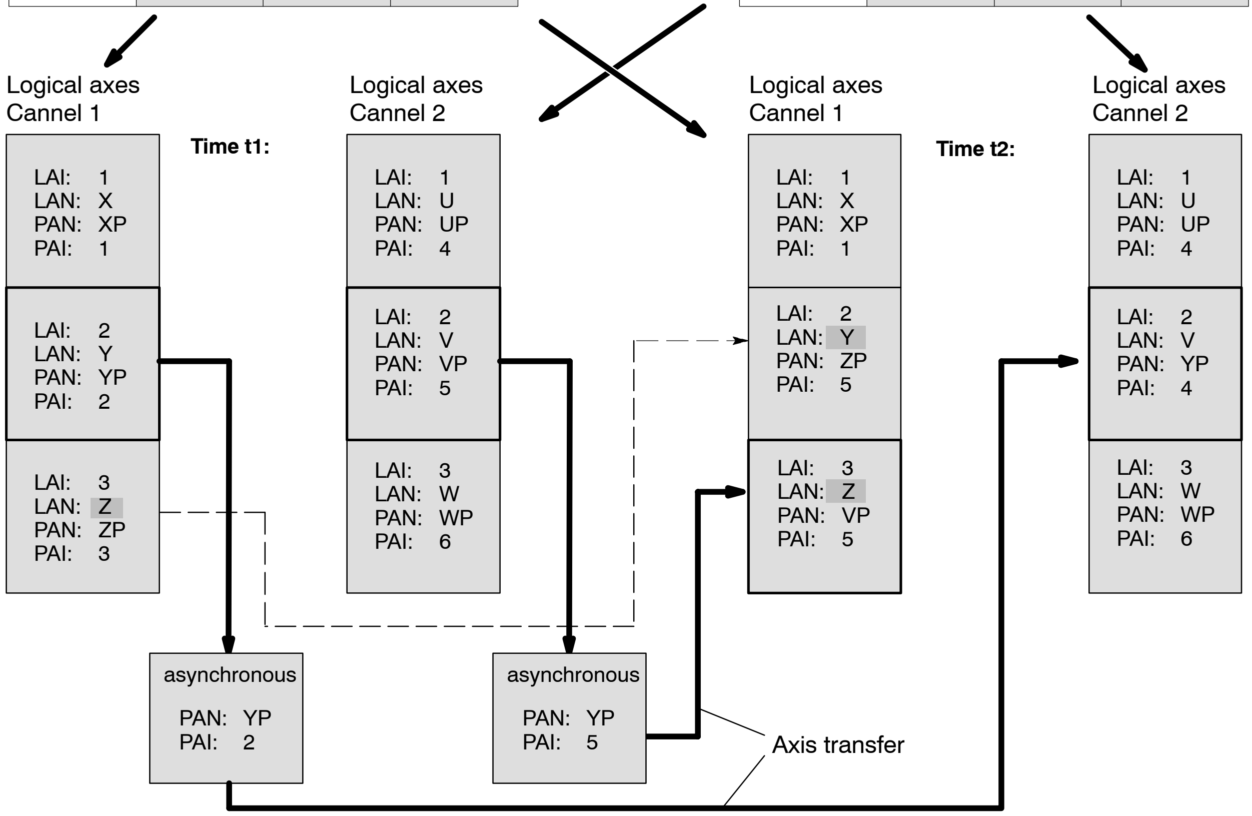
No. 1 XP	No. 2 YP	No. 3 ZP	No. 4 UP	No. 5 VP	No. 6 WP

Zero shift Tab1

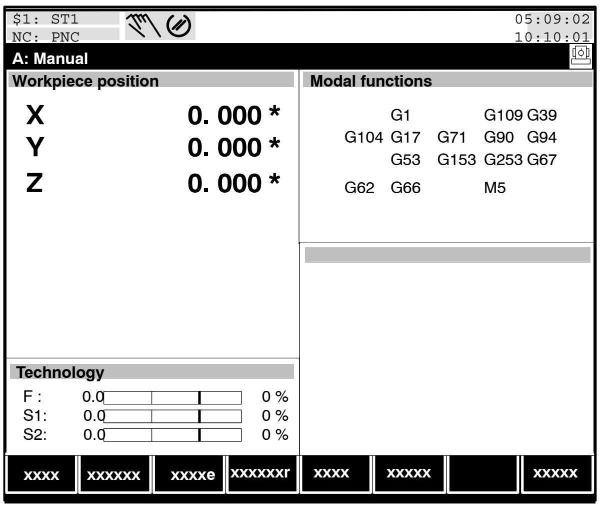
Tab1	X	Y	Z
G54	54.01	54.02	54.04
G55	55.01	55.02	55.04
...			

Zero shift Tab2

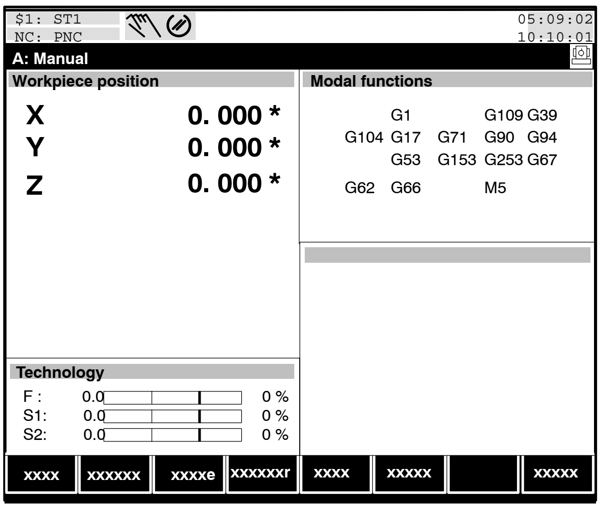
Tab2	U	V	W
G54	54.01	54.02	54.04
G55	55.01	55.02	55.04
...			



Display: Axes of **channel 1** (at time t1)



Display: Axes of **channel 1** (at time t2)



Example 3 :
Exchange of two equivalent axes on a channel

At time t1, logical axes X, Y and Z are used to work with. The X axis is physical axis X1.
The X1 axis is released in block N1210, and the X2 axis is integrated with the name X into block N1220.

Machining is possible with the X2 axis starting at time t2. The zero shift values from the column with the name X2 are valid for to this axis.

The axis display does not change.

Channel 1

```
N1100 G22 V Tab1                        Time t1
N1110 X0 Y0 Z0
; Machining with X, Y, Z
...
; Removing X from axis group
N1210 G512 (X)
; Integrating X2 with name X
N1220 G510 (X2,X)                       Time t2
N1230 X0 Y0 Z0
; Machining with X, Y, Z
...
```

Axis transfer

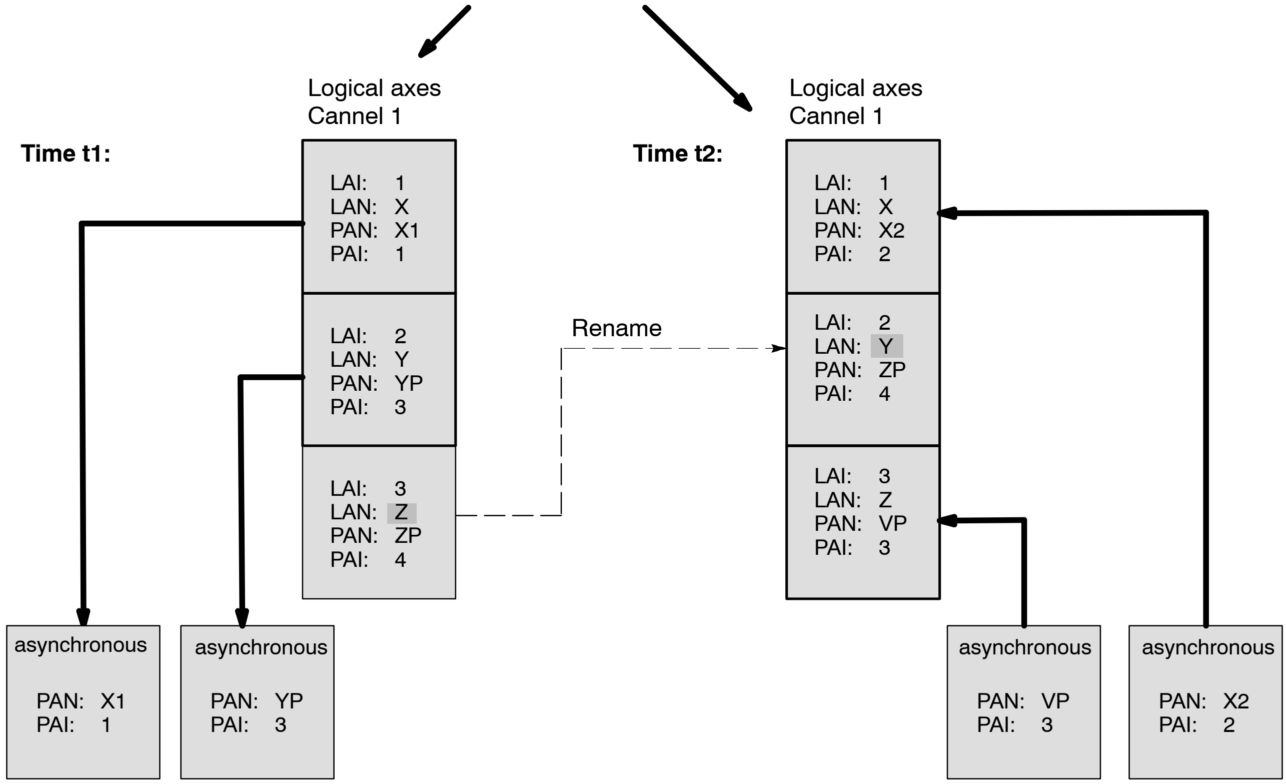
Example 3

Physical axis index and physical axis names

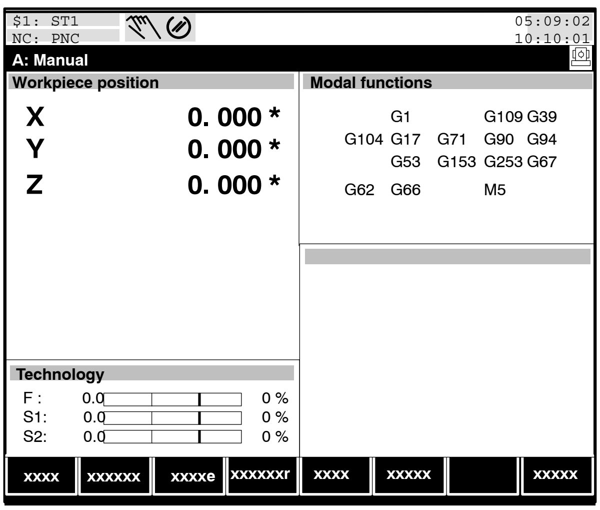
No. 1 X1	No. 2 X2	No. 3 YP	No. 4 ZP	No. 5 AS1	No. 6 AS2

Zero shift Tab1

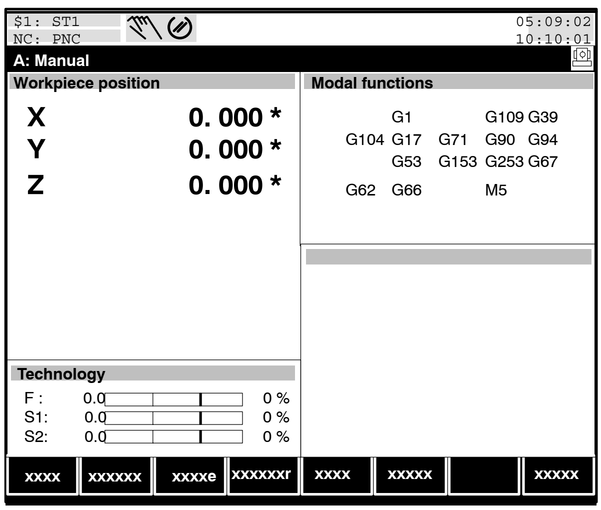
Tab1	X1	X2	Y	Z
G54	54.01	54.02	54.03	54.04
G55	55.01	55.02	55.03	55.04
...				



Display: Axes of **channel 1** (at time t1)



Display: Axes of **channel 1** (at time t2)



Axis transfer

Example 4 :
More than 8 synchronous axes on one channel

At time t1, logical axes X, Y, Z, W1, W2, W3, W4 and W5 are used to work with.

The W2, W3, W4 and W5 axes are released in block N1210, and the physical axes SY6, SY7, SY8 and SY9 are taken over with the names W2, W3, W4 and W5 in block N1230.
The zero shift values in columns SY6 to SY9 are affective from now on.

Machining is possible with the SY6 to SY9 axes starting at time t2.

Channel 1

```
N1100 G22 V Tab1                                  Time t1
N1110 X0 Y0 Z0 W1 0 W2 0 W3 0 W4 0 W5 0   <-----------
;Machining with X, Y, Z, W1, W2, W3, W4, W5
...
;Removing W2, W3, W4, W5 from axis group
N1210 G512 (W2, W3, W4, W5)
N1220 X0 Y0 Z0 W1 0
;Machining with X, Y, Z, W1

;Integrating SY6, SY7, SY8, SY9
;with the names W2, W3, W4, W5
N1230 G510 (SY6,W2, SY7,W3, SY8,W4,
SY9,W5)                                           Time t2
N1240 X0 Y0 Z0 W1 0 W2 0 W3 0 W4 0 W5 0   <-----------
;Machining with X, Y, Z, W1, W2, W3, W4, W5
...
```

Axis transfer

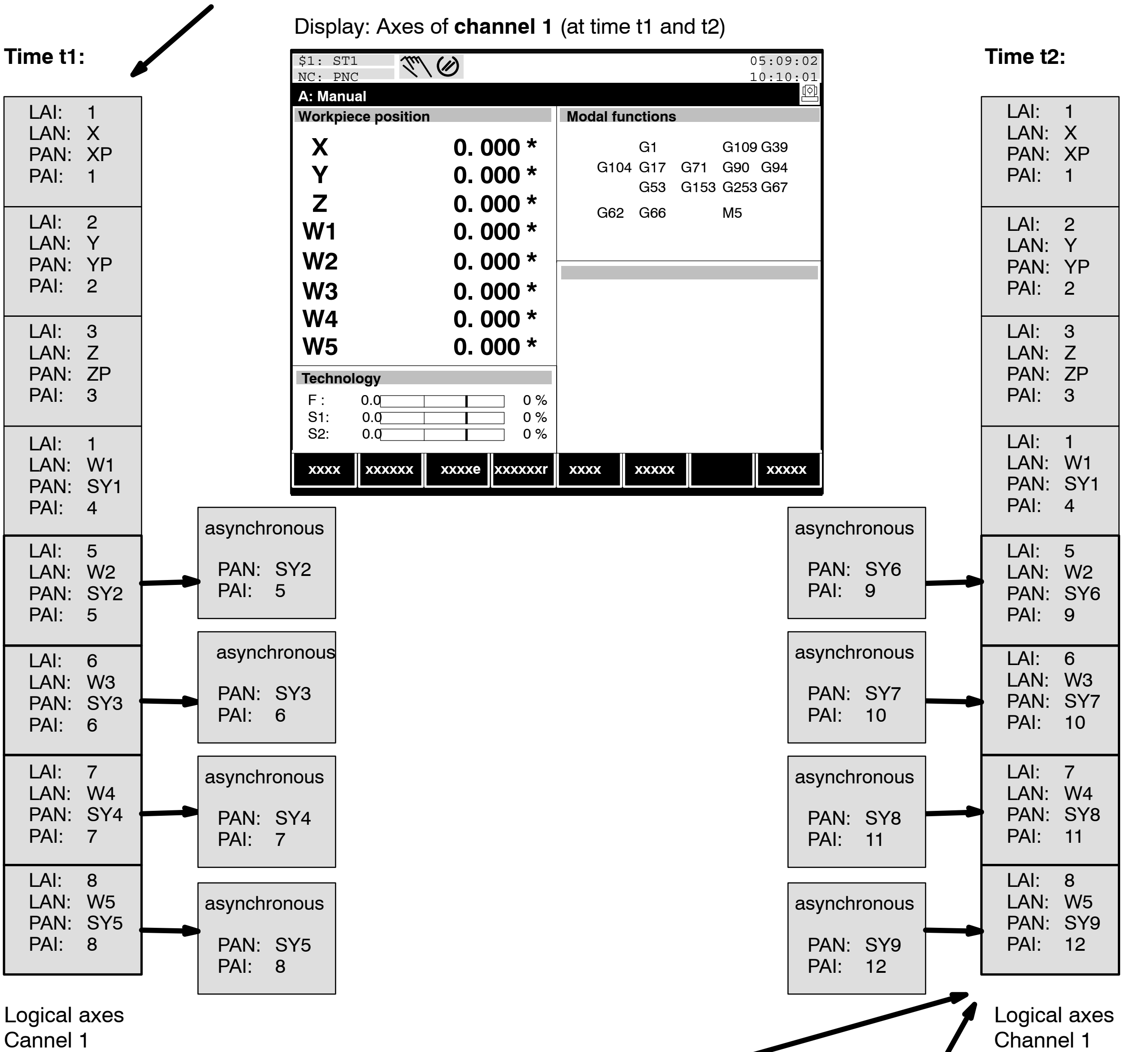
Example 4

Physical axis index and physical axis names

No. 1 XP	No. 2 YP	No. 3 ZP	No. 4 SY1	No. .. SY..	No. 16 SY16

Zero shift Tab1

Tab1	XP	YP	ZP	SY1	SY..	SY5	SY..	SY7	SY8	SY9	SY..	SY16
G54	54.01	54.02	54.03	54.04	...	54.04	...	54.04	54.04	54.04	...	54.04
G55	55.01	55.02	55.03	55.04	...	55.04	...	55.04	55.04	55.04	...	55.04
...												



Zero shift Tab1

Tab1	XP	YP	ZP	SY1	SY..	SY5	SY..	SY7	SY8	SY9	SY..	SY16
G54	54.01	54.02	54.03	54.04	...	54.04	...	54.04	54.04	54.04	...	54.04
G55	55.01	55.02	55.03	55.04	...	55.04	...	55.04	55.04	55.04	...	55.04
...												

4.9.3 Deactivating

There are different ways to deactivate the axis transfer:

- **Program G513:**
  However, this results in a runtime error, if the axis has not yet been released.

  Remedy: Write G513 in the init string, if G513 is supposed to be executed only in case of control reset (keyword #Reset:) or system control reset (keyword #SysRes).

  With **Control reset**, the current axis assignment on the channel is maintained. However, the assigned axes are released, i.e. made available to the remaining system. Other channels can request these axes.
  When a program is selected or a manual entry is made, the axes must previously have been released by the other channels, otherwise there will be an error message.
  Using **System control reset**, the axis assignment can be switched over to MACODA again.

- **End of program (M30):**
  Using G functions, it is possible to release axes pertaining to a different channel or take over the axis assignment from MACODA. The latter case may, however, result in error messages, if the axes have been assigned elsewhere.
- **Channel control reset**
  Analogous to end of program (M30). Error messages may occur here too, if axes have been assigned elsewhere.

4.10 Spindle

Function:

Assigns a connected SERCOS drive as spindle.

A spindle is a machine axis which:

- carries out rotational movements only,
- is usually operated in speed control mode,
- is used to drive the tool.

☞ **If the drive is to operate in position-synchronous operation (spindle coupling), you have to assign it as ”Spindle/C axis” (see page 4–53) with special marginal conditions.**
For detailed information on spindle coupling, please refer to the manual ”DIN Programming instructions”.

☞ **If the functionality ”polar coordinate transformation” is needed for the drive (see page 5–103), assign the drive as ”Spindle/C axis” (see page 4–53).**

☞ **”Spindle orientation” (M19) is also possible with ”Spindle” type drives. Position-controlled spindle mode will remain active within the drive until another spindle command (e.g. M3, M4, M5) becomes effective.**
For detailed information, please refer to Section ”Spindles”, ”Spindle Functions” in the DIN Programming Manual.

Relevant MACODA parameters (MP):

1001 00001: Drive function type.
1040 00001: Selection of spindle type.
1040 00002: Assignment of physical spindle to spindle group.
1040 00010: Number of gear ranges.
1040 00011: Min. spindle speed of the gear range.
1040 00012: Max. spindle speed of the gear range.
1040 00015: Jog spindle speed in rpm.
1040 00020: Speed reaches window in rpm.
1040 00021: Speed reaches window in %.
1040 00031: Max. spindle acceleration.
1040 00041: Spindle override in 32 steps.
1040 00051: Positional interface: 1st acceleration in rad/s^2
1040 00052: Positional interface: 2nd acceleration in rad/s^2
1040 00053: Positional interface: switchover speed for 2nd acceleration
1040 00101 through 1040 00112: Spindle syntax.

☞ **For more information on spindles, please refer to the manual ”DIN Programming instructions”.**

4.10.1 Assigning

Precondition:

- The SERCOS parameters have been correctly set in the drive or will be transferred by the control unit to the drive during SERCOS initialization (addressing, timing, operating mode, weightings, etc.; for information on SERCOS, see page 4–67 ff.).

  For a spindle, the type of weighting, e.g., must be set to ”rotary” and the data reference to ”at the load” in the SERCOS parameters.

  - S-0-0044 (type of weighting for speed data)
  - S-0-0076 (type of weighting for positional data)
  - S-0-0160 (type of weighting for acceleration data)

1. In MP 1001 00001, assign value ”4” to the respective drive.
2. Enter spindle type ”1” (SERCOS spindle) in MP 1040 00001.
3. Parametrize the spindle as required in terms of gear ranges, speeds, dynamics, syntax etc. (see ”Relevant MACODA parameters” above).
4. Ensure that the spindle is correctly included in the PLC program in respect of interface signals.

CAUTION
Danger of damage to the machine or the workpiece!
Do not initiate a control RESET during machining.

---

5. Initiate a control reset.

4.11 Spindle/C axis

Function:

Assigns a connected SERCOS drive as ”spindle/C axis”.

This type of application is a machine axis which

- carries out rotational movements only,
- is used alternatively as spindle (spindle mode) or as rotary/endless axis (C axis mode).

☞ **In C axis mode, the axis can be operated as asynchronous or synchronous axis. For operation as synchronous axis, it is simply taken over into a channel via function ”Axis transfer”.**

Restrictions:

- The measuring system used has to run with the real spindle speed. A motor encoder must only be used for systems without gear. If gears are used, you need an external encoder.
- When general inhibit is activated, the NC saves the current state of the drive (spindle/C axis mode). To be able to deactivate general inhibit, the current state of the drive has to coincide with the state that has been saved.
- When you switch the main spindle (⇒ MP 7020 00010) to C axis mode, G95 (feedrate programming in mm/rev) is not switched off.

Relevant MACODA parameters (MP):

For spindle mode:

1001 00001: Drive function type.

1040 00001: Selection of spindle type.

1040 00002: Assignment of physical spindle to spindle group.

1040 00010: Number of gear ranges.

1040 00011: Min. spindle speed of the gear range.

1040 00012: Max. spindle speed of the gear range.

1040 00015: Jog spindle speed in rpm.

1040 00020: Speed reaches window in rpm.

1040 00021: Speed reaches window in %.

1040 00031: Max. spindle acceleration.

1040 00041: Spindle override in 32 steps.

1040 00101 to 1040 00112: Spindle syntax.

☞ **For more information on spindles, please refer to the manual ”DIN Programming instructions”.**

Additionally for C axis mode:

1003 00001: Physical axis designation.

1003 00004: Axis movement type.

1003 00005: Positioning logic.

1003 00050: Switching the positioning logic of endless axes using G151.

1040 00060: C axis: Tool turret

Relevant G functions:

G518: Switches from spindle mode to C axis mode. See page 4–56.

G517: Switches from C axis mode to spindle mode. See page 4–57.

☞ **G518 and G517, respectively, can be entered in MP 7060 00010 (default state upon start-up) and/or 7060 00020 (default state upon a control reset). Please note in this context:**

- **the spindle is always active after control unit start-up.**
- **the entry in MP 7060 00020 is effective following control reset. In this context, please note the information on switching between C axis and spindle mode given in Section 4.11.2 and 4.11.3.**

Relevant spindle interface signals:

NC-I1.6: C axis on.
Switches over from spindle mode to C axis mode.

NC-I1.7: C axis off.
Switches over from C axis mode to spindle mode.

NC-O17.0: C axis is active

NC-O17.7: C axis switching active

Relevant SERCOS parameters:

S-0-0016: Configuration list AT.
This is where SERCOS parameter S-0-0051 (position actual value 1 motor encoder) or S-0-0053 (position actual value2 external encoder) have to be entered depending on the encoder used.

S-0-0024: Configuration list MDT.
This is where SERCOS parameter S-0-0047 (position setpoint) has to be entered.

S-0-0032: Main operating mode.
Has to be set to ”speed control” mode.

S-0-0033 to S-0-0034: Secondary operating mode 1 to 2.
Has to be set to ”position control” mode.
Bit 8 (drive-controlled change of operating mode) must not be set in S-0-0034.

S-0-0044, S-0-0076 and S-0-0160: Type of weighting...
”Rotary weighting” is required as type of weighting.

S-x-0103: Modulo value.
If ”Modulo format” has been set as processing format in S-0-0076 (bit 7=1), parametrize the modulo value as 360 degrees.

S-0-0026: Configuration list signal status word.We recommend to enter SERCOS parameter Parameter S-0-0403 (status position actual value) in S-0-0026 and cyclically transmit the signal status word (S-0-0144) in the drive telegram to the NC.

4.11.1 Assigning

Preconditions:

- The SERCOS parameters have been correctly set in the drive or will be transferred by the control unit to the drive during SERCOS initialization (addressing, timing, operating mode, weightings, etc.; for information on SERCOS, see page 4–67 ff.).

  For a spindle/C axis, the type of weighting, e.g., must be set to ”rotary” and the data reference to ”at the load” in the SERCOS parameters.

  - S-0-0044 (type of weighting for speed data)
  - S-0-0076 (type of weighting for positional data)
  - S-0-0160 (type of weighting for acceleration data)

1. In MP 1001 00001, assign value ”3” to the respective drive.
2. In MP 1003 00001, enter the physical axis designation for the C axis.
3. In MP 1003 00004, set the axis movement type to ”2” (endless) or ”3” (rotary) depending on your requirements.
   If the axis movement type 2 (endless) is required, the processing format ”Modulo format” (bit 7=1) has to be set in S-0-0076 and the value 360 (degrees) in S-0-0103 in the drive.

☞ **If the NC function ”Threading G33” is to be used, the processing format ”Modulo format” (bit 7=1) has to be set in S-0-0076 and the value 360 (degrees) in S-0-0103 in the main spindle drive.**

4. Set the desired positioning logic in MP 1003 00005.
5. Parameterize the axis as required in terms of speed, dynamics, etc.
6. Ensure that the axis is correctly included in the PLC program in respect of interface signals. The relevant axis interface has to be operated by the PLC in C axis mode. The information whether C axis mode is active is received by the PLC via the corresponding spindle interface signal NC-O 17.0.
7. Enter spindle type ”1” (SERCOS spindle) in MP 1040 00001.
8. Set MP 1040 00060 (C axis : tool turret) to value ”0” (normal mode: no turret).

9. Parametrize the spindle as required in terms of gear ranges, speeds, dynamics, syntax etc. (see "Relevant MACODA parameters" above).
10. Ensure that the spindle is correctly included in the PLC program in respect of interface signals.

CAUTION
Danger of damage to the machine or the workpiece!
Do not initiate a control RESET during machining.

---

11. Initiate a control RESET.

4.11.2 Switching over from spindle mode to C axis mode.

1. Program G518 in the part program or via manual input.
   For information on G518 syntax, see DIN Programming Manual.

   In special cases, the PLC can trigger the switching over from spindle mode to C axis mode via the spindle interface signal NC-I1.6.

☞ **Spindle reset has no effect in the C axis mode.**

After switchover, the axis is located at a random position between 0 and 359.9999 degrees.
The axis has not been assigned to any channel yet (asynchronous axis) and is therefore available to any channel as auxiliary axis.

If you trigger the switchover in the part program, we recommend that you use WAIT commands (see example). It is thus ensured that the axis is actually available when the block is processed in the further program sequence.

Example:

:

`N100 G518(WA)`	The spindle, which has the name WA as axis, is switched over to C axis mode as asynchronous axis.
`N110 WAITA(IC(24,5,x))`	Waits until the spindle interface signal NC-O17.0 (C axis is active) has been set. x: spindle index
`N120 WAIT`	Waits until WAITA() has been executed. Then, the axis is actually available.

2. If the asynchronous axis is to be taken over into the active channel, you have to program WAIT and subsequently G511.
   For information on G511 syntax, see DIN Programming Manual.

Spindle/C axis

4.11.3 Switching over from C axis mode to spindle mode.

1. Make sure that the axis has not been assigned to a channel.
   If the axis is still assigned to a channel, program G512.
   The axis is thus released from the active channel and is available as asynchronous axis to all channels in the system.
   For information on G512 syntax, see DIN Programming Manual.
2. Program G517.
   This is how you switch back to spindle mode.
   G517 can be programmed after G512 in the same block, for information on syntax, see DIN Programming Manual.

☞ **The spindle can only be addressed by all channels or via spindle interface when the G517 block has been executed.**

Example:

`N100 G512 (C)`	Removing the C axis from an active channel.
`N110 G517 (C)`	Switching back to spindle mode.
`N120 WAIT`	Waits until G517 has been executed.

4.12 Spindle/turret axis with motor encoder

Function:

Assigns a connected SERCOS drive as ”spindle/turret axis”. The drive has to be applied as ”spindle/C axis” (see page 4–53).

A ”spindle/turret axis” is a machine axis which

- carries out rotational movements only,
- is used alternatively as spindle (spindle mode) or as rotary/endless axis (C axis operation); here: turret mode).
- drives the tool in spindle mode and the tool turret (of a lathe) in C axis mode.

Controlling a tool turret requires a change to turret mode both within the NC and mechanically, where the turret has its own transmission ratio.

In spindle mode, the drives moves a tool designed as cutter. In case of a tool change, the drive is switched to C axis mode; at the same time, the PLC mechanically applies the motor to the turret axis.

In this state, the NC/PLC can now carry out the tool change via C axis.

Having switched back to spindle mode (NC and mechanically), it is possible to continue machining of the workpiece.

☞ **In C axis mode, the turret axis can be operated as asynchronous or synchronous axis. For operation as synchronous axis, it is simply taken over into a channel via function ”Axis transfer”.**

Restrictions:

- **Servodyn-D drives** have to be used.
- The drive must support the command S-0-0197 (set coordinate system). Servodyn-D meets this condition starting at drive software version V0.048.
- Since the contents of the CPL variable @_C_AXISxy_POS is overwritten when loaded into an archive (file extension *.tar), the turret axis has to be referenced after the loading process.

CAUTION
Danger of damage to the machine.
The CPL variable @_C_AXISxy_POS must not be manipulated at random. Reference the turret axis.

- The motor encoder is used as measuring system for spindle and turret axis.
- When general inhibit is activated, the NC saves the current state of the drive (spindle/C axis mode). To be able to deactivate general inhibit, the current state of the drive has to coincide with the state that has been saved.
- When you switch the main spindle (⇒ MP 7020 00010) to C axis mode, G95 (feedrate programming in mm/rev) is not switched off.

Spindle/turret axis with motor encoder

Relevant MACODA parameters (MP):

For spindle mode:

1001 00001: Drive function type.
1040 00001: Selection of spindle type.
1040 00002: Assignment of physical spindle to spindle group.
1040 00010: Number of gear ranges.
1040 00011: Min. spindle speed of the gear range.
1040 00012: Max. spindle speed of the gear range.
1040 00015: Jog spindle speed in rpm.
1040 00020: Speed reaches window in rpm.
1040 00021: Speed reaches window in %.
1040 00031: Max. spindle acceleration.
1040 00041: Spindle override in 32 steps.
1040 00101 through 1040 00112: Spindle syntax.

☞ **For more information on spindles, please refer to the manual ”DIN Programming instructions”.**

Additionally for C axis mode:

1003 00001: Physical axis designation.
1003 00004: Axis movement type.
1003 00005: Positioning logic.
1003 00050: Switching the positioning logic of endless axes using G151.
1040 00060: C axis: tool turret

Relevant G functions:

G518: Switches from spindle mode to C axis mode (here: turret mode). The required parametrization of MP 1040 00060 (setting: 2) results in the following sequence:
- Spindle is decelerated to a stop.
- Parameter block 4 in the drive is activated
- Turret mode is activated
- The axis position from the CPL variable @_C_AXISxy_POS is taken over

G517: Switches from C axis mode to spindle mode (here: turret mode). The required parametrization of MP 1040 00060 (setting: 2) results in the following sequence:
- the current position of the turret axis is written in the CPL variable @_C_AXISxy_POS
- spindle mode is activated
- in the drive, the parameter block (gear range) which was active last in spindle mode, is activated again.

☞ **The change between spindle mode and turret mode always entails a mechanical change, too. This connection must not be eliminated**

by control reset (channel or system control reset).
Therefore, G518 and G517, respectively, must be entered neither in MP 7060 00010 (default state upon start-up) nor in 7060 00020 (default state upon a control reset).

Relevant spindle interface signals:

NC-I1.6: C-Axis on.
Switches over from spindle mode to C axis mode (here: turret mode).

NC-I1.7: C axis off.
Switches over from C axis mode (here: turret mode) to spindle mode.

NC-O17.0: C axis is active

NC-O17.7: C axis switching active

Relevant SERCOS parameters:

☞ **In the drive, the spindle mode is configured via the parameter block 0 through 3 (1st to 4th gear range).**
Parameter block 4 is relevant for turret mode.

S-0-0016: Configuration list AT.
This is where SERCOS parameter S-0-0051 (position actual value1 motor encoder) has to be entered.

S-0-0024: Configuration list MDT.
This is where SERCOS parameter S-0-0047 (position setpoint) has to be entered.

S-0-0032: Main operating mode.
Has to be set to ”speed control” mode.

S-0-0033 to S-0-0034: Secondary operating mode 1 to 2.
Has to be set to ”position control” mode.
Bit 8 (drive-controlled change of operating mode) must not be set in S-0-0034.

S-0-0044, S-0-0076 and S-0-0160: Type of weighting...
”Rotary weighting” is required as type of weighting.

S-x-0103: Modulo value.
If ”Modulo format” has been set as processing format in S-0-0076 (bit 7=1), parametrize the modulo value as 360 degrees.

S-0-0026: Configuration list signal status word.We recommend to enter SERCOS parameter S-0-0403 (status position actual values) in S-0-0026 and cyclically transmit the signal status word (S-0-0144) in the drive telegram to the NC.

4.12.1 Assigning

Preconditions:

- The SERCOS parameters have been correctly set in the drive or will be transferred by the control unit to the drive during SERCOS initialization (addressing, timing, operating mode, weightings, etc.; for information on SERCOS, see page 4–67 ff.).

  For a spindle/C axis, the type of weighting, e.g., must be set to ”rotary” and the data reference to ”at the load” in the SERCOS parameters.

  - S-0-0044 (type of weighting for speed data)
  - S-0-0076 (type of weighting for positional data)
  - S-0-0160 (type of weighting for acceleration data)

- A suitable PLC program called up by an auxiliary function with compulsory acknowledgment is required for the mechanical switchover between spindle mode and C axis mode (here: turret mode).

1. In MP 1001 00001, assign value ”3” to the respective drive.
2. In MP 1003 00001, enter the physical axis designation for the C axis.
3. In MP 1003 00004, set the axis movement type to ”2” (endless) or ”3” (rotary) depending on your requirements.
   If the axis movement type 2 (endless) is required, the processing format ”Modulo format” (bit 7=1) has to be set in S-0-0076 and the value 360 (degrees) in S-0-0103 in the drive.
4. Set the desired positioning logic in MP 1003 00005.
5. Parameterize the axis as required in terms of speed, dynamics, etc.
6. Ensure that the axis is correctly included in the PLC program in respect of interface signals. The relevant axis interface has to be operated by the PLC in C axis mode. The information whether C axis mode is active is received by the PLC via the corresponding spindle interface signal NC-O 17.0.
7. Enter spindle type ”1” (SERCOS spindle) in MP 1040 00001.
8. Set MP 1040 00060 (C axis: tool turret) to value ”2”.
9. Parametrize the spindle as required in terms of gear ranges, speeds, dynamics, syntax etc. (see ”Relevant MACODA parameters” above).
10. Ensure that the spindle is correctly included in the PLC program in respect of interface signals.
11. If required, specify the current position of the turret axis by directly writing (CPL) the variable @_C_AXISxy_POS (in degrees).

CAUTION
Danger of damage to the machine or the workpiece!
Do not initiate a control RESET during machining.

12. Initiate a control RESET.

4.12.2 Synchronization after control unit start-up

Following start-up of the control unit, the NC is always in spindle mode. Since it cannot be presumed that the motor (on spindle or turret axis) has also changed over mechanically to spindle mode at this point in time, both components need to be synchronized by the PLC.

For this purpose, the PLC should first determine the mechanical state (spindle/turret mode) and carry out suitable action to achieve synchronization.

The **following mechanical** states are possible:

1. Spindle mode.
   The motor is mechanically applied to the spindle.
2. Turret mode.
   The turret axis is located **on a notch position**.
   The motor is mechanically applied to the turret axis.
3. Turret mode.
   The turret axis is located **between 2 notch positions**. The motor is mechanically applied to the turret axis.
4. Intermediate position.
   The mechanical system is neither in spindle mode nor in turret mode.

Actions to achieve synchronization:

ref. 1.: no action necessary because the NC is always in spindle mode following start-up.

ref. 2.: NC must be adapted to the state of the mechanical system.

- PLC switches to turret mode via spindle interface signal NC-I1.6 (C axis on). This initiates the following actions in the NC:
  - Parameter block 4 in the drive is activated
  - Turret mode (C axis mode) is activated
  - The axis position from the CPL variable @_C_AXISxy_POS is taken over
- The PLC should reference the turret axis for safety reasons.

ref. 3.: NC must be adapted to the state of the mechanical system.

- PLC switches to turret mode via spindle interface signal NC-I1.6 (C axis on). This initiates the following actions in the NC:
  - Parameter block 4 in the drive is activated
  - Turret mode (C axis mode) is activated
  - The axis position from the CPL variable @_C_AXISxy_POS is taken over
- When the NC sets the spindle interface signal NC-O17.0 (C axis is active), the PLC has to start referencing the turret axis.
  The NC takes over the position of the turret axis into the CPL variable @_C_AXISxy_POS simultaneously with the

output of the signal NC-O15.6 (Reference point was reached) at the axis interface of the turret axis.

ref. 4.: The PLC has to change the mechanical system either to the spindle mode or the turret mode. Depending on the resulting mechanical position, the appropriate action has to be taken subsequently (see item 1. through 3.).

CAUTION
Wrong turret positioning possible!
If the position of the turret axis has been shifted since the last switch of the NC from turret mode to spindle mode for any reason whatsoever, the contents of the variable @_C_AXISxy_POS no longer coincides with the current position of the turret axis!
This may cause serious damage to the machine or tools.
The turret axis has to be referenced, just in case the contents of the variable @_C_AXISxy_POS no longer coincides with the current position of the turret axis!
For reasons of safety, we recommend referencing the turret after control unit start-up.

4.12.3 Determining the turret axis reference point

Precondition:

- NC and mechanical system have to be in turret mode.

The reference point is determined in the same way as for ”normal” axes.

Upon completion of referencing, the reference point position is automatically taken over into the CPL variable @_C_AXISxy_POS.

4.12.4 Examples: Changing between spindle mode and turret mode

The tool change and thus the change between spindle mode and turret mode normally occurs in context with a tool changing program in the following sequence:

1. Stop spindle with the function ”Spindle orientation”.
   The auxiliary function (M219) used in the example programs below should have compulsory acknowledgment. An alternative is to wait for the spindle interface signal ”Spindle is oriented”.
2. Switch the NC to turret mode (C axis mode) via G518.
3. Use an auxiliary function with compulsory acknowledgment to instruct the PLC to switch the mechanical system to turret mode.
4. Wait until the NC and the mechanical system have switched over.
5. Carry out tool change (incl. activation of the tool compensation).
6. Change back to spindle mode (NC and mechanical system).

Spindle/turret axis with motor encoder

Example 1: turret axis is an asynchronous axis (auxiliary axis)

`:`	
`N010 ...`	
`N020 M219`	Cutting spindle orientation.
`N030 ...`	If necessary, wait for ”Spindle is oriented”. Don’t forget logic in case of error.
`N040 G518(REV)`	Switches NC from spindle mode to turret mode (C axis mode). Name of the turret axis: REV
`N050 Mxxx`	Instruct PLC to switch mechanical system to turret mode.
`N060 WAITA(IC(24,5,x))`	Wait for ”C axis is active” x: Spindle index.
`N070 WAIT`	Wait until WAITA() has been executed.
`N160 ...`	Carry out tool change.
`:`	
`N200 G517(REV)`	Switches NC from turret mode to spindle mode. Name of the turret axis: REV
`N210 Mxxx`	Instruct PLC to switch mechanical system to spindle mode.
`N220 WAITA(IC(24,5,x)=FALSE)`	Wait until ”C axis is active” is deactivated. x: spindle index.
`N230 WAIT`	Wait until WAITA() has been executed.
`M30`	End of program.

Example 2: turret axis is a synchronous axis

`:`	
`N010 ...`	
`N020 M219`	Cutter spindle orientation.
`N030 ...`	If necessary, wait for ”Spindle is oriented”. Don’t forget logic in case of error.
`N040 G518(REV)`	Switches NC from spindle mode to turret mode (C axis mode). Name of the turret axis: REV
`N050 Mxxx`	Instruct PLC to switch mechanical system to turret mode.
N055 WAIT	Wait for the interface signal of the PLC, block preparation stopped
`N060 G511(REV)`	Integrate turret axis REV into the active channel.
`N160 ...`	Carry out tool change.
`:`	
`N200 G512(REV) G517(REV)`	Release turret axis REV from the channel and switches NC from turret mode to spindle mode.
`N210 Mxxx`	Instruct PLC to switch mechanical system to spindle mode.
`N220 WAIT`	Wait until all blocks have been executed.
`M30`	End of program.

4.13 Gear range switching with ”Spindle idling”

Function:

Reduces the wear of a spindle gear.

When engaging a different gear range, the gear wheels may not lock accurately. In this case, an (optimum) coupling of forces between input and output shaft is not ensured. At a ”normal” start of the spindle motor, this may cause wear of the respective gear wheels and definitely results in an unnecessary load on the gear wheel edges.

For this reason, the control unit outputs the required gear range and transmits the command for ”Drive-controlled oscillation” (S-0-0190; bit 0 and 1 = 1) to the PLC when the gear range stated in the feedback does not coincide with the gear range requested.
From this point in time on, the drive carries out a parametrizable movement profile independently (see ”Relevant SERCOS parameters”). Once the movement profile is activated in the drive, the control unit reports this status to the PLC via spindle interface output signal NC-O15.4 (Idling speed reached), thus enabling the PLC to carry out the mechanical switchover.

As soon as the gear range stated in the feedback coincides with the requested gear range, the NC withdraws the command ”Drive-controlled idling” (S-0-0190; bit 0 and 1 = 0).

Relevant MACODA parameters (MP):

1040 00010: Number of gear ranges.

1040 00011: Min. spindle speed of the gear range.

1040 00012: Max. spindle speed of the gear range.

1040 00107 through 1040 00112: Spindle syntax.

1040 00207 through 1040 00212: Spindle group syntax.

☞ **For detailed information on gear ranges and their syntax, please refer to Section ”Spindles”, ”Gear Functions” in the DIN Programming Manual.**

Relevant spindle interface signals:

NC-I5.0 to NC-I5.3: Gear range x acknowledge.

NC-I5.7: Gear idle motion acknowledge.

NC-O14.5: Gear range switching active.

NC-O15.4: Idling speed reached.

NC-O16.0 to NC-O16.3: Output gear range x.

NC-O16.7: Output gear in idle motion.

Gear range switching with ”Spindle idling”

Relevant SERCOS parameters:

S-0-0190: Command ”Drive-controlled oscillation”.
Switches the drive function on/off.

S-0-0213: Oscillation speed.
This speed results in a sinusoidal superimposition of the oscillation offset speed.

S-0-0214: Oscillation offset speed.
Basic speed of the drive function.

S-0-0215: Oscillation cycle time.
Period duration of the sinusoidal superimposition.

4.13.1 Assigning

Preconditions:

- The function ” Drive-controlled oscillation” (S-0-0190) is available in the drive.
- The spindle has to be correctly included in the PLC program in respect of its interface signals.

1. The relevant SERCOS parameters have been correctly set in the drive or will be transferred by the control unit to the drive during SERCOS initialization (for information on SERCOS, see page 4–67 ff.).
2. Make sure that the MACODA parameters mentioned above have been parametrized correctly. If that is the case, no further settings are necessary within the NC.
Otherwise, adapt the parameters in accordance with the respective application.

CAUTION
Danger of damage to the machine or the workpiece!
Do not initiate a control RESET during machining.

---

3. Initiate a control RESET.

4.13.2 Activating/deactivating

”Spindle idling” is automatically initiated by the NC when the requested gear range does not coincide with the gear range stated in the feedback.
If they coincide, the NC deactivates the function automatically.

4.14 Functions for SERCOS handling

The PNC offers a series of functions that will considerably facilitate the handling of connected SERCOS drives:

- SERCOS initialization for all connected SERCOS drives during control unit start-up.
- Automatic parameter download for all connected SERCOS drives during control unit start-up.
- Program-controlled read access to SERCOS parameters of all connected drives through the CPL commands SCS(..) and SCSL(..). Program-controlled write access to SERCOS parameters of all connected drives through G900.

  For detailed information, please refer to the CPL or DIN Programming manual.
- Manual read and write access to SERCOS parameters of all connected drives through the ”SERCOS monitor”.
- Selected setting of a desired SERCOS operating phase in the ”SERCOS monitor”. If desired, the control unit can execute the following actions additionally:
  - Delete diagnostics class 1 errors
  - Parameter download for SERCOS operating phase 2 (file ”sysgrph2.scs” required. See also page 4–70 ff.)
  - Parameter download for SERCOS operating phase 3 (file ”sysgrph3.scs” required. See also page 4–70 ff.)
- Add-on debug functions which can be used to log
  - the SERCOS parameters transmitted in phases 2 and 3 (see page 4–74 ff.)
  - log the SERCOS timing in phase 3 (see page 4–75 ff.)
    during the start-up of the control unit.

4.14.1 SERCOS initialization

Function:

The SERCOS initialization of the PNC has the following tasks:

- Closing of all SERCOS rings. This serves to test whether the optical fiber transmission links are in order.
- Establishing the SERCOS timing and the communication paths to the connected drives.
- Switching the phases of all SERCOS drives up to phase 4 (cyclical operation has been activated).

☞ **The SERCOS initialization can be carried out with and without an automatic parameter download into the drive. For information on the automatic parameter download, please refer to 4–70 ff.**

Relevant MACODA parameters (MP):

1050 00001: Drive manufacturer.
1050 00002: Drive established in the ring.
1050 00003: Assignment of drives to SERCOS ring.
1050 00004: Assignment of drives to SERCOS address
1050 00010: Automatic SERCOS timing
1050 00011: Transmission time of drive messages
1050 00012: Transmission time of master data telegram
1050 00021: Timeout for SERCOS start-up
1050 00022: SCS files – allow disable of phase 3 download
1050 00031: Optical transmission power
1050 00032: Baud rate
9030 00001: NC cycle time:

4.14.2 Assigning

Parametrize and start initialization:

1. Switch off the power supply of the drives.
2. Adapt MACODA parameter group 1050 to your application.
   The function ”Automatic SERCOS timing” (MP 1050 00010) serves to facilitate SERCOS commissioning. When ”Automatic SERCOS timing” is activated, please note:
   - The SERCOS transmission times T1 to T4 are calculated by the control unit automatically during start-up, based on the message lengths (DT, MDT) and the drive protection times (T1min, TATMT, TMTSG, T4min).
   - The following becomes ineffective:
     1050 00011: Transmission time of drive messages
     1050 00012: Transmission time of master data telegram
   - If you use the automatic parameter download (see page 4–70 ff.), the following entries have to be deleted in the files sent to the drives in phase 2:
     S-0-0007 (measuring time actual values)
     S-0-0008 (time for setpoint value valid).
     In the standard configuration, these entries are found in file ”p2common.scs” (data sent to **every** drive in phase 2).
3. Set the matching SERCOS address and baud rate at the individual drives (in case of Servodyn drives: at the personality module).
   Use the information provided in the drive documentation to do this.
   The SERCOS baud rate of all drives has to be in accordance with the value from MP 1050 00032 (default value: 2 MBaud).
4. Make sure that DIP switch ”M” on the personality module of the Servodyn-D drives is set to ON: drive is operated through the SERCOS interface.

CAUTION
With activated parameter download, current drive parameters are overwritten by parameter values contained in the SERCOS files of the control unit.
In case of SERCOS files which have not been adapted, this may lead to unintended or hazardous states on the machine.
Therefore, deactivate the automatic parameter download for all drives where adapted SERCOS files do not yet exist in the control unit.
For information on the deactivation of the parameter download, please refer to page 4–72.

5. Initiate a control RESET.
6. After PNC start-up, check which SERCOS phase is specified. Select DIAGNOSTICS ▶ AXIS DIAGNOSTICS ▶ SERCOS-MONITOR. Ideally, the control unit and all connected drives should display phase ”4”. In Servodyn drives, the current SERCOS phase is displayed on the front side of the inverters.

   If phase 4 is not displayed, please note:

   - The SERCOS monitor displays phase ”0”:
     The control unit cannot close the ring (transmitted MST cannot get back into the control unit).
     Check whether all SERCOS stations in the ring are switched on, the optical fiber connections have been installed correctly and the optical transmission power of all modules (PNC: MP 1050 00031, Servodyn drives: personality module) is sufficient for the ring length used.
   - The SERCOS monitor displays phase ”1”:
     The control unit is unable to find a specified drive.
     Check if all SERCOS addresses parametrized in the control unit (MP 1050 00004) correspond to those of the drives.
   - Drive displays phase ”2”:
     The control unit was unable to establish a communication path or correct timing with the drive, or it was not possible to switch the drive up to phase 3 because of a parametrizing problem.
     Check the drive parametrization (if automatic parameter download has been activated, check the respective SERCOS file for phase 2 in the control unit; if applicable, activate the log function for transmitted SERCOS parameters: see page 4–74).
   - Drive displays phase ”3”:
     It was not possible to switch the drive up to phase 4 because of a parametrizing or timing problem.
     Check the drive parametrization (if automatic parameter download has been activated, check the respective SERCOS file for phase 3 in the control unit; if applicable, activate the log function for transmitted SERCOS parameters: see page 4–74).
     For the purpose of diagnostics, use the log function for SERCOS timing (see page 4–75).

☞ **If drives have not been parametrized correctly yet, leave their power supply switched off!**

4.14.3 Automatic parameter download

Function:

When the parameter download is activated, the control unit is able to completely parametrize the connected drives during start-up.
The prerequisite being that the corresponding adapted SERCOS files (*.scs) exist in the control unit. SERCOS files are always stored in ASCII format.

CAUTION
With activated parameter download, current drive parameters are overwritten by parameter values contained in the SERCOS files of the control unit.
In case of SERCOS files which have not been adapted, this may lead to unintended or hazardous states on the machine.
Therefore, deactivate the automatic parameter download for all drives where adapted SERCOS files do not yet exist in the control unit.
For information on the deactivation of the parameter download, please refer to page 4–72.

Assigning

The two files ”sysgrph2.scs” and ”sysgrph3.scs” are of central importance for the parameter download.
In these files, you determine which SERCOS files are transmitted to which drives through command ”use...”

For transmission

- the file ”sysgrph2.scs” is responsible in phase 2 and
- the file ”sysgrph3.scs” is responsible in phase 3.

Syntax of the ”use” command:

```
use <File name> for (<Number> [,<Number>])
```

SERCOS file whose content is to be sent to drive *<number>*.

System drive index.
If *<file name>* is to be sent to several drives, the individual numbers have to be separated by commas.

Command for downloading.
If the command is in ”sysgrph2.scs”, the contents of *<file name>* is sent in phase 2.
Ff the command is in ”sysgrph3.scs”, the contents of *<file name>* is sent in phase 3.

☞ **MP 1050 00022 (”SCS files – Allow disable of phase 3 download”) determines the behavior of the control unit with regard to all SERCOS files specified in ”sysgrph3.scs”.**
If you allow the disable, the files will only be transmitted to the re-

spective drive on demand (e.g. following a change). For the criteria for this, please refer to Section 4.14.7.

All the files which can be used in ”sysgrph2.scs” and ”sysgrph3.scs” for the automatic download via the ”use” command, may

- contain comment lines and
- lines for drive parametrization.

Comment lines always start with a semicolon (;). All signs after the semicolon till the end of the line will be interpreted by the PNC as comment. Comments are not transmitted to the drives. They only serve to provide a structure and overview.

The lines for drive parametrization are structured as follows:

<ID number> **=** *<Value>* *[***;** *<Comment text>]*

The following applies:

- *<ID number>*: SERCOS parameter in format S-x-xxxx or P-x-xxxx (S or P parameter). For available SERCOS parameters, please refer to the drive documentation.
- *<Value>* : Parameter values in the following formats:
  decimal: e.g. 500
  binary: e.g. 0b...
  string: e.g. ”text”
  Parameter list: e.g. (S-0-0047,S-0-0189)
- *<Comment text>*: comment has to be separated from *<value>* at least by one blank and a semicolon.

Examples:

`; This is a comment line`	(comment)
`S-0-0121 = 1 ;initial rot.`	(parameter with comment)
`S-0-0122 = 1`	(decimal value)
`S-0-0032 = 0b0011`	(binary value)
`S-0-0142 = ”Application type”`	(String)
`S-0-0016 = (S-0-0051,S-0-0189)`	(ID list)

★ If **all** SERCOS files (*.scs) required and adapted for your application are available already, copy all relevant SERCOS files into the ”Root directory” (”/”) of the PNC (for the directory structure, please refer to the ”Operating instructions” manual).
For activation or deactivation, please read the respective sections below.

★ If **no** SERCOS files (*.scs) have been created for your application, you have to create these files in the ”Root directory” (”/”) of the PNC via editor or copy and adapt the supplied example files into the”Root directory” (”/”).

☞ **You will find example files in the directory ”FEPROM” (”`/feprom`”) in the subdirectory ”scsbosch” (for Servodyn drives) or ”scsindra” (for Ecodrive and DIAX drives).**

Activating

1. Remove the sign ";" to the left of the relevant "use" commands in the files "sysgrph2.scs" or "sysgrph3.scs", respectively (for syntax, see above).
   As a result, the control unit will not interpret these signs as comment lines **during the next start-up** but as command for downloading.
2. Save the modified file(s).
3. Make backup copies of the files.

Example:

Automatic downloading (in phase 2) of the files

- "p2linall.scs" into the drives with system drive index 1, 2 and 4
- "p2lin3.scs" into the drive with system drive index 3.

Excerpt from file "sysgrph2.scs" (phase 2!):

```
:
use p2linall.scs for (1,2,4)
use p2lin3.scs for (3)
:
```

Deactivating

1. Set the sign ";" to the left of the relevant "use" commands in the files "sysgrph2.scs" or "sysgrph3.scs", respectively (for syntax, see above).
   As a result, the control unit will not interpret these signs as command for downloading **during the next start-up** but as comment lines.
2. Save the modified file(s).
3. Make backup copies of the files.

Example:

Automatic download for the two files from the previous example is to be deactivated.

Excerpt from file "sysgrph2.scs" (phase 2!):

```
:
;use p2linall.scs for (1,2,4)
;use p2lin3.scs for (3)
:
```

☞ **If backup copies of all relevant SERCOS files exist in the "User FEPROM" ("`/usrfep`"), it is not sufficient to simply delete or rename the relevant SERCOS files in the "Root directory" of the control unit, if you want to deactivate the automatic parameter download (the control unit will search for these files in the "User FEPROM" as well if it doesn't find them in the root directory)!**

4.14.4 Changing parameters in the drive via SERCOS monitor

During commissioning, it is sometimes necessary to display or modify drive parameters.
For these activities, you can use the SERCOS monitor of the PNC. It allows for

- read and write access to all drive parameters as well as
- joint manual switching of all drives to a certain SERCOS operating phase.

☞ **Read or write access to drive parameters is principally only possible if the SERCOS ring is at least in phase 2.**
Please note that changing a drive parameter is only permitted in a specific phase.
For information on this, please refer to the drive documentation.

The SERCOS monitor is started via
DIAGNOSTICS ▶ AXIS DIAGNOSTICS ▶ SERCOS MONITOR.

For information on how to read/write drive parameters or change the SERCOS operating phase, please refer to the manual ”Operating Instructions – Diagnostics Tools”, Chapter ”SERCOS Monitor”.

☞ **Parameters written via SERCOS monitor will at first be changed in the drive only temporarily.**
As a rule, they will be cancelled again by RESET at the drive, a new start-up of the drive or a new parameter download of PNC.

If the changed drive parameters are to be effective permanently, please note:

During start-up, the drive normally copies all parameter values from the drive’s own FEPROM into the RAM.
Since the SERCOS monitor of the PNC exclusively changes the data on the drive RAM, you have to

- initiate the command ”Save working memory” at the respective drive (via the drive’s own commissioning software; for procedure, please refer to drive documentation)
  if the parameter download has been **deactivated**

or

- change the relevant parameter values in the respective SERCOS files of the PNC and save the modified files,
  if the parameter download has been **activated**.

4.14.5 Logging the transmitted SERCOS parameters

All SERCOS parameters transmitted in phase 2 or phase 3, may be logged for diagnostics purposes.
For this purpose, you have to program the ”dbg” command in the file ”sysgrph2.scs” (for phase 2) or ”sysgrph3.scs” (for phase 3).

☞ **The ”dbg” command has to be programmed before the first ”use” command.**

Syntax of the ”dbg” command:

dbg <*File name*>

Name of the file containing the log.

Command ”Create log”.
If the command is in ”sysgrph2.scs”, the log function for phase 2 is activated.
If the command is in ”sysgrph3.scs”, the log function for phase 3 is activated.

☞ **Parameters sent to the drive based on MACODA values cannot be logged.**

Activating

1. Remove the sign ”;” to the left of the ”dbg” command in the files ”sysgrph2.scs” or ”sysgrph3.scs”, respectively.
   As a result, the control unit will not interpret these signs as comment lines **during the next start-up** but as command to log the SERCOS parameters.
2. Save the modified file(s).

Deactivating

1. Set the sign ”;” to the left of the ”dbg” command in the files ”sysgrph2.scs” or ”sysgrph3.scs”, respectively.
   As a result, the control unit will interpret the line as comment line **during the next start-up**.
2. Save the modified file(s).

4.14.6 Logging of SERCOS timing

☞ **Use this function for commissioning only!**

For diagnostics purposes, the PNC can log the SERCOS timing from phase 3 on.
To do this, you have to program the "opt" command in file "sysgrph3.scs".

☞ **The "opt" command has to be programmed after the last "use" command.**

Syntax of the "opt" command:

opt –m <*File name*>

Name of the file containing the timing log.

Command "Logging of SERCOS timing"

Activating

1. Remove the sign ";" to the left of the "opt" command in the file "sysgrph3.scs".
   As a result, the control unit will not interpret these signs as comment lines **during the next start-up** but as command to log the SERCOS timings.
2. Save the modified file.

Deactivating

1. Set the sign ";" to the left of the "opt" command in the file "sysgrph3.scs".
   As a result, the control unit will interpret the line as comment line **during the next start-up**.
2. Save the modified file.

4.14.7 Downloading SERCOS files in phase 3 ”on demand only”

Relevant MACODA parameters (MP):

1050 00022: SCS files – allow disable of phase 3 download

The function minimizes the required start-up time of the machine with activated parameter download.
SERCOS files which are normally **always** transmitted to the drives in phase 3 (defined in ”sysgrph3.scs”) **are only sent** by PNC if the parameters in this kind of file most probably differ from the current drive configuration.

The file ”saveph3.scs” in PNC plays a central role in detecting differences of this type:

- Contents of ”saveph3.scs”:
  The name and date of all SERCOS files to be transmitted for phase 3 are stored in ”saveph3.scs”. During a SERCOS start-up, the control unit compares these data with the name and date of the respective current SERCOS files. If these data differ from each other, the respective SERCOS files were probably changed (or at least loaded into an editor and saved again); PNC then sends the corresponding SERCOS files to the drive again and updates ”saveph3.scs”.
- Date of file ”saveph3.scs”:
  When the command ”Save working memory” is initiated via softkey, the drive will write its data contained in the RAM into its FEPROM (otherwise the current drive parametrization will be lost when the drive is switched off or reset).
  Drive parameter S-0-0142 (type of application), which contains the date of ”saveph3.scs” is also considered here.
  During a SERCOS start-up, the PNC reads S-0-0142 from the drive and compares the value with the current date of ”saveph3.scs”.
  If these data differ from each other, ”Save working memory” was not initiated via softkey following an update of ”saveph3.scs”; PNC then sends all SERCOS files for phase 3 to the drive again.

5 Coordinate systems

5.1 Basics and overview

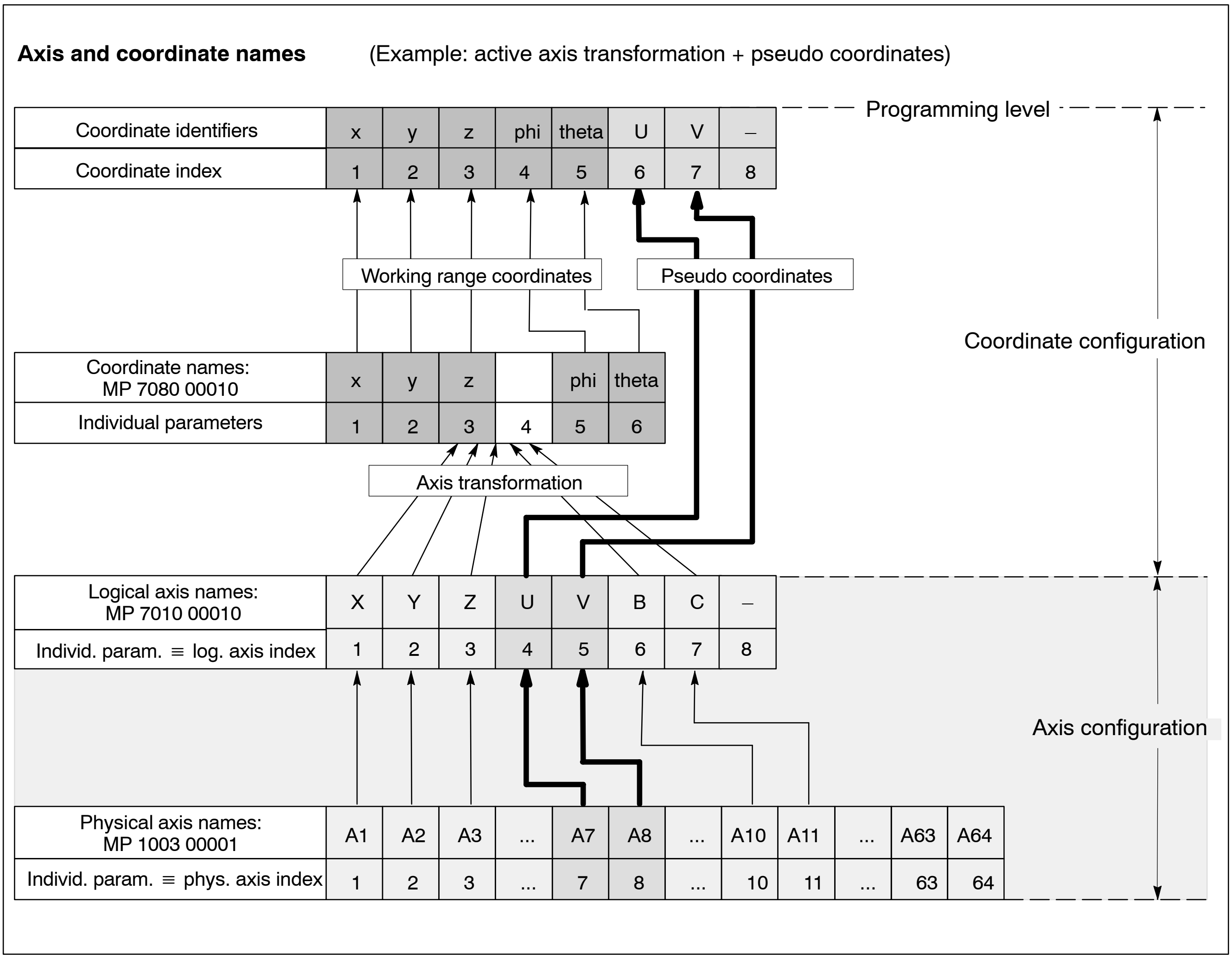
5.1.1 Axis and coordinate names

The following designations are used in the PNC:

- physical axis names
- logical axis names
- coordinate names
- coordinate identifiers

Depending on the application, either axis names or coordinate names are used in connection with specific NC functionalities.

Overview: Axes and coordinates



5.1.2 Physical and logical axes

The drives of a machine tool can be subdivided into two types:

- axes
- spindles

The term **axes** refers in this context to the entirety of the drives controlled by position setpoint values. **Spindles**, in contrast, use a speed interface. The term of spindles/C axes refers to spindles which can be selected both in spindle operation (speed interface) as well as in C axis operation (positional interface). The specific type of drive is determined in MP 1001 00001, ”Drive function type”.

The terminology of the PNC makes a distinction between **physical and logical axes**.

Physical axes

Physical axes are also referred to as **system axes**. Unless its display is suppressed by machine parameter settings, each physical axis corresponds to a real SERCOS axis. Its axis/drive index (1...n) is unique throughout the system and corresponds to the index of the respective individual drive or axis parameters in the MACODA (function area 10, machine). The ”**Physical axis designation**” MP 1003 00001 assigns a unique, system-wide name to each physical axis.

The ”Channel assignment” MP 1003 00002 can be used to accurately assign each physical axis to a machining **channel**. On one channel, all the axes which are in one and the same machining-technological relationship are grouped. The aggregate of all axes of one channel is referred to as **axis group**. Considering that different axis groups are assigned to different channels, it is possible for several axis groups to perform various machining operations independently of each other and in parallel. In this context, the channel can only select the axis group which is assigned to it.

Physical axes which are not assigned to a specific channel are referred to as **asynchronous** axes or **auxiliary axes**. These can be selected from any channel, but are in no interpolary relationship to other axes. Auxiliary axes may, for instance, be used to drive tool or pallet changers.

Logical axes

The axes of an axis group are referred to as the logical axes of the channel. Since there is an interpolary relationship between the axes of a specific channel, they are also referred to as **synchronous** axes.
On a channel, all logical axes are numbered consecutively. This **logical axis index** results from the order of the **physical axis indices**, i.e. on one channel the axis with the lowest physical axis index will receive the logical axis index of 1. Consequently, the axis with the highest physical axis index will receive the highest logical axis index.

When a new axis is transferred to a channel, the logical axis indices, including the physical axis index of the new axis, are **resorted** in accordance with the **new** order of physical axis indices of all existing axes.

MP 7010 00010, "Logical axis designation", allows to define a **logical axis name** for every logical axis on a channel. In this context, a logical axis name is assigned to every logical axis index.
Every logical axis name must be unique on the respective channel. If no logical axis name has been defined for a channel axis, it implicitly takes over the physical axis name as logical axis name. An axis can be selected both under its logical and under its physical axis name.
In the case of identical names, the logical name always takes precedence over the physical name!

Logical axes can be selected from no other channel than from the one they have been assigned to via MACODA or a G function (please refer to axis group above).

The assignment of axes to channels as set in the MACODA can be changed using the "Axis transfer" functionality (G510-G518) in the part program.

Example of an axis configuration:

The logical axis names for each channel are taken from MP 7010 00010!

Physical axis/drive index = MACODA parameter index	1	2	3	4	5	6	7	8
Physical axis name (1003 00001)	X1	Y1	X2	Y2	–	Z1	Z2	E
Channel assignment (1003 00002)	1	1	2	2	–	1	2	0
Drive function type (1001 00001)	axis	axis	axis	axis	spindle	axis	axis	axis
Logical axis index Channel (7010 00010)	**1**	**2**	–	–	–	**3**	–	–
Logical axis name Channel 1	X	Y	–	–	–	Z	–	–
Logical axis index Channel2 (7010 00010)	–	–	**1**	**2**	–	–	**3**	–
Logical axis name Channel 2	–	–	X	Y	–	–	Z	–
Axis type	synchr. axis	synchr. axis	synchr. axis	synchr. axis		synchr. axis	synchr. axis	asynchr axis

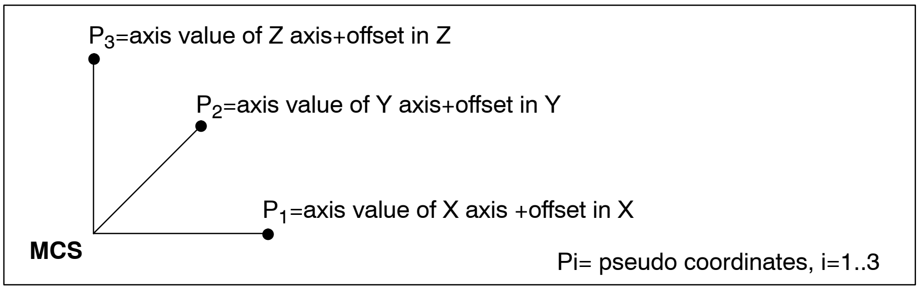
5.2 Coordinate identifiers

Coordinate identifiers describe one point (position of the tool) on the machine via working range coordinates and/or pseudo coordinates. They are used in the NC program.

5.2.1 Pseudo coordinates

Direct programming of axis positions is referred to as programming in pseudo coordinates. **In the process,** the pseudo coordinates stand parallel with the coordinates of the MCS machine coordinate system. The given positions P_i correspond directly to the axis values A_i, except for an axis specific offset (compensation):

P_i=A_i+offset.



If the machine is composed of further axes (e.g. rotary axes), the procedure described above is extended to include these axes.

The channel-specific names of the pseudo coordinates are determined in MP 7010 00010 and correspond to the logical axis names.

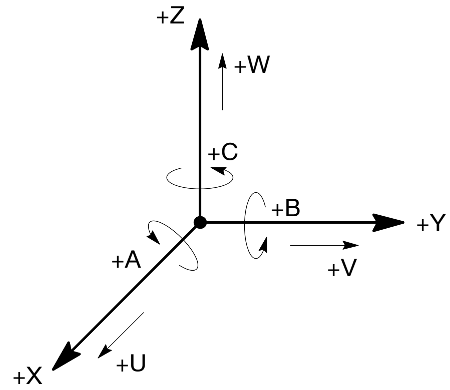
Coordinate designation

The majority of machines have a Cartesian axis structure, i.e. the first 3 axes, which are normally designated ”X”, ”Y” and ”Z” form a rectangular trihedral.

In DIN 66217, the arrangement and designation of the coordinates of the machine coordinate system are defined as follows:

- right-handed rectangular coordinate system with the linear X, Y and Z axes which are assigned the
- axes of rotation A, B and C.

Coordinate identifiers



- If the machine is equipped with a minimum of one spindle, the **Z direction of movement** is parallel with the main spindle, otherwise it is perpendicular to the clamping surface.
- **Positive axis direction** corresponds to the direction of the workpiece relative to the tool.
- The **X direction of movement** is parallel with and horizontal to the workpiece clamping surface. In analogy to the above coordinate system, the **Y direction of movement** results from X and Z.
- The X, Y and Z axes are also referred to as main axes.
- If other independently controlled axes exist, which are parallel with the main axes, they are referred to as U, V and W, and if there are even more axes of this description, they are called P, Q and R. Rotation movements around the main axes (e.g. rotary axis) executed by machine components are referred to as A, B and C.
- If there are other rotary movements in addition to the rotary A, B and C main movements, they are called D or E (address letter must be available).
- The **sense of rotation** is mathematically positive if one looks from one coordinate vertex in the direction of the coordinate origin and the sense of rotation is counter-clockwise.
- Non-parallel linear axes can be designated freely as convenient.
- **Indexing**: In the case of machines with various positioning or parallel-motion options, these movements can be indexed using a letter and a number (e.g. X1). The index has to be an integer and greater than zero. The main movements can be performed with or without an index. This way, it is possible to have indexed and non-indexed movements on one and the same machine. Indexing can be used for axes of rotation, too.
- Certain address letters (e.g. G, M and F) cannot be used for movements.

Coordinate identifiers

5.2.2 Working range coordinates

The programming in working range coordinates is independent of the axis configuration. The following is programmed:

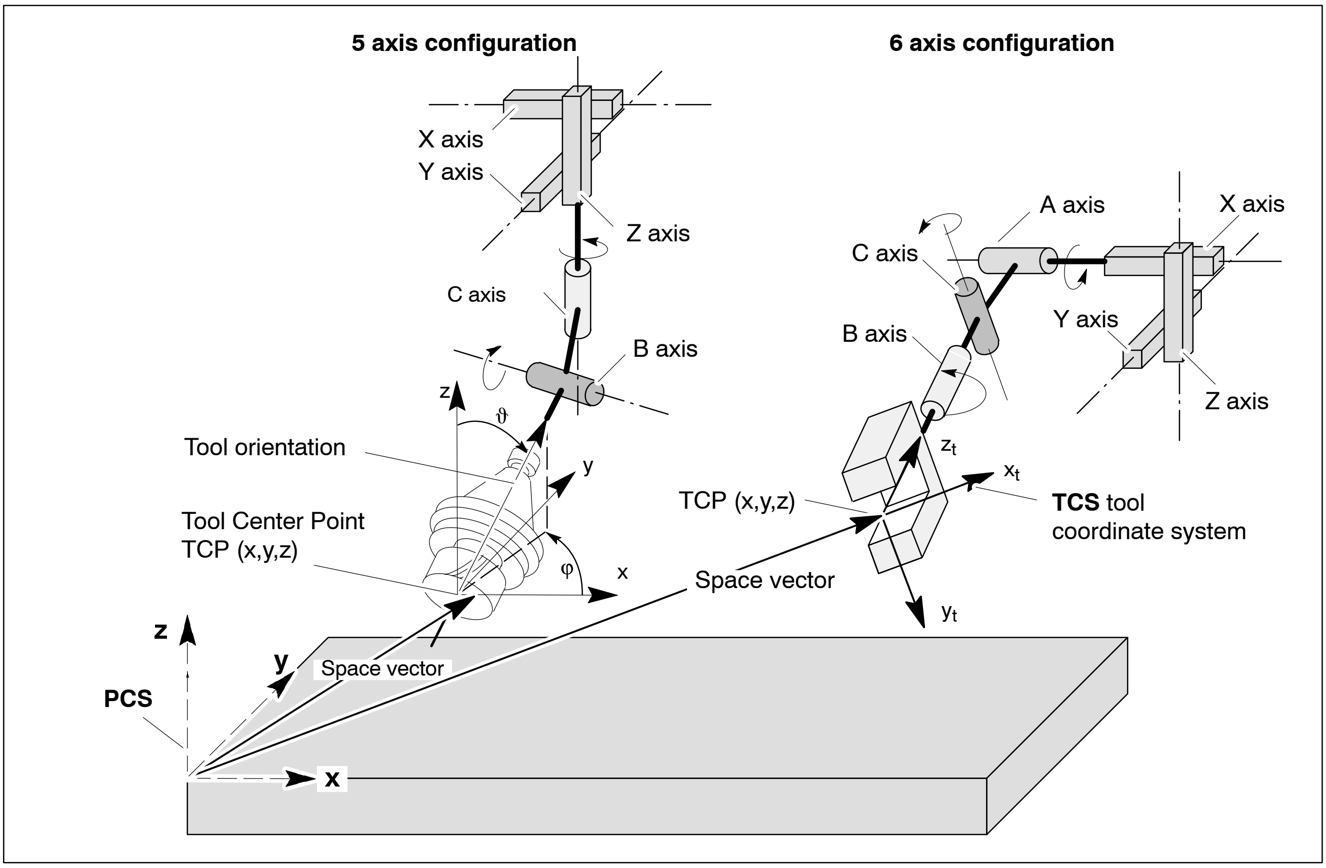
- **position** of the tool tip (TCP) and
- **orientation** of the tool.

The working range coordinates refer to a Cartesian coordinate system with a maximum of 6 degrees of geometrical freedom ($R_1..R_6$), namely 3 translational and 3 rotary degrees of freedom:

- **translational degrees of freedom**: 3 position coordinates (Cartesian coordinate: x, y, z) describe the position of the tool tip in relation to the current program coordinate system (PCS).
- **rotary degrees of freedom**: 3 orientation coordinates (see Section 5.5) describe the orientation of the tool in space. The orientation is determined by the 3 so-called Eulerian angles φ, ϑ und ψ (polar angles). These are also related to the program coordinate system (PCS).

Example:

- 5 axis transformation: cutting tool (rotation symmetrical) with 3 position coordinates (x,y,z) and 2 orientation coordinates (φ, ϑ).
- 6 axis transformation: gripping tool (non-rotation symmetrical) with 3 position coordinates (x,y,z) and 3 orientation coordinates (φ, ϑ and ψ).



Coordinate identifiers

In working range coordinate programming, the number of program coordinates is less than or equal to the number of logical axes on the channel. Channel-specific names of the working range coordinates are determined in the MACODA parameter 7080 00010.

Programming an "axis transformation" activates the working range coordinates. In the PNC, there are 5 axis, 6 axis transformations as well as polar coordinate transformation.

Example: 5 axis milling machine with 5 axis transformation

Working range coordinates: positions R_1, R_2, ...R_n
Pseudo coordinates: axis values P_1, P_2, ... P_n

In the case of a 5 axis machine tool, one has to distinguish between 3 **typical** types of position settings:

1. P= (P_1, P_2, P_3, ..., P_n) —> program coordinates = pseudo coordinates

   There is no transformational relationship between the program coordinates and the axis values. All of the program coordinates are pseudo coordinates.

   Example:
   P= (P_1, P_2, P_3, P_4, P_5, P_6, P_7) = (X_A, Y_A, Z_A, A_A, B_A, U_A, V_A).
   Axis values are programmed directly for all program coordinates. There are only pseudo coordinates.

2. P= (R_1, R_2, R_3, R_4, R_5) —> program coordinates = working range coordinates

   There is a transformational relationship for all coordinates between the program coordinates and the axis values. Programming thus becomes independent of the actual axis kinematics. All program coordinates are programmed as working range coordinates. The axis values are calculated from the programmed (Cartesian) working range coordinates with the aid of an **axis transformation**.

   Example:
   P= (R_1, R_2, R_3, R_4, R_5) = (x, y, z, phi, theta).
   All program coordinates are programmed as working range coordinates.

3. P= (R_1, R_2, R_3, P_4, P_5, ..., P_n) —> program coordinates = pseudo and working range coordinates

   There is only a transformational relationship as regards "position-creating program coordinates". These are programmed as working range coordinates. For the other program coordinates, programming consists again in the direct programming of the axes values (pseudo coordinates).

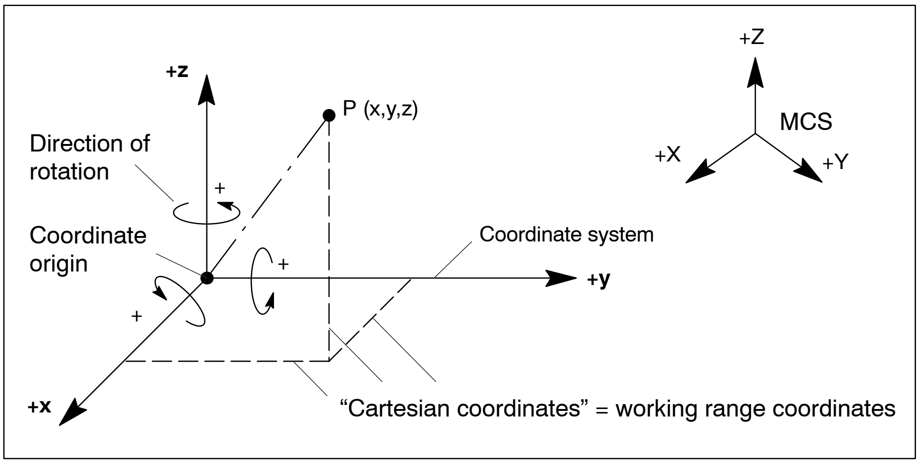
   Example:
   P= (R_1, R_2, R_3, P_4, P_5, P_6, P_7) = (x, y, z, B_A, C_A, U_A, V_A). The "position-creating program coordinates" are programmed as working range

coordinates. As a consequence, the position of the tool tip is independent of the position of the rotary axes (**T**ool **C**enter **P**oint programming). For the rotary axes and other axes, programming consists in the direct programming of the axes values (pseudo coordinates).

☞ **Index A means that the address corresponds to a pseudo coordinate, i.e. directly to a logical axis address.**

Cartesian coordinates

The "Cartesian coordinates" of point P (x,y,z) refer to its distance to three coordinates standing perpendicularly on each other. The coordinates form a right-handed Cartesian rectangular coordinate system with the working range coordinates x, y, z.



Rotation around Cartesian coordinates:
The **sense of rotation** is mathematically positive if one looks from one coordinate vertex in the direction of the coordinate origin and the sense of rotation is counter-clockwise.

5.3 Coordinate systems of the control unit

To be able to execute part programs without any modifications using different tool dimensions or for different machines, different coordinate systems are defined for each channel in the control unit (for an overview, please refer to page 5–15). In part, these are permanent and machine-dependent, the majority, however, is freely definable and machine-independent.

Coordinate systems may be related to:

- axis coordinates or
- workpiece coordinates

The **axis coordinates** are defined by the entirety of all axes currently assigned to a machining channel.
A traversing motion entered in the direction of an axis coordinate will only move the real (physical) axis pertaining to this coordinate.

The **workpiece coordinates** are often Cartesian coordinates and are defined by the coordinates currently programmed on the channel.

Overview of the used coordinate systems with their reference coordinate system. The transitions between the different coordinate systems are performed by the control unit internally with the aid of “transformations”:

Coordinate system		**Reference coordinate system**	
PCS	Program coordinate system	Workpiece coordinates	$WCS^{(n)}$
Coordinate transformation			
BCS	Basis workpiece coordinate system	Workpiece coordinates	$WCS^{(0)}$
Axis transformation			
ACS	Axis coordinate system	Axis coordinates	$ACS^{(n)}$
Axis zero shifts			
MCS	Machine coordinate system	Axis coordinates	$ACS^{(0)}$

Coordinate transformation

Using different coordinate transformations (see Section 5.6), the program coordinates are transformed to the permanent BCS basis workpiece coordinate system.

Axis transformation

Axis transformation (see Section 5.4.2) is used to determine the transition between basis workpiece coordinate system and axis coordinate system. It describes the respectively valid kinematic relationship between the axes and the programmable coordinates.

In the most complex case, all P_i program coordinates have transformational relationships with regard to the A_i axis values.
The axis values are calculated from the programmed (Cartesian) **working range coordinates** with the aid of the axis transformation.
Calculation of the axis values is performed from the interpolated coordinate values within the axis transformation in the framework of cyclical interpolation.

The axis transformation considers the current tool dimensions (tool lengths), which are taken into account in the direction of the coordinates of the TCS tool coordinate system, so that the compensation will always be taken into account correctly, even in case of continuously changing **tool orientation** .

When the data specified for a coordinate is changed, this always results in only one traversing movement in a coordinate, because all geometric-kinematic relationships of machine and tool are taken into consideration in the axis transformation. Among others, the movement in one coordinate direction may result in a traversing movement of several axes due to the compensation movement.

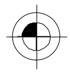
Without axis transformation, each coordinate is directly assigned to one axis. In this context, all coordinates are referred to as **pseudo coordinates**.

Axis zero shift

The axis zero shift takes into consideration the offset between ACS axis coordinate system and MCS machine coordinate system.

5.3.1 MCS machine coordinate system

The **MCS machine coordinate system** is defined by the entirety of all axes defined on one channel. The zero points of the coordinates are defined by the zero points of the position measuring systems of the axes. The machine position where all axes are in their zero position, is also referred to as machine zero point.

Zero-point designation and symbol: **machine zero point M** 

The **R reference point** represents the position within the machine coordinate system which results when all the axes are in their reference position.
The reference points of the individual axes (with cyclic absolute measuring systems) are defined axis by axis and refer to the respective axis zero position. In most cases, they are located in the marginal area of the axes and can be approached automatically.

Reference points are required to establish the dimensional reference between an axis equipped with an incremental measuring system and the respective axis zero position. After power-up or power failure, axes of this type first require an approaching the reference point operation.

Designation and symbol: **R reference point**

Axes with absolute measuring systems require no reference point because the measuring system supplies the axis positions directly after power-up.

5.3.2 ACS axis coordinate system

The **ACS$^{(1..4)}$ axis coordinate systems** are offset against the MCS by the value of the axis zero point shifts.
Since the ACS axis coordinate system is dependent on machine kinematics and machine type, it is not used for programming, as a rule.

5.3.3 BCS basis workpiece coordinate system

A **BCS basis workpiece coordinate system** is defined for each channel, which is mainly Cartesian and thus independent of the machine kinematics. The basis workpiece coordinate systems may be defined identically for various axis groups (channels).

Zero-point designation and symbol: **W workpiece zero point**

The zero point of the basis workpiece coordinate system is the first $W^{(0)}$ workpiece zero point in a series of n possible $W^{(n)}$ workpiece zero points , which may be created through zero point shifts (for details, please refer to Section 5.6 ff.)

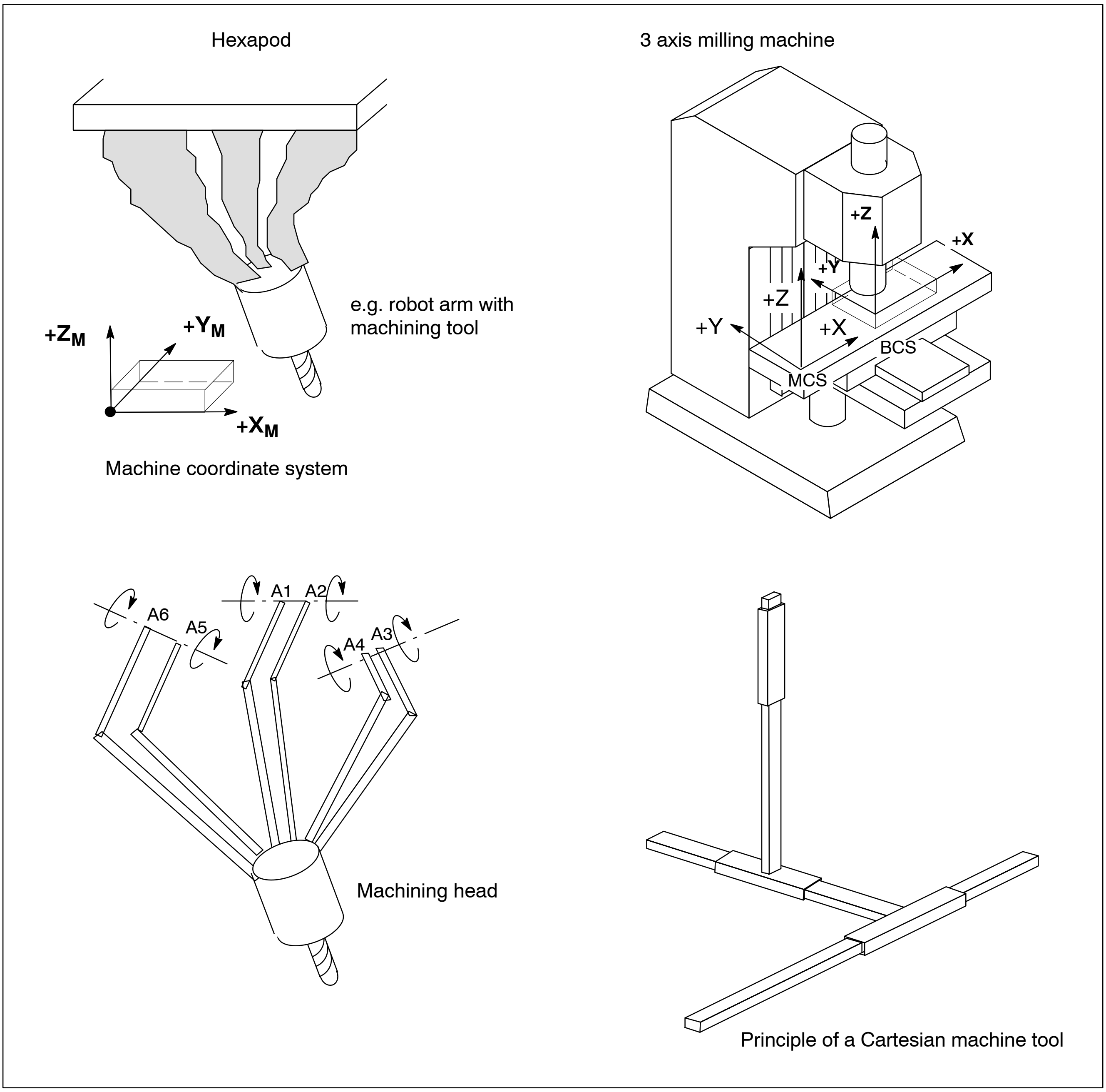
A relationship between the basis workpiece coordinate system and the related ACS axis coordinate system is created by the axis transformation which is also referred to as reverse transformation. In this axis transformation, the “axis zero position” of the involved machine axes is defined within the basis workpiece coordinate system via MACODA, in order to obtain the $W^{(0)}$ workpiece zero point.

If no axis transformation exists because the machine axes run parallel to the Cartesian coordinates, the $W^{(0)}$ workpiece zero point of the BCS corresponds to the zero point of the $ACS^{(1...4)}$ axis coordinate system with the highest index or the MCS machine coordinate system, if there is no axis zero shift.

In addition to the BCS basis workpiece coordinate system, there are other **WCS workpiece coordinate systems** which may be freely offset and rotated in relation to the BCS.
Several different $W^{(i)}$ (i=1...n) workpiece coordinate systems may be related to each other and/or act additively.
Whereas, for instance, a workpiece coordinate system describes a local $W^{(1)}$ coordinate zero point as a function of the absolute $W^{(0)}$ workpiece zero point , another workpiece coordinate system represents the relationship between 2 local workpiece zero points (e.g. $W^{(1)}$, $W^{(2)}$).

Coordinate systems of the control unit

Example of the relationship between the basis workpiece coordinate system and the related machine coordinate system:

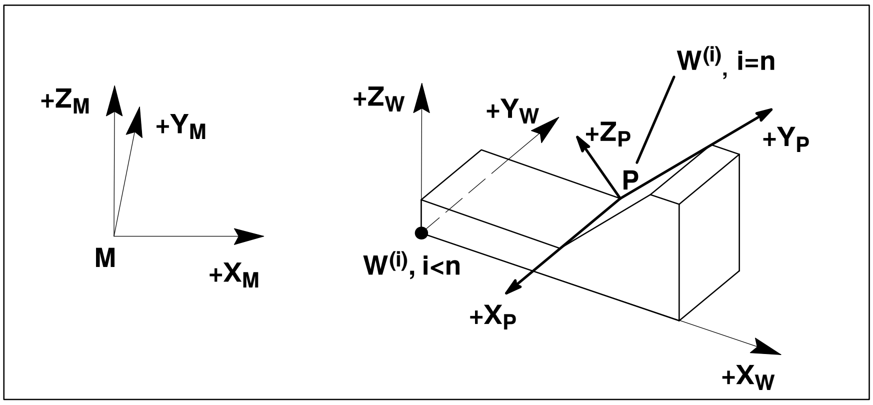


5.3.4 PCS program coordinate system

The nth workpiece coordinate system with the highest index ($W^{(i)}$, with i=n) in the active program is referred to as **PCS program coordinate system**.

Zero-point designation and symbol: **P program zero point**

The program coordinate system is the point which all programmed coordinate values of the program refer to. According to the definition of the workpiece coordinate systems, the program coordinate system may be freely offset and rotated in relation to any $W^{(i)}$ workpiece zero point .



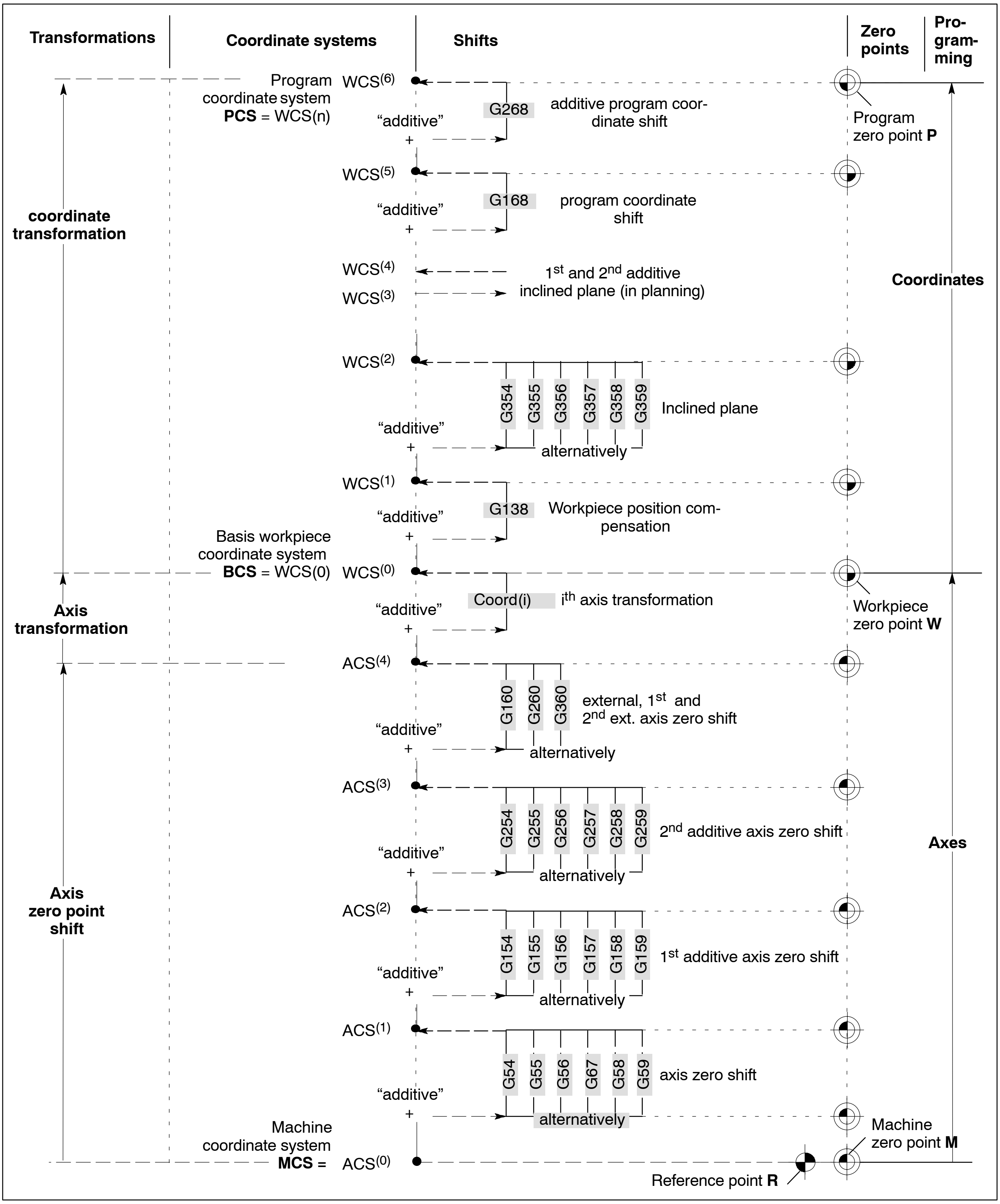
Relationship between the machine, workpiece and program coordinate systems

5.3.5 TCS tool coordinate system

The TCS tool coordinate system is an exception because its position and orientation is only calculated by the control unit implicitly with the aid of the tool compensation and the spatial tool position. This applies in particular to the 5 axis and 6 axis transformation. However, compensations of the TCS position can be carried out using special functions. For detailed information, please refer to Section 8.1.3 (Tool coordinate system with active axis transformation) as well as to Tool compensations on page 5–87 and 5–99.

5.4 NC functions for defining coordinate systems

An overview of the different coordinate systems and the functions for defining the coordinate systems are shown below:



The coordinate system at the top of the figure above is the PCS.

- Without “program coordinate shift, inclined plane and workpiece position compensation”, PCS = WCS$^{(0)}$ = BCS.
- Without “axis transformation” in addition, PCS = WCS$^{(0)}$ = BCS = ACS$^{(n)}$ = ACS.
- Without “axis zero shifts” in addition, all coordinate systems are identical. Then PCS = MCS.

5.4.1 Shifting the machine zero point (axis zero shift)

The machine zero point M is shifted through the following axis zero shifts.

- G53, G54 .. G59 ”Axis zero shift”
- G153, G154 .. G159 ”1^{st} additive axis zero shift”
- G253, G254 .. G259 ”2^{nd} additive axis zero shift”
- G160, G260, G360,G167 ”External axis zero shift”

5.4.2 Axis transformation: ACS <–> BCS

Activating the axis transformation by the command:

- Coord(i)

The number i (i=1,..,5) refers to one of the five axis transformation blocks in the MACODA (see MACODA Manual).

The programming of Coord initiates the following actions:

1. The axis transformation with number i is activated if the configuration in MACODA has no incorrect entries.
2. If the activated axis transformation supports the vector or tensor orientation, the corresponding orientation NC function is activated.
3. The programming of coordinate names is activated. Depending on the type of axis transformation, a certain subset of the maximum of six coordinates (e.g. x,y,z,phi,theta,psi) is programmable.
4. Axis positions are converted to coordinate values (the workpiece coordinate display jumps from axis positions to coordinate values).

Deselecting the axis transformation by the command:

- Coord(0).

The programming of Coord(0) initiates the following actions:

1. Active axis transformation is deactivated.
2. If an orientation NC function is active, it is deactivated.
3. The programming of coordinate names is deactivated. The axis names can then be used again without restrictions.

4. Coordinate values are converted to axis positions (the workpiece coordinate display jumps from coordinate values to axis positions). Changing directly from one axis transformation to another without previous deactivation is also possible.

☞ **The kinematic structures of machines may be very different. Therefore it is possible to realize other, customer-specific transformations in addition to the standard transformations for 5 axis and 6 axis machine tools that are stored in the control unit.**

5.4.3 Shifting the workpiece zero point

Basis workpiece coordinates

Absolute positional data normally refers to machine coordinates and thus to the M machine zero point . For practical purposes, all dimensions and paths to be traveled as stated in the part program refer to the W workpiece zero point or the P program zero point.
This way you ”unlink” your part programs from the machine-specific machine coordinates/machine zero point. Owing to the programmed shifts, you can execute every program at any point within the machine working range without having to adapt any dimensional data in the program.

All shifts and rotations have an additive effect to the basis workpiece coordinate system. Both positive and negative shift or rotation values are permitted. As long as no working range coordinates are active, the names of the logical axes of the channel are synonyms for the designation of the workpiece and program coordinate system.

☞ **If no shifts have been programmed, all part program values are interpreted as being machine coordinates!**

Workpiece coordinates

- The position of a workpiece can be corrected by spatially shifting the W workpiece zero point in X, Y and Z direction including a rotation of the workpiece coordinate system in the X/Y plane with the following instructions:
  - G138, G139 ”Workpiece position compensation”
- The W workpiece zero point can be shifted spatially and the workpiece coordinate system (WCS) rotated spatially by the following instructions:
  - G353, G354..G359 ”Inclined plane”
  - G453, G454..G459 ”1^{st} additive inclined plane” (in planning)
  - G553, G554..G559 ”2^{nd} additive inclined plane” (in planning).

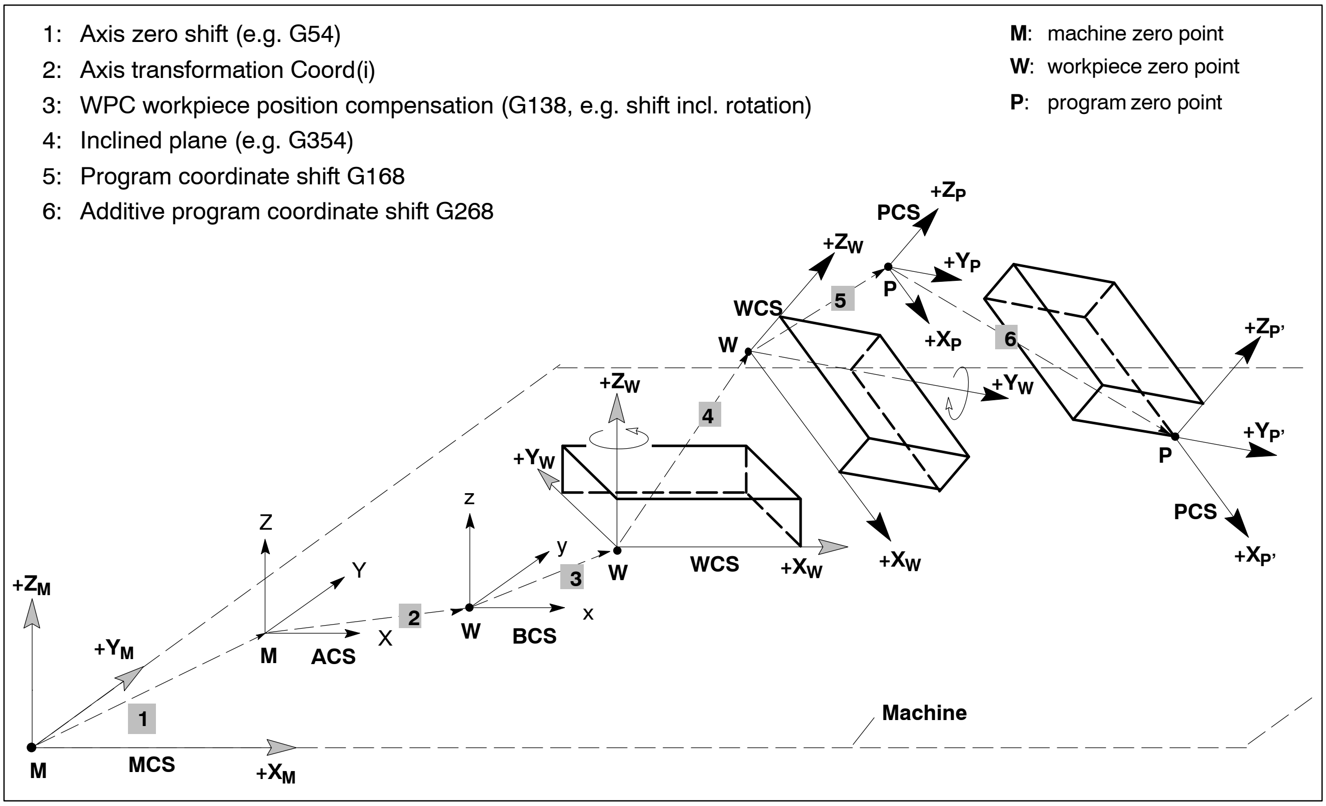
NC functions for defining coordinate systems

Program coordinates

The last coordinate system in the series of workpiece coordinate systems is also referred to as program coordinate system (PCS) (see also Section 5.3.4).

- When shifting the P **program zero point**, you can shift the position of the PCS **program coordinate system** in relation to the WCS(i) workpiece coordinate system last active by using the following instructions:
  - G169, G168 ”Program coordinate shift”
  - G269, G268 “Additive program coordinate shift”

Example: Effects of the individual functions



☞ **You should position the workpiece zero point in such a manner that the dimensions of the production drawing can be easily translated into coordinates or, resp., imported.**

Activating shift values

Depending on the individual G-functions, the values of the various kinds of shifts are activated in different ways:

- **Axis zero shift tables**":
  "Axis zero shift", "1st additive axis zero shift" and "2nd additive axis zero shift"

  The **axis zero shift table** is a machine zero point shift and constitutes the offset between MCS and ACS, i.e. the individual coordinates of the ACS axis coordinate systems are shifted by the axis zero shift in relation to those of the MCS.

  If **no axis transformation** is active, you use the axis zero shift tables to store the "distance" between the "M" machine zero point and the "P" program zero point or "W" workpiece zero point for each axis (for determination of the "distance", please see the following figure). If the corresponding axis zero shift is then activated, the control unit will automatically add the stored "distance" to each programmed absolute axis position.

  Axis zero shift tables in the form of ASCII files are available in the file system. They are activated channel-specifically using G22.

☞ **For details on how to edit these tables, please refer to the operating instructions.**

- "**External axis zero shift**
  The shift values are set for each axis **by the PLC**. For further details regarding programming, please refer to G160, G167, G260 and G360.

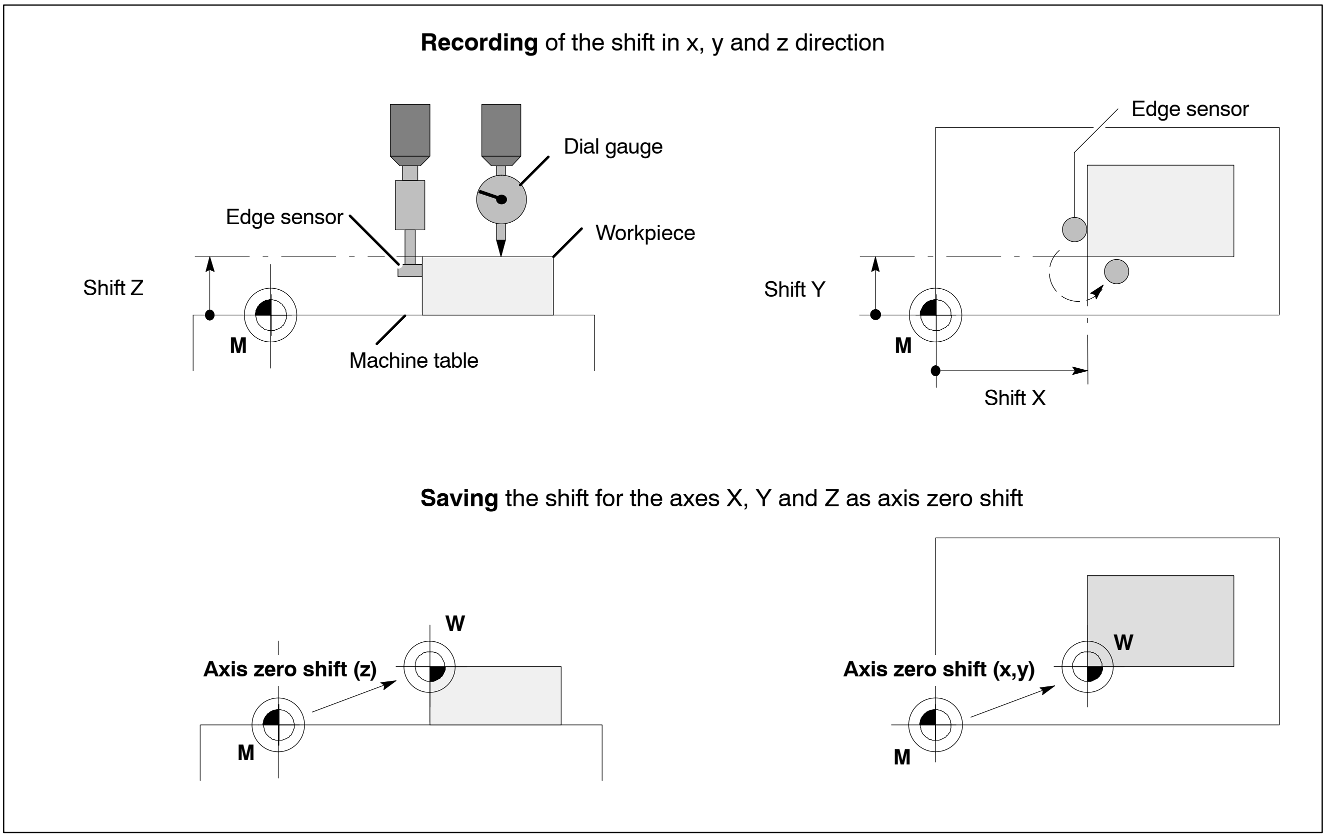
- "**Workpiece position compensation (WPC)**"
  The shift values for individual logical axes and the rotation of the X/Y plane are programmed **directly in the path command of the G138 G instruction**. Recording of the shift value e.g. via edge sensor. For further details, please refer to "G138".

- **"Inclined plane":**
  The shift values for individual **logical** axes and the unrestricted three-dimensional rotation are stored in an ASCII file with a freely definable name, which is addressed using the ID syntax. The values are activated by G22. Data for spatial alignment e.g. via programming systems. For further details, please refer to "G352-G359".

- **”Shifting the program zero point”:**
  The shift values for individual **logical** axes (on the X/Y, X/Z, Y/Z planes) are programmed **directly in the path command of G168 G instruction and/or G268**. For further details, please refer to ”G168 and G268”.

Programming and activating axis zero shifts is described under the G53-G59, G138, G153-G159, G160, G253-G259 G instructions. Axis zero shifts can also be activated using the ”Inclined plane” functions G352-G359 (and, in future, also using G452-G459, G552-G559).

Example: Axis zero shift recording



First, the required zero shift values are determined when setting up the machine (see next figure). The shift values are independent of the tool lengths:

- in the X and Y direction using edge or centering sensors
- in the Z direction using a stop gauge, dial gauge with tripod, a scribing block or a barrel gauge.

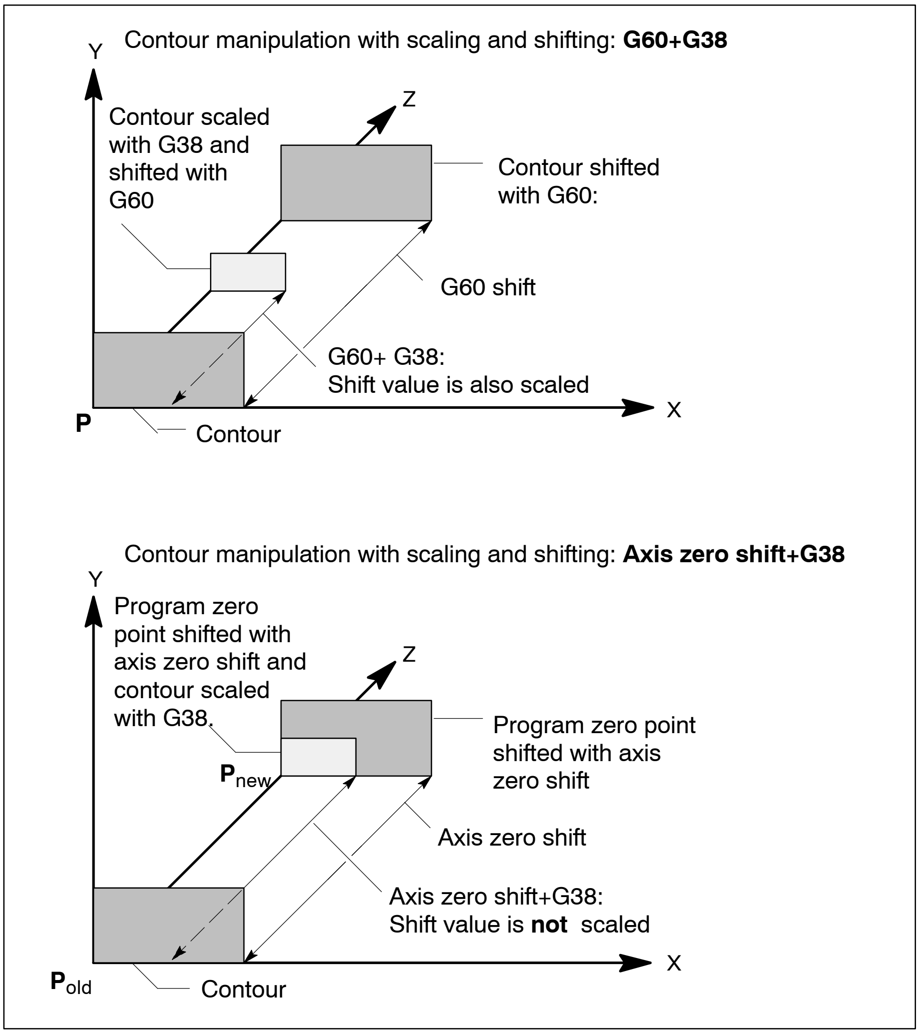
Subsequently, you enter the values so determined in the axis zero shift table selected.

5.4.4 Functions for manipulating the programmed contour

Within the programming system, the control unit is able to manipulate a programmed contour as follows:

- **shift** (G60 – programmed contour shift) and/or
- **mirror, scale** and **rotate** the contour around an axis parallel to the coordinates of the program coordinate system (G37, G38).

☞ **Please note: using the G37, G38 and G60 instructions, you cannot influence the program coordinate system itself, but only change the position, orientation and scaling of the programmed contour in relation to the program coordinate system.**
In contrast to the other zero shifts, a G60 shift will also be influenced (e.g. scaled: see fig. below) by G37 and G38. For further details, please refer to the corresponding functions.



5.5 Eulerian angle

Orientations for coordinate systems and tools may be specified with the aid of Eulerian angles.
Thus the main axes of a tool, laser, gripping device and similar are aligned in a specified direction in space. In this context, it makes no difference whether the orientation of the tool axis runs parallel with one of the main directions of the machine axes or not.

Up to three independent orientation coordinates, the so-called Eulerian angles φ (phi), ϑ (theta) und ψ (psi) can be used to describe an orientation in space. They refer to the PCS program coordinate system.

If only 2 angles (φ (phi), ϑ (theta)) are specified (e.g. 5 axis transformation), these angles are also referred to as **polar coordinates**.

The Eulerian angles are required for the following NC functions:

- **Inclined plane**
  This function can be used to freely orient the WCS workpiece coordinate system in space with regard to the BCS basis workpiece coordinate system with the aid of the 3 Eulerian angles (for details, see page 5–42 ff.).
- **6 axis transformation**
  The TCS tool coordinate system permanently coupled with a non-rotation symmetrical tool is freely oriented in space with regard to the PCS program coordinate system with the aid of the 3 Eulerian angles (see 6 axis transformation, page 5–90 and tensor orientation, page 5–59).
- **5 axis transformation** The orientation vector permanently coupled with a rotation symmetrical tool is freely oriented in space with regard to the PCS program coordinate system with the aid of the φ (phi), ϑ (theta) polar coordinate angles (see 5 axis transformation, page 5–80 and vector orientation, page 5–56).

The following definition intervals are applicable to Eulerian angles φ, ϑ and ψ:

φ (phi): $0° \leqslant \varphi < 360°$
ϑ (theta): $0° \leqslant \vartheta \leqslant 180°$
ψ (psi): $0° \leqslant \psi < 360°$

The effect of the restricted definition areas is that for any orientation of a target coordinate system there is a unique value tupel for the Eulerian angles relative to the initial coordinate system.

☞ **The Eulerian angles are interpreted in different ways by programming systems. For instance, the Eulerian angles may refer to other axes around which a rotation is taking place.**
Therefore make sure that the programming system used supports Eulerian angles in accordance with the PNC programming instructions!

Eulerian angle

Options for Eulerian angle orientation using the angles phi, theta and psi

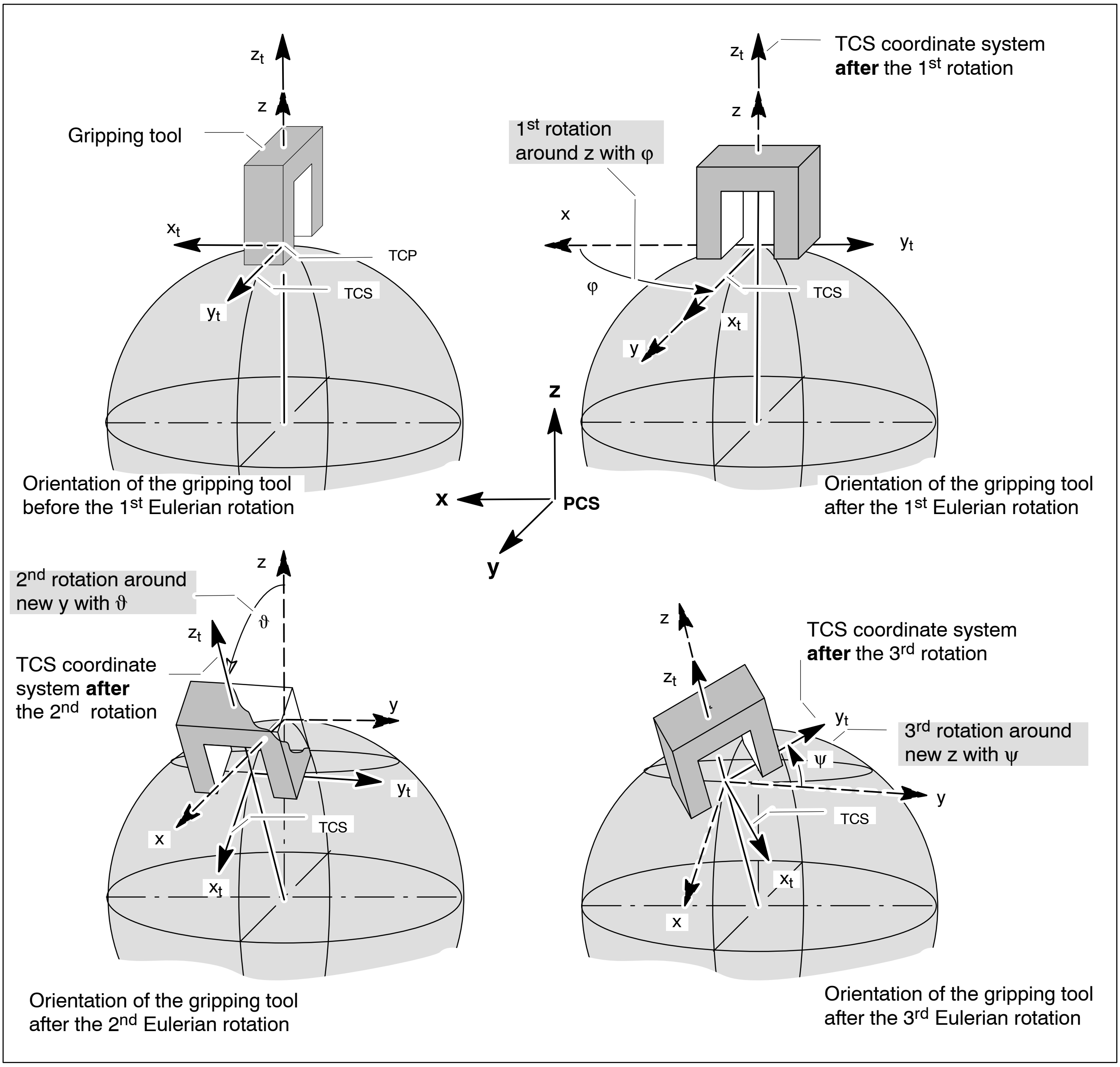
There are **3 options** for the orientation of the tool coordinate system (TCS) using Eulerian angles, where the result of the orientation is always the same. Each of these options provides for the initial coordinate system to be transferred to the target coordinate system by 3 consecutive basis rotations.

Option 1 Rotation around each new coordinate

1. Rotation around the Z coordinate of the initial coordinate system with angle φ (**phi)**.
   This results in the 1st intermediate coordinate system.
2. Rotation around the Y coordinate of the 1st intermediate coordinate system with angle ϑ **(theta)**.
   This determines the 2nd intermediate coordinate system.
3. Rotation around the new Z coordinate of the 2nd intermediate coordinate system with angle ψ **(psi)**.
   Thus the target coordinate system is reached.

Example:
Gripping tool orientation (in case of a 6 axis transformation). For better clarity, the tool coordinate system is shifted to the North pole of the program coordinate system.

Eulerian angle



In this option, the X-Y main plane is already defined by the first two basis rotations. The 3rd basis rotation merely serves to define the X–Y coordinates within the target coordinate system.

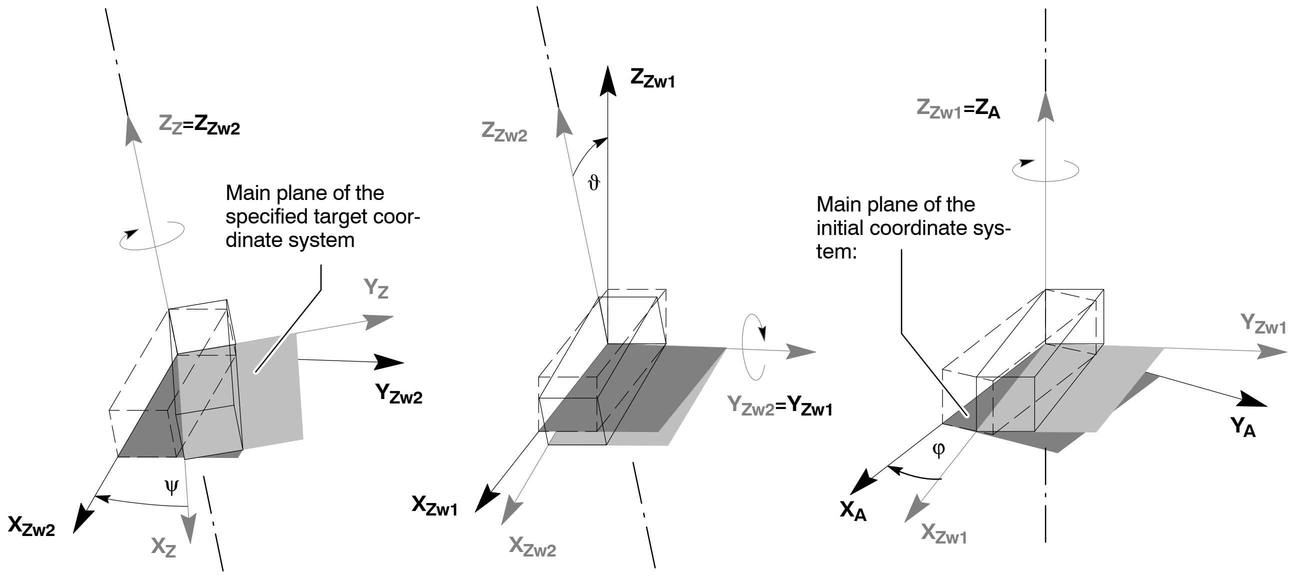
Determining the Eulerian angles for the target coordinate system

For a **specified** target coordinate system, the 3 Eulerian angles with regard to an initial coordinate system are determined in a simple manner:

1. Rotating the target coordinate system in negative sense of rotation around its Z axis until its Y axis is aligned parallel to the X–Y plane of the initial coordinate system. The value of the angle of rotation is the Eulerian angle ψ **(psi)**.

Eulerian angle

2. Subsequently, rotating in negative sense of rotation around the new Y axis until the X axis is aligned parallel with the X–Y plane of the initial coordinate system or the Z axis is parallel to the Z axis of the initial coordinate system. The value of the angle of rotation is the Eulerian angle ϑ **(theta)**.
3. Finally, rotating in negative sense of rotation around the Z axis, until both coordinate systems are in parallel with each other. The value of the angle of rotation is the Eulerian angle φ (**phi)**.
   Thus the target coordinate system is reached.



Basis rotation around psi

Basis rotation around theta

Basis rotation around phi

Main plane before a basis rotation

Main plane after a basis rotation

Zw_x = x^{th} intermediate coordinate system

Eulerian angle

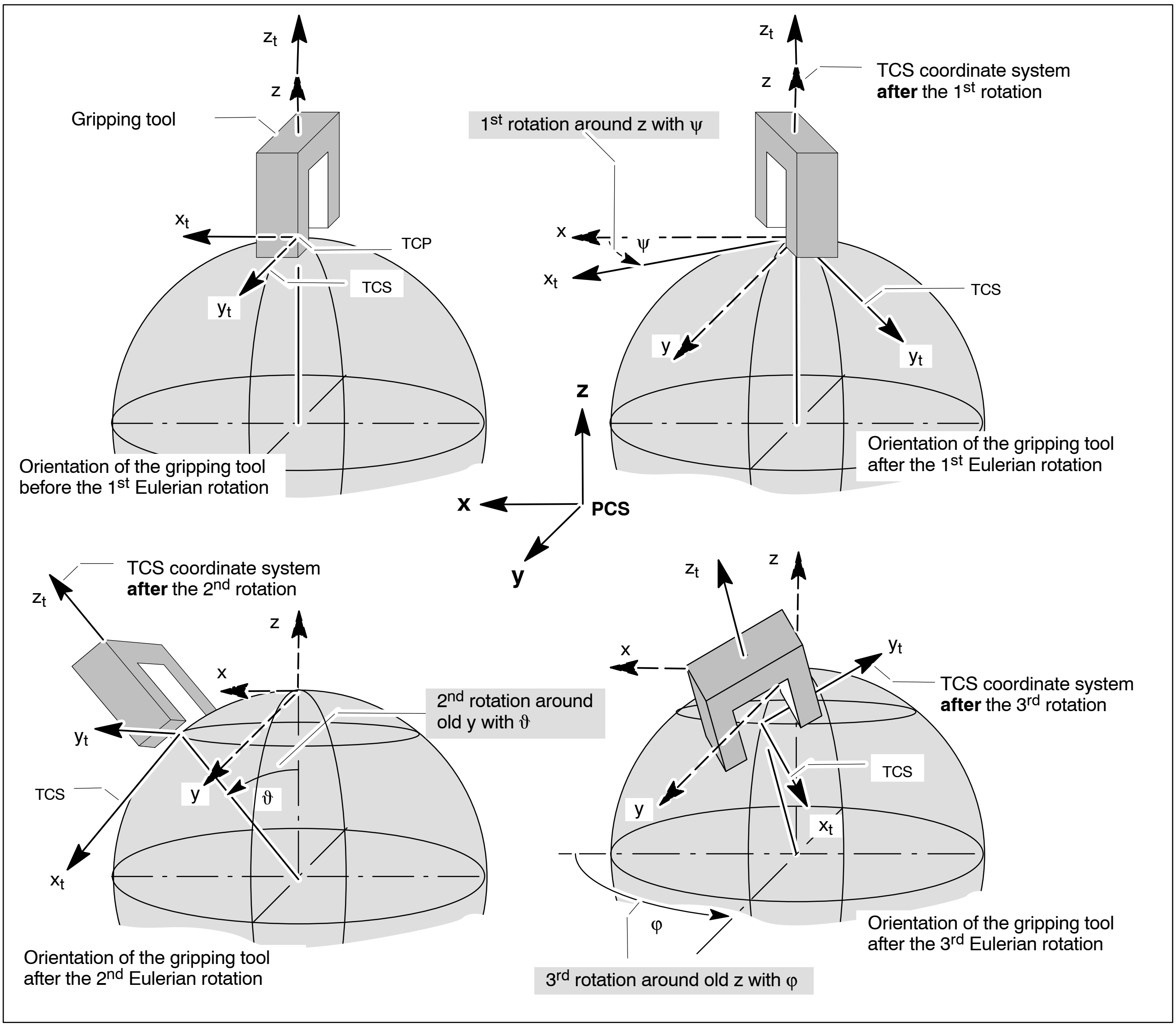
Option 2 Rotation around coordinates fixed in space

1. Rotation around the Z coordinate of the initial coordinate system with angle ψ **(psi)**.
2. Rotation around the old Y coordinate of the initial coordinate system with angle ϑ **(theta)**.
3. Rotation around the old Z coordinate of the initial coordinate system with angle φ (**phi)**.
   Thus the target coordinate system is reached.

In this option, the Eulerian angles can be read immediately:

ϑ (theta): degree of latitude of z_t seen from the North pole
φ (phi): degree of longitude of z_t
ψ (psi): deviation of x_t from the southern direction

Example: Gripping tool orientation (in case of a 6 axis transformation). For better clarity, the tool coordinate system is shifted to the North pole of the program coordinate system.



Eulerian angle

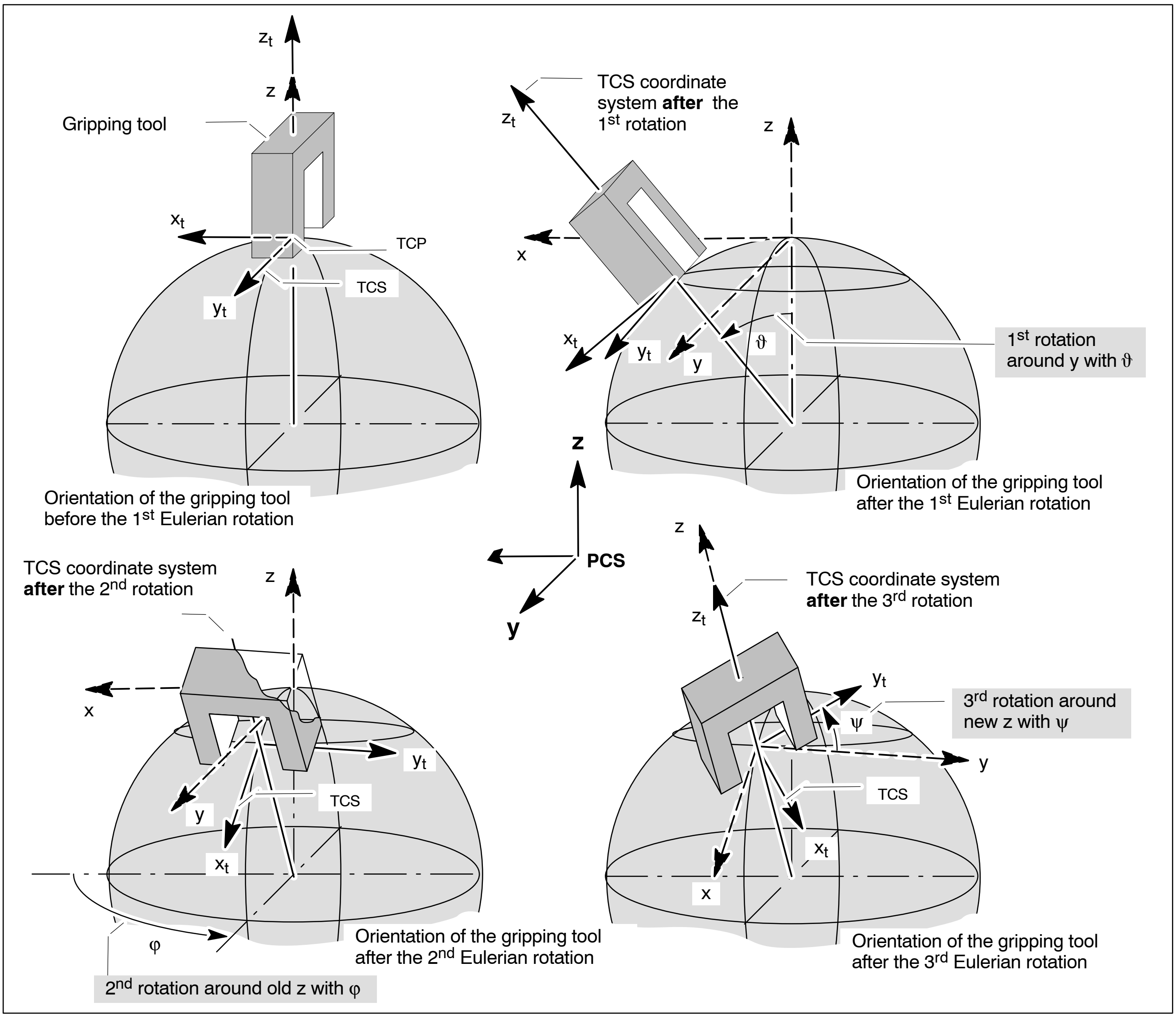
Option 3 Rotation around each new coordinate fixed in space

1. Rotation around the Y coordinate of the initial coordinate system with angle ϑ **(theta)**.
2. Rotation around the old Z coordinate (fixed in space) of the initial coordinate system with angle φ (**phi)**.
   This determines the 1st intermediate coordinate system.
3. Rotation around the new Z coordinate of the "TCS coordinate system after the 2nd rotation" with angle ψ **(psi)**.
   Thus the target coordinate system is reached.

In this option, the Eulerian angles can be read immediately:

ϑ (theta): degree of latitude of z_t seen from the North pole
φ (phi): degree of longitude of z_t
ψ (psi): deviation of x_t from the southern direction

Example: Gripping tool orientation (in case of a 6 axis transformation). For better clarity, the tool coordinate system is shifted to the North pole of the program coordinate system.



5.6 Shifts

Program-controlled shifts are used to shift an existing reference coordinate system to another location, which results in a new reference coordinate system. In the process, the old zero point is offset by the shift value within the control unit. The workpiece-related display of the control unit now shows the momentary position in relation to the new zero point.
Although the displayed axis position has changed, none of axes has been physically moved up to this point. From the point of view of the control unit, the axes are only located in a different position **in relation to the new workpiece zero point**.

Shift values may be provided via:

- an axis zero shift table (axis ZS table)
- a PLC as ”external” axis shift values
- special shift functions and, if applicable, axis by axis programming of required data.

☞ **Some coordinate transformations additionally allow for a rotation of the coordinate system.**

Depending on the group assignment, shifts may have different effects:

- shifts from different groups are added
- shifts within one group overwrite each other.

Which of the cases is applicable depends on which groups the individual G functions, used to activate a shift/rotation, are assigned to.
For information on groups, see DIN Programming Manual, overview of the G instructions.

The relationship between shift, group assignment and how the shift values are made available is shown in the table below:

Shifts

Shift	G function for activation and group assignment	Made available by
Axis zero shifts:		
• axis zero shift	G54...G59 (group 17)	Axis ZS table
• 1st additive axis zero shift	G154...G159 (group 18)	Axis ZS table
• 2nd additive axis zero shift	G254...G259 (group 19)	Axis ZS table
• external axis zero shift	G160, G260, G360 (group 24)	PLC
Workpiece position compensation	G138 (group 23)	G138
Program coordinate shifts:		
• PCS	G168 (group 46)	G168
• additive PCS	G268 (group 47)	G268
Inclined plane:		
• Inclined plane	G352..G359 (group 26)	ID table
• 1st additive inclined plane	G452..G459 (group 26)	in planning
• 2nd additive inclined plane	G552..G559 (group 26)	in planning

For details on the structure of axis zero shift tables, see page 5–34 ff.
For details on the structure of inclined plane tables, see page 5–45 ff.

Special cases

- **Zerosetting of modulo axis (linear endless axis) G105**

  G105 determines the programming zero point. Using this point the control unit calculates the distance from the zero point of the command-value system. The resulting offset is internally added to all subsequent values.

  Using the G105 ”Modulo axis zerosetting” function, the point of reference (programming zero point) of a linear endless axis can be determined (see page 4–5 ff.). As soon as the modulo value is reached, the actual value of the linear endless axis is automatically set to zero. This modulo calculation prevents an overflow of the axis values and enables the axis to travel at ”endless”.

- **Setting actual value G92**

  G92 re-determines the actual value.

  Programming G92 **without** axis information: the current actual value of all axes is set to the machine coordinates without taking into account compensations and zero shifts.

  Programming G92 **with** axis information: the current actual value of an axis is set to the programmed value.

5.6.1 Axis zero shifts (Gx53-Gx59)

Function:

Using the axis zero shifts (axis ZS, 1st additive axis ZS, 2nd additive axis ZS), the machine zero point is shifted to a new position in relation to the axis. The reference system is the machine coordinate system (MCS). If no other shift from the coordinate transformations and no other axis transformation is active when the external axis ZS is called up, the shifted zero point can also be considered as “workpiece zero point”.

The shift values of each individual axis are stored in axis zero shift tables.
Each axis zero shift table contains 3 groups with 6 axis zero shifts per axis (G54..G59; G154..G159; G254..G259).

☞ **For details on naming conventions, activation and structure of axis zero shift tables, see page 5–34 ff.**

To activate the shift, you first activate the desired axis ZS table using G22. Subsequently, you select the shift values of the respective axes using the “G54..G259” axis ZS functions. The structure of the axis zero shift tables has to be so that the shift values of all relevant axes are available when an axis ZS function is called.

Example:
axis zero point shifts of table “npvtab.npv” active only!

Shifts

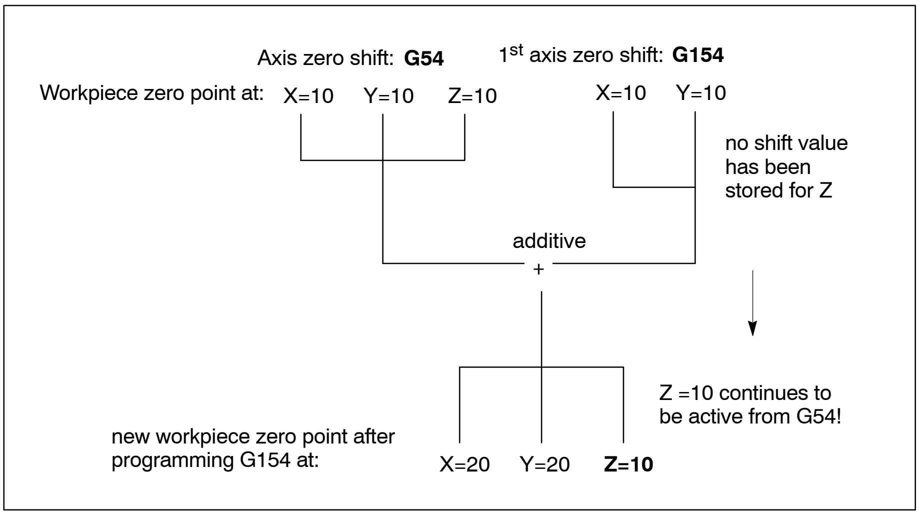
Program	Machine zero point	Machine zero point in relation to workpiece coordinates	Workpiece zero point	Workpiece zero point in relation to machine coordinates
N10 G22 V npvtab.npv	X=0 , Y=0 , Z=0			
N20 G0 X0 Y0 Z100				
N30 G54		X=–5 , Y=–5 , Z=–10	X=0 , Y=0 , Z=0	X=5 , Y=5 , Z=10
N40 X0 Y0 Z0 G1 F500				
N50 X5				
N60 Y5				
N70 X0				
N80 Y0				
N85 Z 100				
N90 G154		X=–35 , Y=–40 , Z=–12	X=0 , Y=0 , Z=0	X=35 , Y=40 , Z=12
N100 X0 Y0 Z0				
N150 X5				
N160 Y5				
N170 X0				
N180 Y0				

Calling G54

Calling G154

	Axis zero shift table npvtab.npv		
	Axes		
Axis ZS	X	Y	Z
G54	5.00	5.00	10.00
G55	55.01	55.02	55.03
G56	56.01	56.02	56.03
G57	57.01	57.02	57.03
G58	58.01	58.02	58.03
G59	59.01	59.02	59.03
G154	30.00	35.00	2.00
G155	155.01	155.02	155.03
G156	156.01	156.02	156.03
G157	157.01	157.02	157.03
G158	158.01	158.02	158.03
G159	159.01	159.02	159.03
G254	254.01	254.02	254.03
G255	255.01	255.02	255.03
G256	256.01	256.02	256.03
G257	257.01	257.02	257.03
G258	258.01	258.02	258.03
G259	259.01	259.02	259.03

Activating an axis zero shift does not necessarily mean that shift values are applied to all existing axes; therefore, the following special case may occur when **additive axis zero shifts** are used:



Restrictions:

- The axis zero shifts of a group have a modal effect and deselect each other mutually within their own group. This applies to:
  – G54...G59 Axis zero shift
  – G154...G159 1st additive axis zero shift
  – G254...G259 2nd additive axis zero shift
  – G160, G260 G360 External axis zero shift
- Active axis zero shifts from different groups have an additive effect.
- Other shift types have an additive effect on axis zero shifts. This includes the following functions:
  – G138 Workpiece position compensation
  – G168 and G268 Program coordinate shift
  – G352...G359 Inclined plane
  (– G60 Programmed contour shift)
- If G138 is programmed following an active axis ZS, the coordinate system shifted by the axis ZS is offset by the programmed shift value of G138 and subsequently rotated as specified.
- Shift values for the axis ZS and the 1st/2nd additive axis ZS are provided in the axis ZS tables. An axis ZS table contains additional information on
  - axis names and assignment to table columns
  - channel assignment
  - unit of the stored shift values (see also MP 9020 00010)

☞ **Older axis ZS tables (prior to SW version 4.3.x) do not include this information, so that their operability has to be checked.**

- Axis ZS values are not influenced by the functions rotating, mirroring, scaling (G38).
- To approach a fixed machine axis position (G76; absolute position, in relation to the machine coordinate system), the control unit will not take into account the active axis zero shifts (G54..G259).

Relevant G functions:

G22 V <*Path/table name*>:
activating axis zero shift table via NC block.
G54...G59: activating an "internal" axis zero shift.
G154...G159: activating an "internal" 1st additive axis zero shift.
G254...G259: activating an "internal" 2nd additive axis zero shift.
G53: **all** active "internal" axis zero shifts off.
G153: 1st additive axis zero shift off.
G253: 2nd additive axis zero shift off.

Relevant MACODA parameters (MP):

9020 00010: Specifies the unit (mm or inch) used to internally store the table values.

Shifts

Relevant CPL functions:

FXCR(..)	Creating new axis ZS table.
FXINS(..)	Inserting column in axis ZS table.
FXDEL(..)	Deleting column in axis ZS table.
FXC(..):	Access to axis zero shift values.
MCODS(14)	Name of the active axis ZS table (Motion Control data services of the NCS via CPL).
MCODS(15)	Active axis ZS values (Motion Control data services of the NCS via CPL).
CPOS(...)	Last programmed absolute position of a *synchronous* axis (machining axis) in relation to the **program zero point** active at the time of block preparation.
WPOS(...)	Interpolated workpiece setpoint position (in relation to the current **workpiece zero point**) of an axis that is current at the time of CPL block preparation (in which WPOS is programmed).
MPOS(...)	Interpolated machine setpoint position (in relation to the current **machine zero point**) of an axis that is current at the time of CPL block preparation (in which MPOS is programmed).
PPOS(...)	If touch-trigger switching probes have been connected, the function will transmit the current axis actual position to a synchronous axis (machining axis) in the switching point of the probe without taking into account all active axis zero shifts.

5.6.2 Structure of axis zero shift tables

If you use the axis zero shifts (axis ZS, 1st additive axis ZS, 2nd additive axis ZS; see page 5–28 ff.), the shift values will be provided in an "axis zero shift table" (axis ZS table).

It is possible to store 18 shifts per applied axis in one axis ZS table. The control unit saves these tables as ASCII files with an accurately defined structure.

The table editor is available (see PNC operating instructions) for comfortable editing of these tables at the control unit.
The table editor starts automatically when an axis ZS table has been selected in the operating area MANAGE and you press the EDIT softkey (for information on the required preconditions, please refer to MP 3080 00200).
If the control unit loads the selected table into the NC editor (ASCII editor), the required preconditions have not been fulfilled. In these cases, we recommend that you close the editor without changing the table and edit the table manually via MANAGE ▶ TABLES and subsequent path/file selection via the table editor.

☞ **Program block -B24FLTAB (access to geometry compensation or axis zero shift tables) is no longer supported from software version 5.1.18 on!**

Table name

The table name is composed of a freely selected file name and a file name extension (e.g. npvtab.npv).
The file name extension can be changed, for details, see MP 3080 00200 on page 5–37.

Structure

An axis ZS table is structured as follows:

- 2 headlines containing an internal system coding of necessary axis information as well as the relevant logical or physical axis names.
- 18 lines for shift values.
  Each line starts with the G function designation and, after the semicolon, the corresponding shift values of the axes involved (see example on the next page).
  Both positive and negative shift values are permitted.

Shifts

Example: axis zero shifts G54 .. G259 for 5 axes

[5005 0 0 0 24]						
		X	**Y**	**Z**	**A**	**B**
G54	:	54.01	54.02	54.03	54.04	54.05
G55	:	55.01	55.02	55.03	55.04	55.05
G56	:	56.01	56.02	56.03	56.04	56.05
G57	:	57.01	57.02	57.03	57.04	57.05
G58	:	58.01	58.02	58.03	58.04	58.05
G59	:	59.01	59.02	59.03	59.04	59.05
G154	:	154.01	154.02	154.03	154.04	154.05
G155	:	155.01	155.02	155.03	155.04	155.05
G156	:	156.01	156.02	156.03	156.04	156.05
G157	:	157.01	157.02	157.03	157.04	157.05
G158	:	158.01	158.02	158.03	158.04	158.05
G159	:	159.01	159.02	159.03	159.04	159.05
G254	:	254.01	254.02	254.03	254.04	254.05
G255	:	255.01	255.02	255.03	255.04	255.05
G256	:	256.01	256.02	256.03	256.04	256.05
G257	:	257.01	257.02	257.03	257.04	257.05
G258	:	258.01	258.02	258.03	258.04	258.05
G259	:	259.01	259.02	259.03	259.04	259.05

☞ **If no shift is required for an axis, all lines in the corresponding columns have to contain the value zero (e.g. 0.000) nevertheless. The specified format of the table has to be maintained. Blanks may be used between the columns for this purpose.**
Otherwise, an error message will be output.

Structure at axis transfer

The structure of a newly created table can take into account:

- the channel configuration or
- all system axes or
- a random table as template.

Criteria for creating a table:

- In case of “channel configuration” the table contains the preset axis assignment of a channel

- In case of "all system axes" the table contains columns for **all possible axes** in the system with physical axis names. Only the columns of the synchronous axes of the channel are used for the interpretation of the table.
  The table columns of non-configured axes or axes for which no column exists, are ignored. Entering an additional identification in the table may result in a runtime error instead.
- When an existing table is taken over, the axis names and unit of measure system of this table are adopted.

Example:
In the figure below, the logical axes X, Y, YA and YB are active on the channel.
Having activated table Tab1 in block N210, the 1st column is assigned to axis X, the 2nd column to axis Y and the 4th column to axis YA.
No zero shift is active for axis YB. Column 3 of rotary axis B is not used.

Tab1	X	Y	B°	YA
G54	54.01	54.02	54.04	54.05
G55	55.01	55.02	55.04	55.05
...				

```
N100 X0 Y0 YA0 YB0
N210 G22 V Tab1
N220 ...
```

The following functions can be used to change a table via table editor, should this be required due to a modified axis assignment:

- Delete columns
- Insert columns:
  An axis name must be specified when a column is inserted. In case of a logical axis name, it is possible to choose between rotary axis (degrees) and linear axis (Inch/metric). In case of physical axis name, the unit of measure system has been established.
  When the table is evaluated, the logical axis names have priority over the physical axis names. If the logical name is not available, the column with the physical name will be used.
- Change columns:
  The axis names of the columns may be changed. A logical axis name can turn into a physical axis name and vice versa. The unit of measure system can be specified additionally in case of logical axis names.
  Columns can be exchanged in the table editor. This has no effect on the interpretation of the table.
- Compulsory strict assignment:
  If this option has been set for the table, the axis group in the respective program section has to coincide exactly with the axis group in the table, otherwise there will be a runtime error.

Shifts

Relevant G functions:

G22 V <*Path/table name*>:
activating axis zero shift table via NC block.

Relevant MACODA parameters (MP):

3080 00200: Determines in individual parameter 1 which file name extension the control unit uses as identification criterion for axis zero shift tables (standard: “npv”).
Selected files with this file name extension are automatically opened with the table editor (and not with the ASCII editor) for editing.
Please note that the entry is case sensitive: ”NPVTAB.NPV” and ”npvtab.npv” designate different files for the control unit!

3080 00001: Search path.

9020 00010: Specifies in which unit of measure the table values are stored internally and how the control unit is supposed to interpret the axis zero shift values transmitted by the PLC (in case of external axis zero shifts).

☞ **Changes in MP 902000010 also influence geometry compensation tables, tables for the ”inclined plane” and the interpretation of geometry compensation values transmitted by the PLC!**

Relevant CPL functions:

FXC(..) Read and write access to axis zero shift values of axis ZS tables and to the external axis zero shift values stored internally in the CNC.

FXCR(..) Creating new axis ZS table.

FXINS(..) Inserting new columns in axis ZS table.

FXDEL(..) Deleting columns from an axis ZS table.

5.6.3 Activating axis ZS table

To activate an axis ZS table via an NC block, use G22 with address V:

Syntax:

`G22 V` <*Path*><*Table name*>

☞ **If <*Path*> has not been specified, the PNC uses the search path (MP 3080 00001) to search for the <*Table name*>.**

Examples:

Activating the axis zero shift table ”npvtab.npv” in the directory /**database** of the control unit:

`G22 V npvtab.npv`

Activating the axis zero shift table ”npvtab1.npv” in the **root directory** of the control unit:

`G22 V /npvtab1.npv`

Activating the axis zero shift table ”npvtab2.npv” in the directory /**mnt** (mounted directory on an NC operating panel):

`G22 V /mnt/npvtab2.npv`

To activate the shift values, proceed as follows:

1. First activate the desired axis ZS table as described above.
2. Activate the desired axis zero shift using the respective G function (G54..G259).

To switch off active axis zero shifts, program G53, G153 or G253, respectively.

5.6.4 External axis zero shift (G160, G260, G360)

Function:

Using the "external axis zero shifts" (G160 = 1st external axis ZS, G260 = 2nd external axis ZS, G360 = 3rd external axis ZS), the machine zero point is shifted to a new position in relation to the axis. The reference system is the machine coordinate system (MCS).
If no other shift from the coordinate transformations and no other axis transformation is active when the external axis ZS is called up, the shifted zero point can also be considered as workpiece zero point.

As opposed to the "axis zero shift" (see page 5–28 ff.), however, the shift values are not provided via axis zero shift table but transferred automatically from the PLC to the CNC (required program module –B23XTNPV).

One out of a maximum of 3 external axis zero shifts (G160, G260, G360) can be executed per applied machining axis (=synchronous axis).
Following the transfer of the relevant shift values, the desired axis zero shift is activated via G160, G260 or G360.

Restrictions:

- The external zero shifts act modally and deselect each other mutually. This applies to:
  – G160, G260 G360 External axis zero shifts
- Active axis zero shifts from different groups have an additive effect.
- Other shifts have an additive effect on external axis zero shifts. This includes the following functions:
  – G138 Workpiece position compensation
  – G168 and G268 Program coordinate shift
  – G352...G359 Inclined plane
- If G138 is programmed following an active axis ZS, the coordinate system shifted by the axis ZS is offset by the programmed shift value of G138 and subsequently rotated as specified.
- Shift values for the external axis ZS are provided by the PLC (required program module –B23XTNPV). They are stored internally in the control unit, only zeroset following a control unit start-up and can be changed by the PLC as well as via CPL.
  Control reset does not influence the internal shift values.
- If the PLC changes the currently active external shift values while executing a part program, this change will only become active in the **block under preparation as the next** block.
  Under certain circumstances this can mean that even more blocks are due to execution without this change. In order to avoid this effect you must program the "WAIT" CPL command directly after the block, causing the PLC to hand over the new shift values. By doing so you hold the block preparation of the PNC until all program blocks ahead of "WAIT" have been executed.
  Subsequently, in the program block following "WAIT", the new values will already be active.

- External zero shifts can be specified for logical axes or for system axes.
  Since the sequence of the logical axes can change in programs with axis transfer, it is recommended to specify the zero shift for system axes.
  Otherwise, it must be ensured that the sequence in which logical axes are entered in the PLC corresponds to the sequence on the channel. The sequence on the channel can be determined at the bit interface.

☞ **In programs with axis transfer, it is recommended to specify the zero shifts for system axes.**

Relevant MACODA parameters (MP):

902000010: Defines in which unit of measure the shift values for the external axis ZS are specified by the PLC.

Relevant G functions:

G160: Activating 1st external axis zero shift.

G260: Activating 2nd external axis zero shift

G360: Activating 3rd external axis zero shift

G167: All external axis zero shifts off

Relevant interface signals:

NC-O19.4 through

NC-O19.5: “External axis zero shift 2^0...2^1“ (channel interface). The signals show whether and which external axis zero shift (G167, G160, G260 or G360) on the channel is active.

Relevant CPL functions:

FXC(..): Access to external axis zero shift values.

MCODS(52) Provides the active external axis ZS values (Motion Control data services of the NCS via CPL).

CPOS(...) Last programmed absolute position of a *synchronous* axis (machining axis) in relation to the **program zero point** active at the time of block preparation.

WPOS(...) Hands over the interpolated workpiece setpoint position (in relation to the current **workpiece zero point**) of an axis that is current at the time of CPL block preparation (in which WPOS is programmed).

MPOS(...) Hands over the interpolated machine setpoint position (in relation to the current **machine zero point**) of an axis that is current at the time of CPL block preparation (in which MPOS is programmed).

PPOS(...) If touch-trigger switching probes have been connected, the function will transmit the current axis actual position to a *synchronous* axis (machining axis) in the switching point of the probe without taking into account all active axis zero shifts.

5.6.5 G138 workpiece position compensation

- Shifting and rotating the workpiece coordinate system
- Defining the workpiece zero point
- Acts additively on the active axis zero shift
- ”Inclined plane” function acts additively.

Restrictions:

- **Workpiece position compensation without axis transformation**: Using the workpiece position compensation, it is exclusively possible to rotate the 1st and 2nd logical axis around the 3rd logical axis.

  Workpiece position compensation with axis transformation: In case of active 5 axis or 6 axis transformation, the workpiece position compensation rotates the 1st and 2nd linear coordinate around the 3rd linear coordinate.
- The workpiece position compensation under G138 refers to the workpiece coordinates.
- G138/G139 must never be programmed in combination with a traversing motion.

5.6.6 G168 an G268 program coordinate shift

- Shifting the program coordinate system
- Defining the program zero point
- Acts additively on the active axis zero shift
- Acts “additively” to the ”Inclined plane” function

Restrictions:

- G169 switches off G168 as well as a possibly active G268.
- G168 and G268 must never be programmed in combination with a traversing motion.

5.6.7 G60 programmed contour shift

- Shifting a contour in the program coordinate system
- Active axis zero shifts have an “additive” effect.
- Acts “additively” to the program coordinate shift
- Acts “additively” to the ”Inclined plane” function

☞ **G60 is no coordinate system shift!**

Restrictions:

- The shift values of G60 are influenced by G38 (rotating, mirroring, scaling).

5.6.8 G352...G359 inclined plane

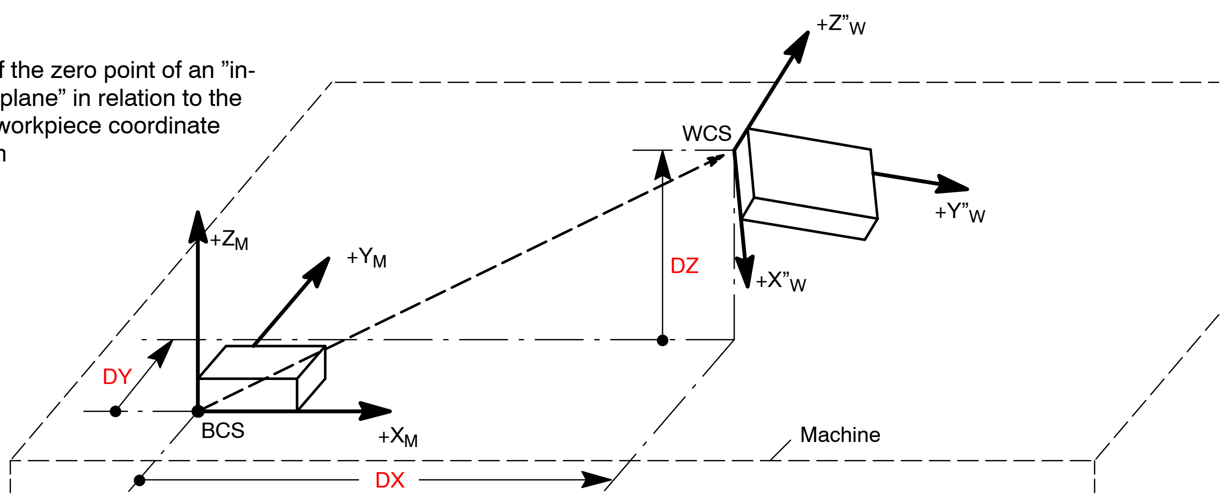
Function:

Using the ”Inclined plane” function, the workpiece (WCS) or program coordinate system (PCS), respectively, can be **shifted and oriented** freely in space. The basis workpiece coordinate system serves as reference. This allows for shifting of the workpiece or program zero point, respectively, in addition to rotating the WCS/PCS around several coordinates.
As a result, circular movements are possible within freely defined planes in space, and part programs for the machining of e.g. desk-type workpieces are considerably simplified.

The final position of the inclined plane is achieved by:

- basis rotations (Eulerian angles phi, theta, psi), see Section 5.5
- shifts (distances DX, DY, DZ), see figure.

Shift of the zero point of an ”inclined plane” in relation to the basis workpiece coordinate system



Positioning of the coordinate system around the distance DX, DY, DZ and orientation around the angles phi, theta and psi in relation to the basis workpiece coordinate system

BCS= basis workpiece coordinate system
WCS= workpiece coordinate system
(or program coordinate system)

The required shift values and angles of rotation may either

- be stored in tables, or
- be programmed in the G352 block directly.

An ”inclined plane” table can take up the data of max. 6 ”inclined planes” (see page 5–45 ff.).

To activate an ”inclined plane”, you first select the relevant table via G22. Subsequently, you select the desired ”inclined plane” using the G354..G359 functions.

Restrictions:

- ”Inclined plane” **without axis transformation**:
  The function has an effect on the first 3 logical axes of the channel. The names are set in the MP 701000010.

  ”Inclined plane” **with axis transformation**: The names of the working range coordinates of the first 3 linear coordinates in the MACODA parameter 7080 00010 (e.g. x,y,z) must be used for programming.
- The ”Inclined plane” functions G352 through G359 act modally and deselect each other mutually.
- The ”Inclined plane” functions G352 through G359 must not be programmed in combination with traversing motions.
- After control reset, the ”Power-up condition” function becomes active (see page 3–6). If the “Inclined plane” function is marked as “not active” in the init string, it will be deselected with control reset.
- Values for the ”inclined plane” are provided in tables. These contain information on the unit of the stored shift values (see also MP 902000010).

Relevant G functions (all workpiece coordinate systems):

G138: Workpiece position compensation on

G139: Workpiece position compensation off

G168: Program coordinate shift on

G169: All program coordinate shifts off

G268: Additive program coordinate shift on

G269: Additive program coordinate shift off

G60: Programmed contour shift on

G67: Programmed contour shift off

G22 IP<*Path/table name*>:
Activating ”inclined plane” table via NC block.

G352: Programming and switching on ”inclined plane” with all required parameters in the G352 block. An ”inclined plane” table is not necessary in these cases.

G353: Canceling active ”inclined plane”.

G354...G359: Activating one out of a max. of 6 ”inclined planes” from an ”inclined plane” table.

Relevant MACODA parameters (all workpiece coordinate systems):

3080 00200: Determines which file name extensions the control unit uses as identification criterion for axis zero shift and geometry compensation tables.
The file name extension possibly used in an ”inclined plane” table, must not be identical with the one entered here.

3080 00001: Search path for the file.

7010 00010: Determines the “logical axis designation“ for each channel (only necessary for G138 and G352..G359).

7010 00030: Classification for logical axes.
Defines the functional relevance of all axes on the channel.

7080 00010: Designations of the working range coordinates of the first 3 linear coordinates for G138 and G352..G359 in case of active axis transformation.

9020 00010: Specifies the unit (mm or inch) used to internally store the table values.

☞ **Changes in MP 9020 00010 also influence geometry compensation /axis zero shift tables and the interpretation of the compensation and shift values transmitted by the PLC!**

Relevant CPL functions (all workpiece coordinate systems):

COF(..) Provides the contour shift **G60** of an axis programmed last to the current channel (here, channel on which the program with the COF command is running).

DPC(..) Provides the parameters of the **G138** workpiece position compensation programmed last of an axis (shift values and angles of rotation) for the current channel (here: channel on which the program with the DPC command is running).

PPOS(...) If touch-trigger switching probes have been connected, the function will transmit the current axis actual position to a *synchronous* axis (machining axis) in the switching point of the probe without taking into account the active **G138** workpiece position compensation (only shifting, no rotating!).

SD(68), SD(168), SD(268) Reads system data of the control unit:
Totals and values of the G168, G268 program coordinate shifts programmed last.

5.6.9 Structure of ”inclined plane” tables

If you use the ”Inclined plane” function in connection with G354...G359, the shift values and angles of rotation have to be provided in a table. The required data for 6 ”inclined planes” can be specified per table. The control unit saves these tables as ASCII files with an accurately defined structure.

☞ **No special editor is available for these tables. Therefore, use a normal ASCII text editor or the NC editor.**

Table name

The table name is composed of a freely selected ”file name” and a ”file name extension”, if applicable.

☞ **If you would like to use a file name extension, please make sure that the extension does not collide with extensions already used for other files (e.g. ”.npv”, ”geo” etc.; see also MP 3080 00200)!**

File structure

”Inclined plane” tables are structured as follows:

- 6 lines for shift values and Eulerian angles.

All 6 lines have the same structure:

```
<G..><:><D1> <D2> <D3> <Phi> <Theta> <Psi>
```

The following applies:

`<G..>`	G function to activate an inclined plane (G354 through G359)
`<:>`	Colon
`<D1>` to `<D3>`	Shift values for the N^{th} coordinate (N=1..3), related to the BCS basis workpiece coordinate system.
`<Phi>`	Eulerian angle φ (see page 5–23). Syntax: PHI, Phi, phi
`<Theta>`	Eulerian angle ϑ (see page 5–23). Syntax: THETA, Theta, theta, The, the
`<Psi>`	Eulerian angle ψ (see page 5–23). Syntax: PSI, Psi, psi

Both positive and negative shift values and orientation angles may be entered. The unit of measure applying to the shift values depends on the unit as set in MP 9020 00010.

Structure of the ASCII file for inclined plane tables:

G354	:	<D1>	<D2>	<D3>	<Phi>	<Theta>	<Psi>
G355	:	<D1>	<D2>	<D3>	<Phi>	<Theta>	<Psi>
G356	:	<D1>	<D2>	<D3>	<Phi>	<Theta>	<Psi>
G357	:	<D1>	<D2>	<D3>	<Phi>	<Theta>	<Psi>
G358	:	<D1>	<D2>	<D3>	<Phi>	<Theta>	<Psi>
G359	:	<D1>	<D2>	<D3>	<Phi>	<Theta>	<Psi>
		Shift			Orientation		

☞ **The columns must be separated from each other by blanks or tabs. If a shift value or orientation angle is not required, value 0.000 must be entered.**

Example:

Shift values and orientation angles for 2 inclined planes (G354 and G355). Contents of the ASCII file:

```
G354 :  10.001 11.059 -22.050 25.000 -10.000 1.500
G355 : 100.345  0.000  30.444 25.000  10.000 0.000
G356 :   0.000  0.000   0.000  0.000   0.000 0.000
G357 :   0.000  0.000   0.000  0.000   0.000 0.000
G358 :   0.000  0.000   0.000  0.000   0.000 0.000
G359 :   0.000  0.000   0.000  0.000   0.000 0.000
```

5.6.10 Activating “inclined plane tables”

To activate an ”inclined plane” table via an NC block, use G22 with address IP:

Syntax:

`G22 IP` <*Path*><*Table name*>

☞ **If <*Path*> has not been specified, the PNC uses the search path (MP 3080 00001) to search for the <*Table name*>.**

Examples:

Activating the table ”schief.se” in the directory /**database** of the control unit:

`G22 IP schief.se`

Activating the table ”schief.se” in the **root directory** of the control unit:

`G22 IP /schief.se`

To activate the shift values and orientation angles, proceed as follows:

1. First activate the desired table as described above.
2. Activate the desired ”inclined plane” using the respective G function (G354..G359).

To deactivate an active ”inclined plane”, program G353.

5.6.11 Assigning shifts

1. Use MP 9020 00010 to specify in which unit of measure the values of axis zero shift tables and “inclined plane” tables are stored within the CNC and how the control unit is supposed to interpret the shift values transmitted by the PLC (in case of external axis zero shifts).

☞ **Changes in MP 9020 00010 also influence geometry compensation tables, tables for the ”inclined plane” and the interpretation of geometry compensation values transmitted by the PLC!**

CAUTION
Danger of damage to the machine or the workpiece!
Do not initiate a control reset during machining.

---

2. Initiate a control reset.

Applies additionally when workpiece position compensation is used:

1. If you intend to use G138 (workpiece position compensation), ensure the following:
   - Workpiece position compensation **without axis transformation**: The axis designations of the first 3 logical axes of the channel must be set to X, Y and Z in MP 7010 00010.
   - Workpiece position compensation **with axis transformation**: The function acts on the first 3 logical axes of the channel set in MP 7010 00010.

Applies additionally when axis zero shifts are used:

1. Create new axis zero shift table:
   MANAGE ▶ EDIT NEW FILE ▶ ZERO SHIFT TABLE

☞ **The file name extension of axis ZS tables are defined via MP 3080 00200 (individual parameter 1).**
Presetting: *.npv

2. Enter name.

3. Define directory for the table:
   via SWITCH DIRECTORY softkey
4. Press OK softkey.
5. Fill table with shift values for machining.

For information on how to call and process the axis ZS tables, please refer to ”Operating instructions, standard operator interface”.

Applies additionally when ”external axis zero shifts” are used:

1. Adapt PLC program module –B23XTNPV for the provision of the shift values and integrate it into the PLC program.

Applies additionally when the “inclined plane” is used:

1. If you intend to use G352..G359, please make sure for “inclined plane” **with axis transformation** that the first 3 coordinates in MP 7080 00010 are linear coordinates. Their designations have to be defined (e.g. x,y,z).

CAUTION
Danger of damage to the machine or the workpiece!
Do not initiate a control reset during machining.

---

2. If you have changed one of the configuration parameters mentioned above, initiate a control RESET.

Applies additionally when “inclined plane” tables are used:

1. Create new ”inclined plane” table:
   MANAGE ▶ EDIT NEW FILE ▶ TEXT-FILE / NC PROGRAM
2. Fill table with the required data. The required table structure is described in Section 5.6.9, page 5–45 ff.
3. Store finished table:
   Select FILE ▶ SAVE AS.
4. Enter name.
   The control unit is case sensitive!

☞ **If you have specified a file name extension, it must not collide with the extension of GEO tables (presetting ”.geo”) or axis ZS tables (presetting ”.npv”)! In case of doubt, check MP 3080 00200 or refrain from using file extension.**

5. Define directory for the table:
   via SWITCH DIRECTORY softkey
6. Press OK softkey.

5.6.12 Activating shifts

Axis zero shift

Precondition:

- The tool must not be in contact with the workpiece.
- Axis zero shift table for the machining program exists and contains the correct shift values.
- Any shifts still active must be taken into account when programming an axis zero shift because these may have an additive effect or overwrite each other mutually.

1. Activate axis zero shift table (see page 5–34).
2. Call the desired axis zero shift (G54..G259) at the required point in the program.
3. Execute the next machining phase with new zero point.

External axis zero shift

Precondition:

- PLC program module –B23XTNPV has been adapted for the provision of the shift values and integrated into the PLC program.
- The tool must not be in contact with the workpiece.
- Any shifts still active must be taken into account when programming an axis zero shift because these may have an additive effect or overwrite each other mutually.

1. Transfer relevant shift values to the control unit with the aid of the PLC program module – B23XTNPV.
   You should then program "WAIT" in order to halt the block preparation. This is the only way top make sure that the current shift values will be taken into account in all subsequent blocks.
2. Activate external axis zero shift via G160, G260 or G360, respectively.
3. Execute the next machining phase with new zero point.

Other shifts

Precondition:

- The tool must not be in contact with the workpiece.
- Any shifts still active must be taken into account when programming other shifts because these may have an additive effect or overwrite each other mutually.

1. Program the desired shift function (G138, G168, G268, G352..G359, G60) using the respective parameters.
2. Execute the next machining phase with new zero point.

Inclined plane

Precondition:

- The tool must not be in contact with the workpiece.
- Any other shifts still active (e.g. G138 workpiece position compensation) must be taken into account when activating an "inclined plane" because these may have an additive effect.

1. If you want to use G354 through G359, activate the matching "inclined plane" table (see page 5–45).
2. Call the desired "inclined plane" at the required point in the program. For this purpose, program
   G354 ... G359 ("inclined plane" table required)
   – or –
   G352 with the respective parameters.
3. Execute the next machining phase with new zero point.

5.6.13 Deactivating shifts

Axis zero shift

Precondition:

- The tool must not be in contact with the workpiece.

1. Active axis zero shifts are completely deactivated using G53!
   If only the 1st or 2nd additive axis zero shift is to be deactivated, you have to program G153 or G253, respectively.

External axis zero shift

Precondition:

- The tool must not be in contact with the workpiece.

1. Program G167.

Other shifts

Precondition:

- The tool must not be in contact with the workpiece.

1. Deactivate the respective shift function using G139, G169, G269 or G67.

Inclined plane

1. Deactivate any G60 (programmed contour shift) still active.
2. Deactivate any G168 or G268 (program coordinate shift) still active.
3. Program G353.

5.7 Working range coordinate programming

Function:

Working range coordinate programming is used to:

- program the **position of the tool tip** (TCP)
- program the **tool orientation**
- superimpose the positioning of the tool tip with an **orientation movement of the tool**.

The working range coordinate programming allows for:

- contour programming independent of the axis
- taking tool compensations into account automatically
- optimum path stability of the TCP movement with simultaneous orientation movement
- the creation of a defined feedrate on the TCP path.

The precondition for working range coordinate programming is a previously activated “axis transformation” (see Section 5.8, page 5–77 ff.). During the interpolation, the axis transformation determines the axis setpoint values of all necessary physical axes at the machine from the programmed working range coordinates.
Furthermore, the axis transformation considers the current tool dimensions (tool lengths) and takes these into account in the direction of the coordinates of the tool coordinate system (TCS: tool coordinate system). As a consequence, compensations can be taken into account correctly, even with a continuously changing tool orientation.
An axis-specific axis transformation is activated for each machine, so that the working range coordinates used in the NC program will be interpreted correctly.

Axes not taken into account in the axis transformation, are still programmable using their axis addresses in case of active axis transformation. The program coordinates for axes of this type are also referred to as pseudo coordinates.

Working range coordinate programming is determined by the following **properties**:

- **working range coordinates** are composed of 3 position coordinates and max. 3 orientation coordinates:
  - **position coordinates** (TCP: tool center point, tool tip) are determined by three Cartesian coordinates x, y and z (see page 5–53)
  - **orientation coordinates** φ (phi), ϑ (theta), ψ (psi) determine the tool orientation.
    A distinction is made between:
  - scalar orientation (see page 5–55)
  - vector orientation (see page 5–56)
  - tensor orientation (see page 5–59)

- The names of all six working range coordinates are defined for each channel via MACODA parameter.

Working range coordinate programming

- Working range coordinates refer to the current active workpiece coordinate system (WCS).
- Working range coordinates do not have axis-type properties that can be configured (e.g. maximum acceleration or positioning type).
- Working range coordinates can only be used in connection with an axis transformation.
- The axis transformation determines which subset of the six coordinates can be used.
- The position coordinates are programmed in Inches or mm, in accordance with G70/G71.
- Orientation coordinates are always programmed in degrees.

  Their values are restricted to the intervals
  φ (phi): $0° \leqslant \varphi < 360°$,
  ϑ (theta): $0° \leqslant \vartheta \leqslant 180°$,
  ψ (psi): $0° \leqslant \psi < 360°$
- An **orientation movement** of the tool can be programmed independent of the path movement.

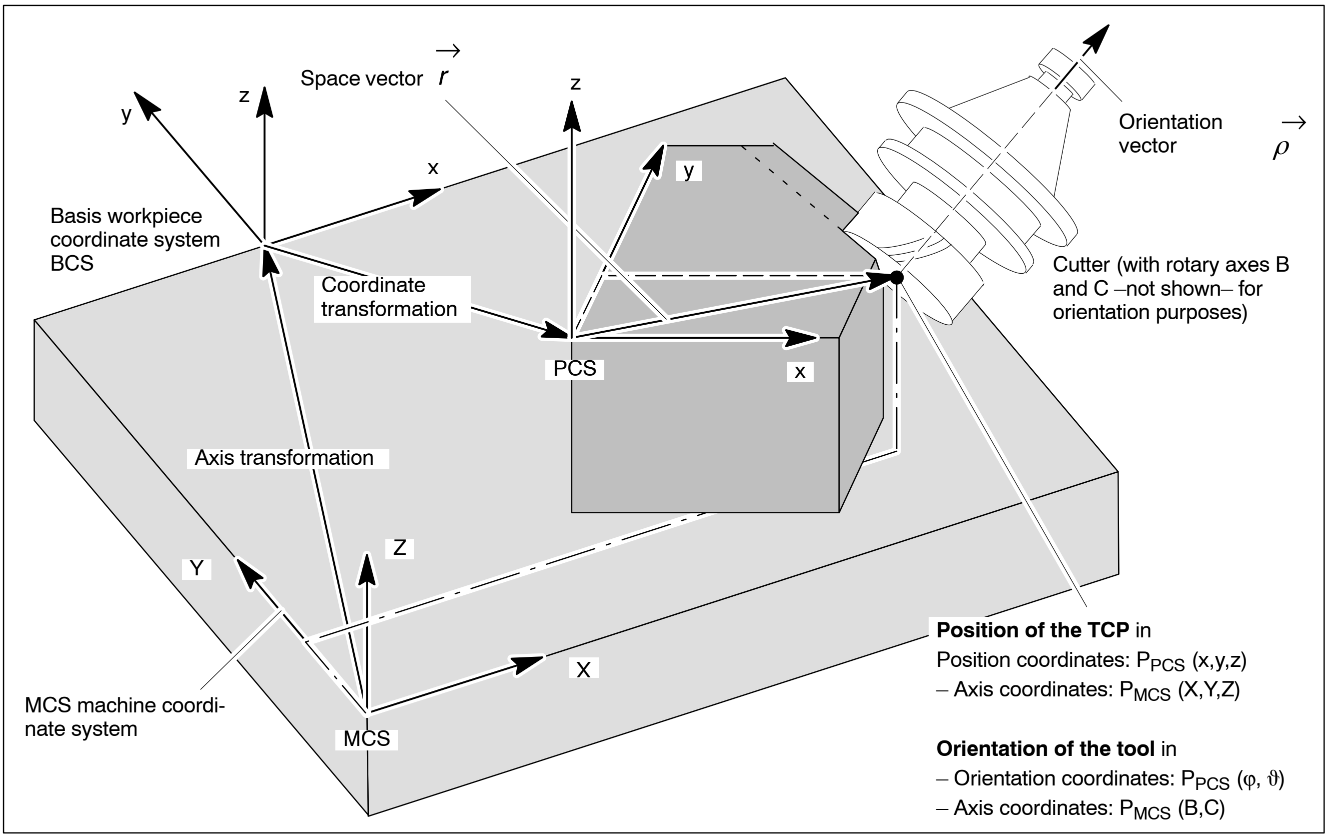
Feedrate for working range coordinate programming:

- The programmed feedrate refers exclusively to the programmed (linear) position coordinates. Orientation and further pseudo coordinates are carried along synchronously.
- If no position coordinates but orientation coordinates have been programmed, the programmed feedrate is transferred to the orientation coordinates. The programmed pseudo coordinates may be guided along synchronously, if applicable.
  The alternative in case of active G94 is ”Omega” programming.
- If no working range coordinates have been programmed, but exclusively pseudo-coordinates, the behavior with regard to the feedrate will be as if there is no active working range coordinate programming.

Working range coordinate programming

5.7.1 Position of the tool tip (TCP)

3 position coordinates of the working range coordinates describe the position of the tool tip (TCP) in relation to the current program coordinate system (PCS).
Position coordinates will be designated ”x”, ”y” and ”z” in the following.



For conclusions from working range coordinates (programmed coordinates in the NC program) to axis setpoint values, the control unit has to execute different “reverse transformations”.

1. With the aid of the active “coordinate transformations”, the programmed position coordinates (PCS-related) are “calculated back” to the basis workpiece coordinate system (BCS) first.
   Coordinate transformation includes the following compensations
   - G138 workpiece position compensation
   - G352, G354...G359 inclined plane.
   - G168, G268 program coordinate shift
2. From the basis workpiece coordinate system (BCS), the position coordinates relating to the basis workpiece coordinate system are now transformed to the machine coordinate system (MCS) in the form of axis setpoint values (for the physical axes of the machine) using the machine-specific axis transformation.

The working range coordinates are calculated into axis setpoint values in the “interpolation cycle”.

5.7.2 Tool orientation (vector and tensor)

Overview

The kinematics specific to the machine has to be taken into account when a tool is oriented is space. And it makes no difference, whether the orientation is effected by rotary axes, telescope axes or other drive types.

Using **orientation**, the main axes of a tool, laser, gripping device and similar are aligned in a specified direction in space. In this context, it makes no difference whether the orientation of the tool axis runs parallel with one of the main directions of the machine axes or not.

At present, the following orientations exist in the PNC

Orientation	By specification of the rotary axis positions	Vector orientation	Linear orientation movement	Tensor orientation
Orientation movement	Linear in the rotary axes	Rotation of the orientation vector around an axis of rotation fixed in space (coordinate of rotation movement)	The tip of the orientation vector moves on a curved path on an imaginary $\varphi - \vartheta$ plane.	Rotation of the orientation tensor around an axis of rotation fixed in space
Axis transformation	3032101 3033101	3232201	3232101	3333301

Depending on machine kinematics, the orientation in space is conditional on whether one, two or three **orientation coordinates** are used. The orientation is related to the program coordinate system (PCS) and is determined by the angles φ (phi), ϑ (theta), ψ (psi) or using Cartesian components (see tensor, vector orientation).

- Vector orientation
  This function deals with the orientation of rotation symmetrical tools (e.g. lasers, cutters). The tool orientation is described by the polar coordinates φ and ϑ or in the same manner by the orientation vector $\vec{\rho}$. The orientation vector rotates around an axis of rotation fixed in space (large circular movement of the vector tip). φ jumps can occur during the movement (for details on singularity, please refer to page 5–64), which the corresponding axis transformation deals with as required.
- Linear orientation movement
  The basic function is identical with the vector orientation. However, the orientation movement is always carried out on a straight line on an imaginary $\varphi - \vartheta$ plane. In contrast to the vector orientation, no φ jumps occur in the movement.

Working range coordinate programming

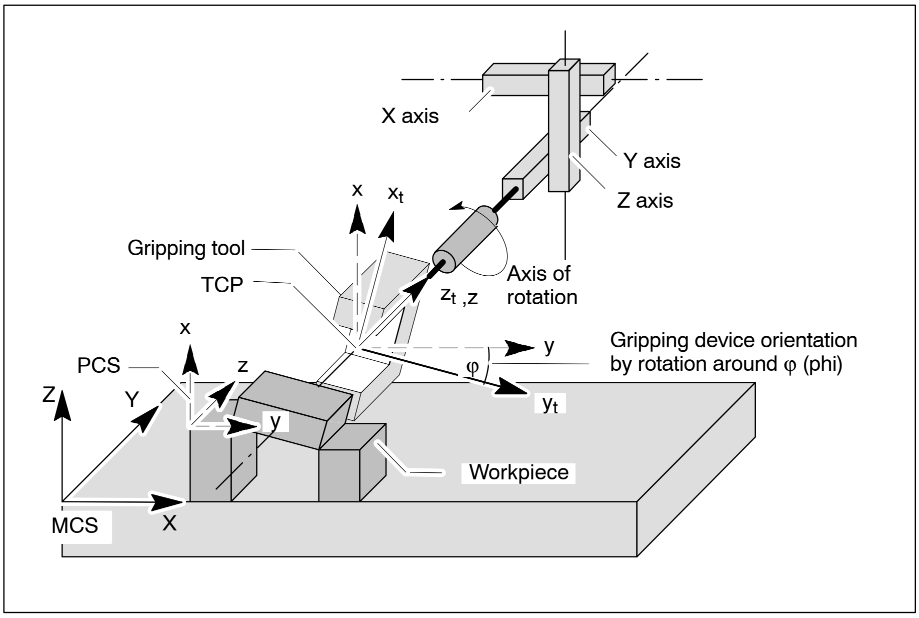
- Tensor orientation:
  This function is used to describe the orientation of non-rotation symmetrical tools.
  A tool coordinate system (TCS) that is permanently connected to the tool has an orientation in relation to the program coordinate system (PCS) which is determined by the Eulerian angles φ, ϑ, and ψ or the 3x3 orientation tensor. The movement of the orientation tensor is a rotation around an axis of rotation fixed in space.

Scalar orientation

In case of “scalar orientation” only one orientation coordinate (e.g. phi (φ)) is programmed.

Orientation of the tool (cutter, laser, robot gripping device) is generated by specifying the ”phi” angle. The movement is not dependent on any special axis kinematics.

Example: Rotation of a robot gripping device around φ with a rotary axis.



The configuration shown in the example can also have several rotary axes to generate the rotation of the gripping device.

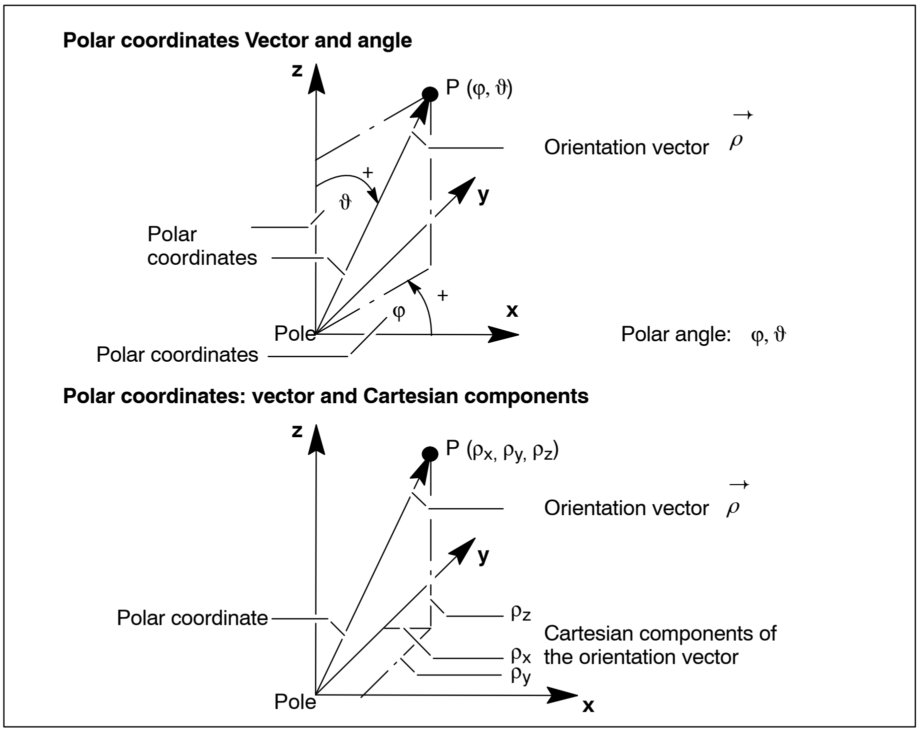
An axis transformation of this type has not been realized in the PNC at present.

Working range coordinate programming

Vector orientation

The orientation of a **rotation symmetrical tool** is programmed using the first two **orientation coordinates phi (φ)** and **theta (ϑ)**. They define an **orientation vector** $\vec{\rho}$, specifying the orientation of the tool longitudinal axis in relation to the pole of the coordinate system. The angles and the vector have the meaning of **polar coordinates**.

The "polar coordinates" of a point P(φ,ϑ) are the **distance** (= length of the orientation vector) of P to the pole of the coordinate system and the **polar angles** φ (phi), ϑ (theta) which are located between the polar axes x and z and the orientation vector.



Alternatively, the orientation of a tool can also be achieved using the **Cartesian components** ρ_x, ρ_y, ρ_z**)** of the orientation vector $\vec{\rho}$ (with standardized length 1).

$$\vec{\rho} = \begin{bmatrix} \rho_x \\ \rho_y \\ \rho_z \end{bmatrix} \quad \text{with } |\vec{\rho}| = \sqrt{\rho_x^2 + \rho_y^2 + \rho_z^2} = 1$$

To be able to program the vector orientation, it has to be activated with the machine-specific axis transformation (at present Type 3232201). The axis transformation is configured in MACODA.

Properties of the orientation vector:

- The orientation vector is relevant for rotation symmetrical tools (e.g. cutter or laser).
- The orientation vector is located along the tool symmetry axis and points to the tool holder.

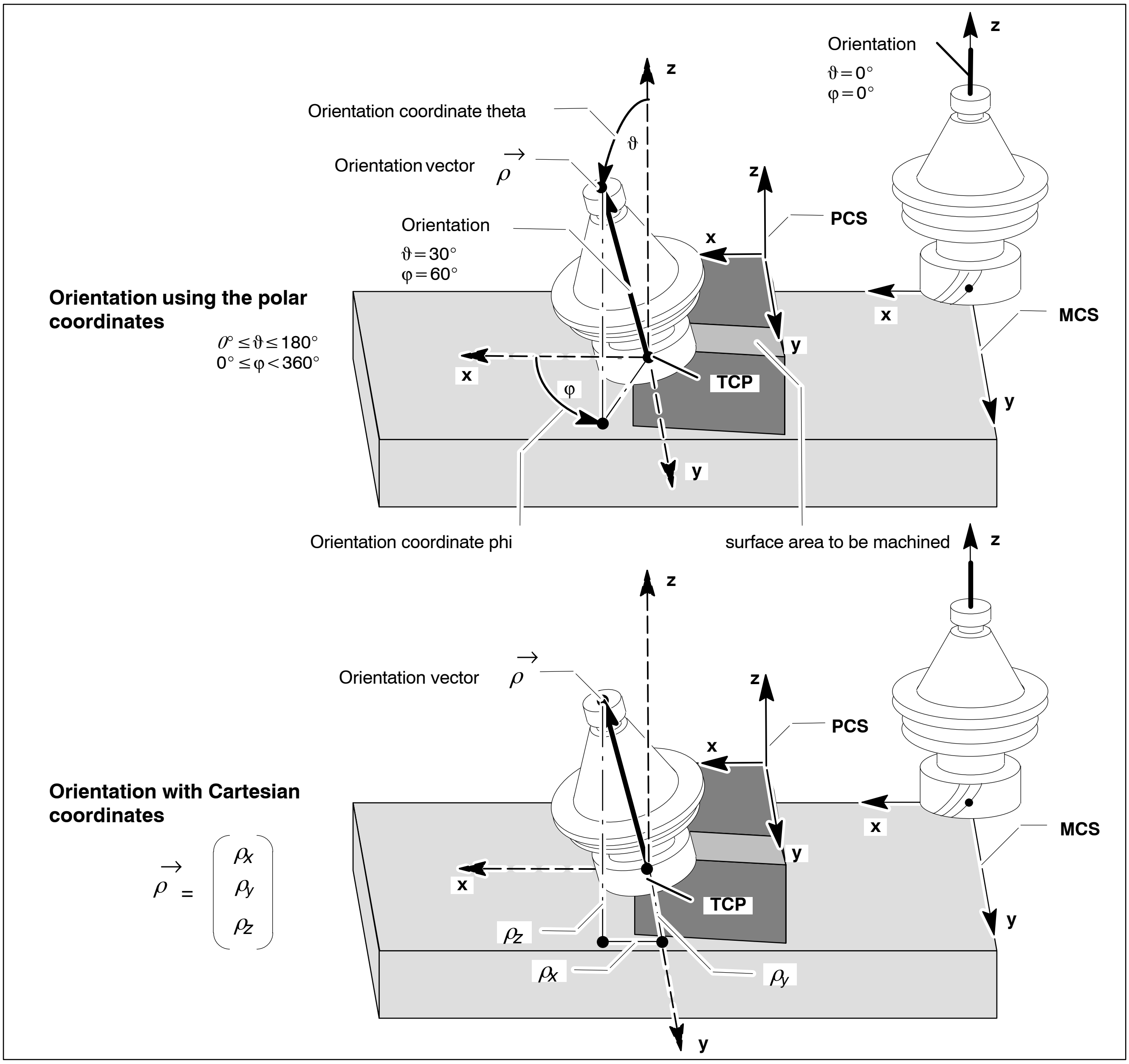
- As long as the orientation vector is programmed with constant orientation coordinates phi (φ) and theta (ϑ), its orientation relative to the reference coordinate system will not change during a positioning process.
- The orientation vector rotates around an axis of rotation fixed in space (large circular movement of the vector tip). φ jumps can occur during the movement (see page 5–64 ff.), which the corresponding axis transformation deals with as required.

Example (2 orientation options):

A workpiece surface intended to be machined is spatially inclined with respect to the program coordinate system (PCS) at the angles φ= 60° and ϑ= 30°.

- The rotation symmetrical tool is oriented in space so that the orientation vector $\vec{\rho}$ of the tool longitudinal axis is parallel with the inclined workpiece surface with regard to both inclination angles.
  Using φ = 60° and ϑ = 30°, the programming for the orientation vector would be: O(60,30).
- Another possibility is that the orientation of the tool longitudinal axis as Cartesian position of the tip of the orientation vector $\vec{\rho}$ is specified with regard to the programming coordinate system.
  Using e.g. x=10, y=15, z=40, the orientation vector programming would read: O(10,15,40).

Working range coordinate programming



Working range coordinate programming

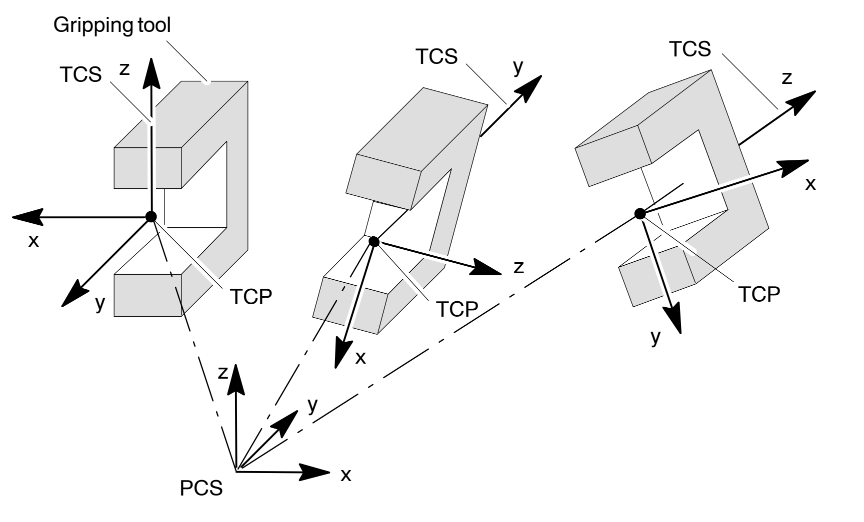
Tensor orientation

This orientation is relevant for **non-rotation symmetrical tools** in space (e.g. robot gripping device).
A tool coordinate system (TCS), which is permanently connected to a tool, is spatially oriented against a reference coordinate system (PCS, MCS,..) using a **3x3 orientation tensor** $\overleftrightarrow{O}$.

Alternatively, the orientation of the TCS can be programmed using the **Eulerian angles** (3 orientation coordinates φ (phi), ϑ (theta) and ψ (psi)).

To be able to program the tensor orientation, it has to be activated using the machine-specific axis transformation (at present type 3333301). The axis transformation is configured via MACODA.

Different orientations and reference positions of the TCS



Orientation using the 3x3 orientation tensor

The orientation of the tool coordinate system (TCS) is expressed by a so-called ortho-normalized orientation tensor $\overleftrightarrow{O}$ – also referred to as **rotation matrix**.

Definition:

The rotation matrix is composed as follows (see figure below):

$$\overleftrightarrow{O} = \begin{bmatrix} O_{11} & O_{12} & O_{13} \\ O_{21} & O_{22} & O_{23} \\ O_{31} & O_{32} & O_{33} \end{bmatrix}$$

The lines and columns of the tensor have length ”1” and stand on each other perpendicularly.

The columns of the tensor are the vector components of the **basis vectors** $\vec{e}^{\,t}_x$, $\vec{e}^{\,t}_y$, $\vec{e}^{\,t}_z$ of the rotated TCS relating to the reference coordinate system:

$$\overleftrightarrow{O} = \begin{pmatrix} o_{11} & o_{12} & o_{13} \\ o_{21} & o_{22} & o_{23} \\ o_{31} & o_{32} & o_{33} \end{pmatrix}$$

$$\vec{e}^{\,t}_x = \begin{pmatrix} o_{11} \\ o_{21} \\ o_{31} \end{pmatrix} \quad \vec{e}^{\,t}_y = \begin{pmatrix} o_{12} \\ o_{22} \\ o_{32} \end{pmatrix} \quad \vec{e}^{\,t}_z = \begin{pmatrix} o_{13} \\ o_{23} \\ o_{33} \end{pmatrix}$$

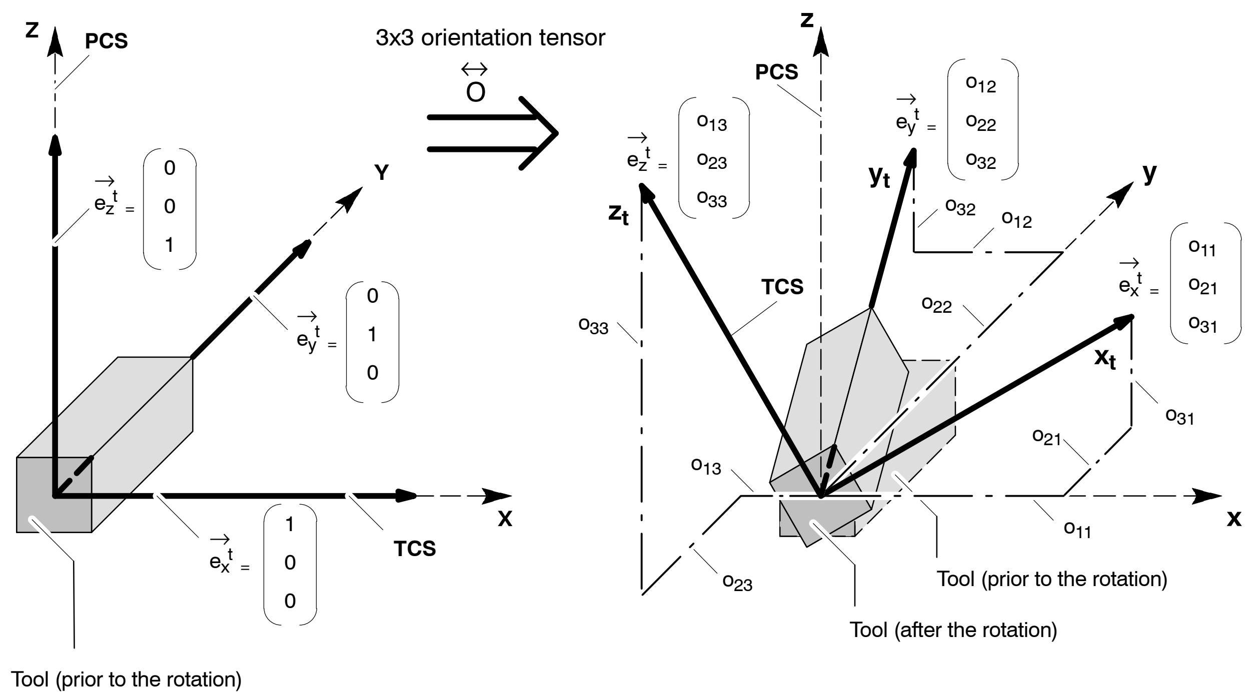
The basis vectors $\vec{e}^{\,t}_x$, $\vec{e}^{\,t}_y$, $\vec{e}^{\,t}_z$ ultimately define the “arms” of the TCS.

Properties of the orientation tensor:

- The orientation tensor is relevant for non-rotation symmetrical tools (e.g. gripping device).
- As long as the orientation tensor is programmed with constant Eulerian coordinates phi (φ) and theta (ϑ) and psi (ψ), its orientation relative to the reference coordinate system will not change during a positioning process.

Working range coordinate programming

Example:
3x3 orientation tensor of a non-rotation symmetrical tool in space.



$\overleftrightarrow{O}$ = 3x3 orientation tensor: (Rotation matrix)

$\vec{e}$ = Basis vectors (column vectors) of the TCS

o_{xx} = Vector components of a basis vector (=element of the orientation tensor)

☞ **The disadvantage of the orientation tensor is that its nine components are not independent of each other, because three coordinates are sufficient for a general orientation. Thus the Eulerian angles, which can be used to define an orientation, represent an alternative to the orientation tensor.**

Orientation of the orientation tensor using the Eulerian angles φ, ϑ and ψ:

Three orientation coordinates are sufficient for a general orientation of the TCS. Three consecutive rotations with the Eulerian angles φ (phi), ϑ (theta) und ψ (psi) around the main coordinates of the PCS gives the TCS its new orientation (for details, see Section 5.5).

5.7.3 Orientation movement of the tool

Each orientation is simultaneously connected with a movement of the tool around its TCP, if new orientation parameters have been programmed in relation to a prior orientation.
This achieves, e.g. a synchronous new orientation of the tool axis relative to the path movement.

In case of the orientation movement, a distinction is made between

- the movement of the **orientation vector** (in case of vector orientation of rotation symmetrical tools, e.g. 5 axis machine)
- the movement of the **orientation tensor** (in case of TCS orientation of non-rotation symmetrical tools, e.g. 6 axis machine).

Orientation vector movement

For the orientation vector, as opposed to the orientation tensor (axes of a non-rotation symmetrical tool), it is only important how **one** coordinate of the TCS is aligned spatially. This coordinate is, e.g. the longitudinal axis of a rotation symmetrical tool. A movement of the orientation vector corresponds to a movement of the tool longitudinal axis around its TCP.

There are two options for the orientation movement:

1. Orientation movement with **internally calculated axis of rotation**:
   - the start orientation vector $\vec{\rho}_a$ rotates around an internally calculated axis of rotation $\vec{u}$ to the position corresponding to the end orientation vector $\vec{\rho}_e$.
   - the axis of rotation $\vec{u}$ stands perpendicular to the orientation vector.
   - the movement of the orientation vector describes the angle $\beta \leqq 180°$ and thus always takes the shortest distance between start and end orientation.
   - the orientation movement is always carried out so that the orientation vector remains in the plane defined by its start condition $\vec{\rho}_a$ and the programmed end condition $\vec{\rho}_e$. This is a plane surface.
   - to be able to calculate an axis of rotation for the orientation internally, the start and end orientation of the orientation vector may not run parallel or anti-parallel. If the axis of rotation is programmed with ROTAX(..), this restriction is not applicable.

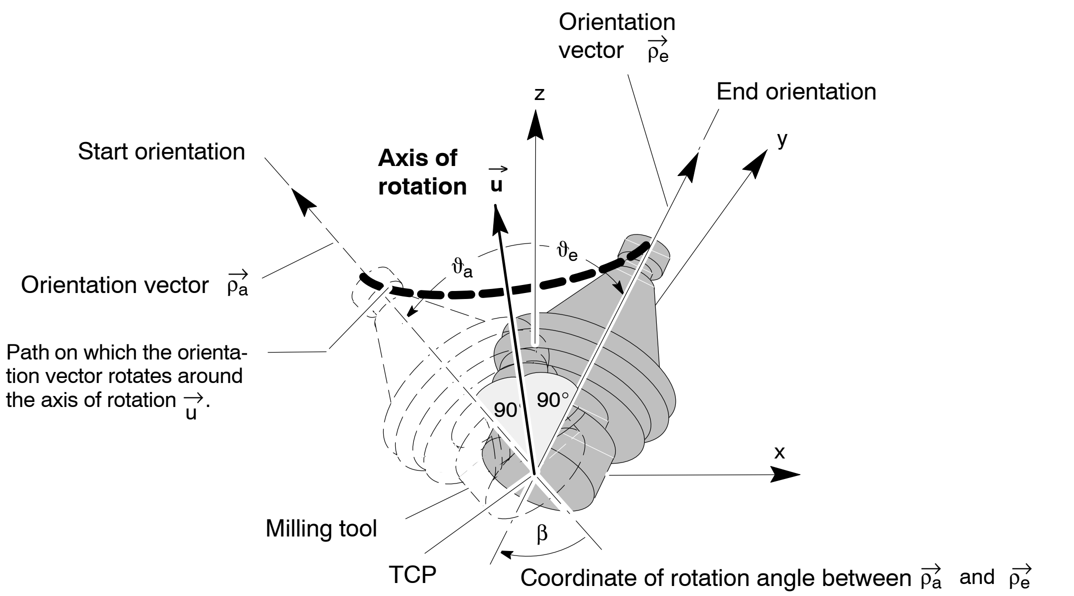
2. Orientation movement with **programmed axis of rotation**:
   - the start orientation vector $\vec{\rho}_a$ rotates around a programmed axis of rotation $\vec{u}$ at a programmed angle of rotation β.
   - the axis of rotation is programmed with ROTAX(..), the angle of rotation with O(..).
     The restriction that the axis of rotation $\vec{u}$ has to stand perpendicular on the orientation vector $\vec{\rho}$ is not applicable.
   - the programmed angle of rotation may have freely defined values, the restriction $\beta \leqq 180°$ is not applicable. It is generally interpreted as incremental value. E.g. an angle of O(360) rotates the

Working range coordinate programming

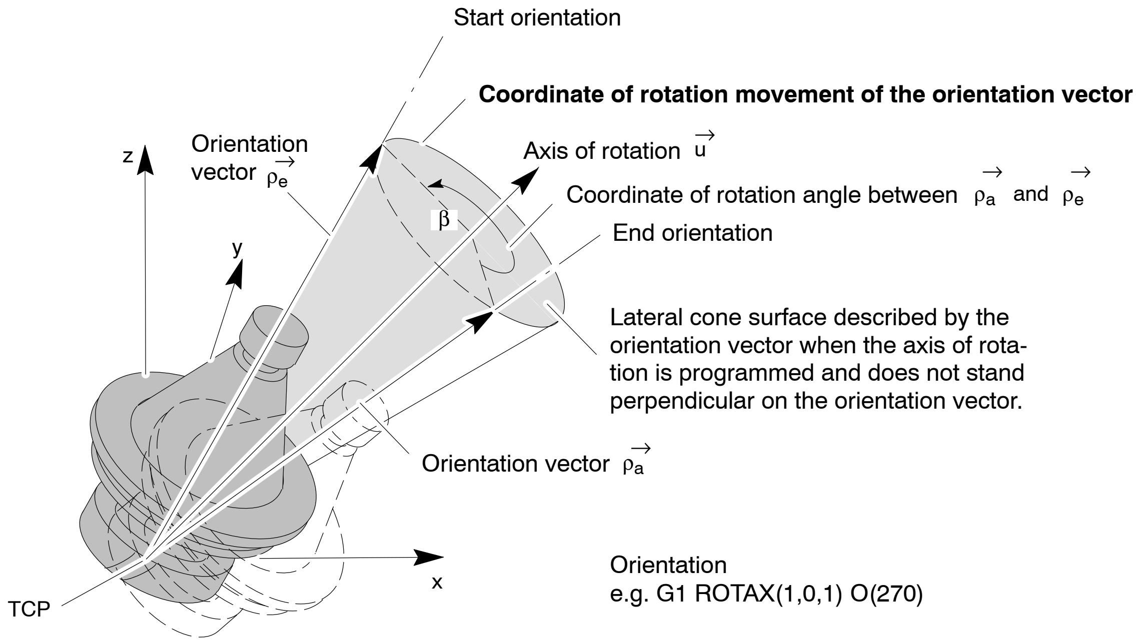
start orientation vector $\vec{\rho}_a$ by exactly 1 rotation to its original position.

- the desired sense of rotation is generated by positive and negative angles of rotation.
- the orientation vector describes a lateral cone surface.

Orientation movement with internally calculated axis of rotation



Orientation movement with programmed axis of rotation



Rotation speed of the orientation vector:

The rotation speed of the orientation vector depends on whether a TCP movement is taking place in addition to the orientation movement.

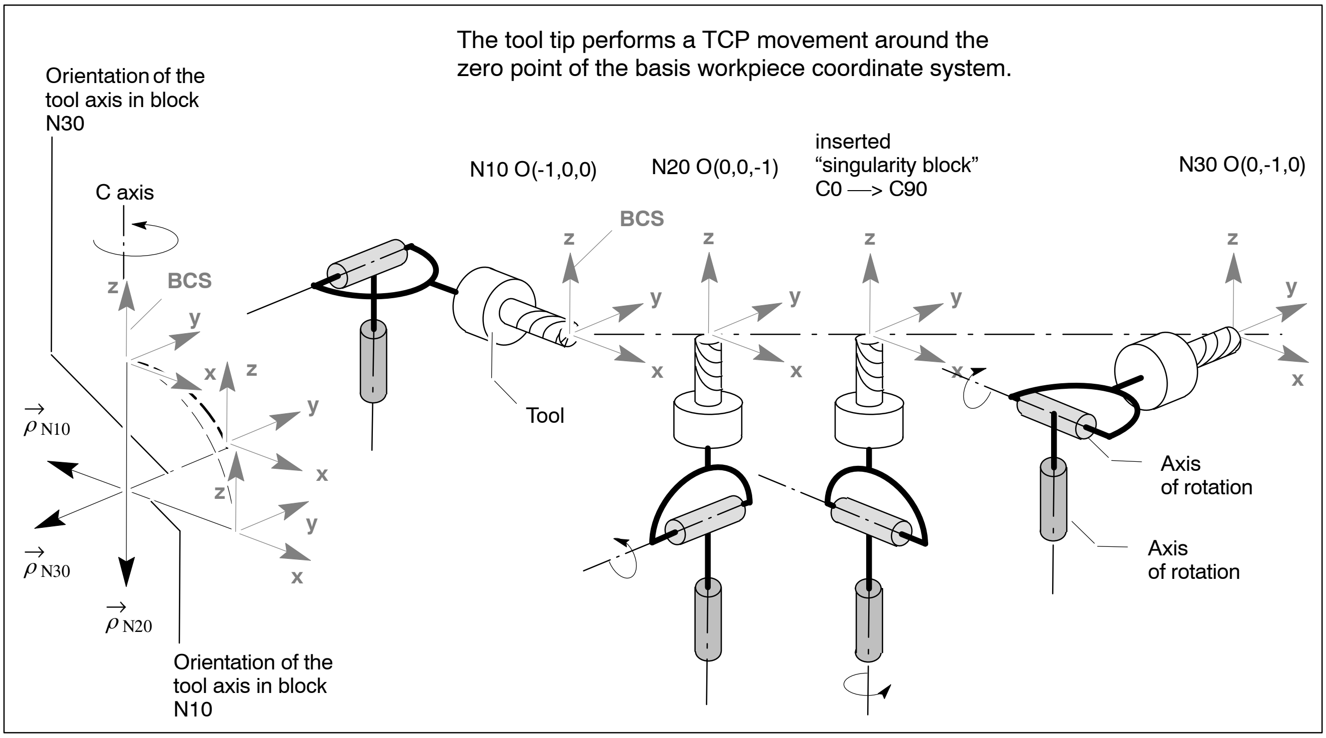
- TCP and orientation movement:
  The programmed feedrate refers exclusively to the TCP movement. The orientation movement ”follows” synchronously.
- Pure orientation movement:
  The programmed feedrate is the angle speed of the coordinate of rotation movement around the axis of rotation $\vec{u}$.
  Movements of other axes without TCP share ”follow” synchronously. If OMEGA is programmed for feedrate programming in addition to F, the angle speed corresponds to the OMEGA value.

Singularity of the orientation vector movement:

- In case of axis configurations – as shown in the figures below – there will be discontinuous axis movements if the orientation vector is moved non-tangentially by position $\vec{\rho} = \begin{bmatrix} 0 \\ 0 \\ \pm\ 1 \end{bmatrix}$.

  Example: jump of a C axis from 0 degrees to 90 degrees.

  In these cases, the PNC is capable of inserting a **“singularity block” automatically** , which rotates the C axis to the required position before positioning is continued. See figure below:



Working range coordinate programming

Linear orientation movement

The **linear orientation movement** also uses the coordinates phi (φ) and theta (ϑ). However, the movement is carried out on a straight line on an imaginary φ–ϑ plane.

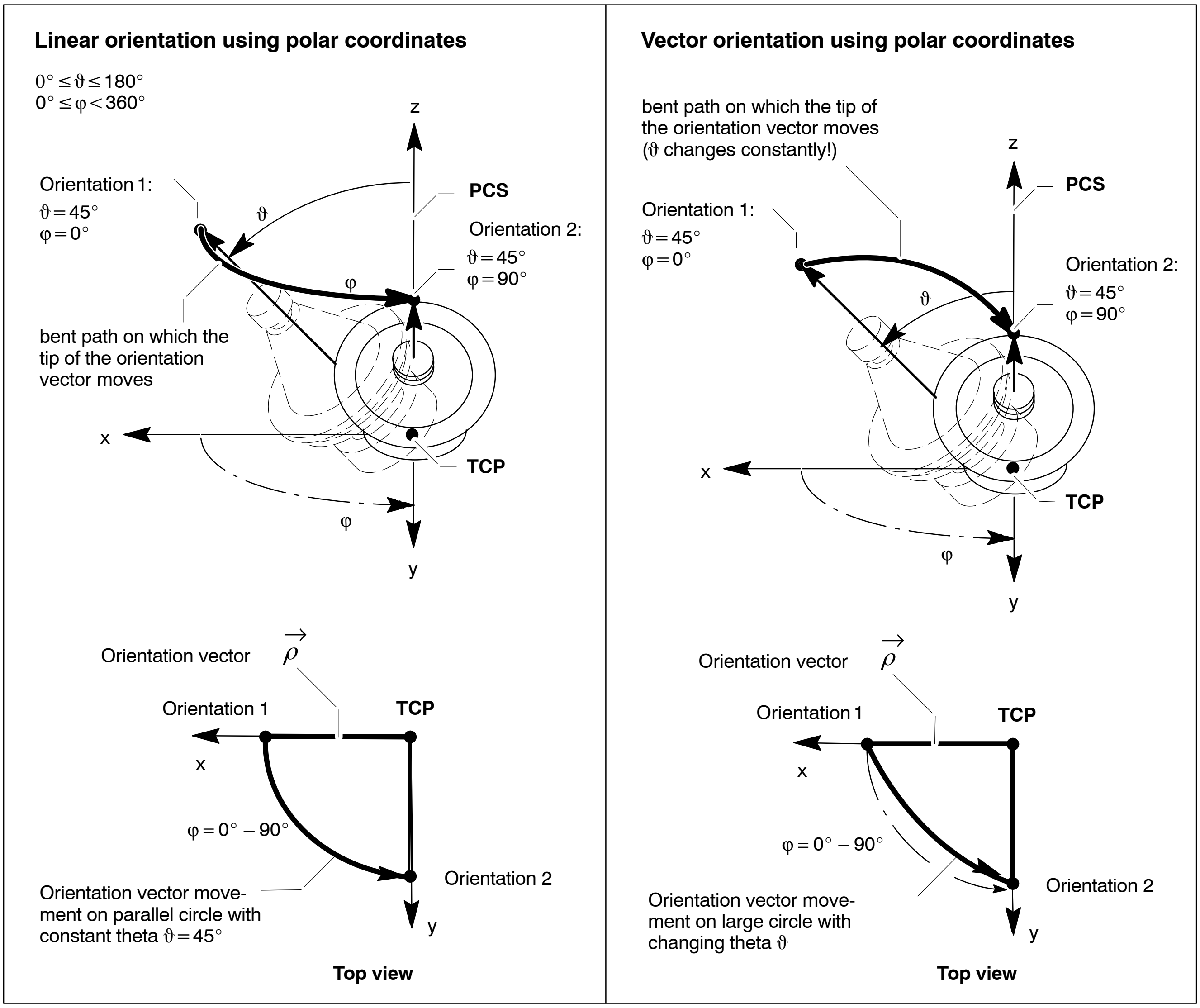
On the unit sphere, the tip of the orientation vector does not describe a large circle, as opposed to the vector orientation, but a more or less **bent path** (see also figure below).
The linear orientation movement does not require a singularity treatment by the corresponding axis transformation (page 5–64), because a φ jump cannot occur.

The special properties of the linear orientation movement and the difference to the vector orientation are outlined below:

- The prerequisite is an axis transformation with two orientation coordinates and the coordinate identification 1 configured in MACODA (example 3232101). Activating this type of axis transformation using Coord(<n>) automatically activates the linear orientation movement.
- Programming is effected via the coordinate names **phi..** and **theta..** or via the vector syntax O(<ρ_x>,<ρ_y>,<ρ_z>). Both programming methods are equivalent.
- A special path search logic makes sure that rotations greater than 180° do not occur.
- A conversion of the $\varphi - \vartheta$ values to the definition interval $0° \leqslant \varphi < 360°$, $0° \leqslant \vartheta \leqslant 180°$, is not performed. In the display, $\varphi - \vartheta$ values outside of the definition intervals thus occur.
- In case of a path movement (TCP movement), the orientation coordinates have no share in the feedrate, they ”follow” synchronously.
  If no TCP movement is carried out, the programmed feedrate F or OMEGA is related to the straight line movement of phi (φ) and theta (ϑ).
- Programming an axis of rotation with the functions ”ROTAX(..) O(..)” is **not** permitted.
- The mirroring and scaling functions are programmed the same way as for vector orientation.
- The function “Online compensation” can be used for phi (φ) and theta (ϑ).

Working range coordinate programming

The figure below shows the “linear orientation movement” in comparison with the “vector orientation movement“:



Comparison of orientation movements:

Orientation movement	**Path curve of the orientation tip**	**Orientation coordinate theta**	**Orientation coordinate phi**
Linear	on parallel circle	constant between the limits $0° \leq \vartheta \leq 180°$ while phi is running	$0° \leq \varphi < 360°$
Vector	on large circle (bent less strongly than on parallel circle)	varying between the limits $0° \leq \vartheta \leq 180°$ while phi is running	$0° \leq \varphi < 360°$

Working range coordinate programming

Orientation tensor movement

In contrast to the orientation vector, all three coordinates of the TCS have to be aligned spatially in case of non-rotation symmetrical tools. This is achieved with the orientation tensor which controls the orientation of the tool coordinate system.

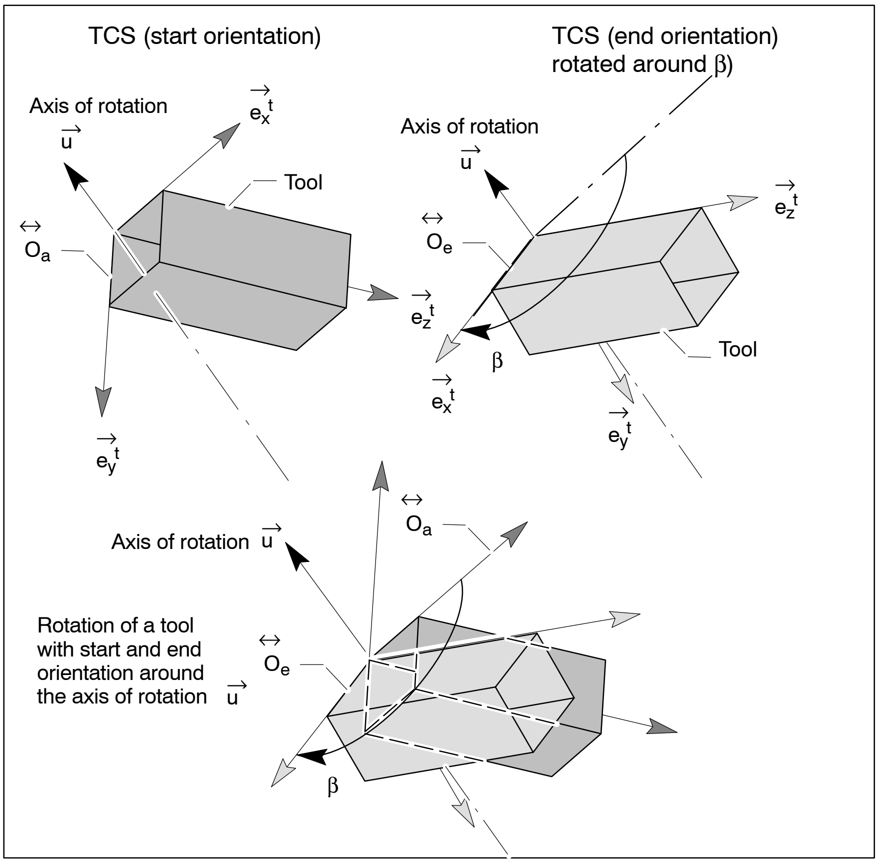
The following variables are involved in the movement of the orientation vector:

- $\overleftrightarrow{O}_a$: start orientation tensor (start orientation of the TCS)
- $\overleftrightarrow{O}_e$: end orientation vector (end orientation of the TCS)
- $\vec{u}$: Axis of rotation
- β : Angle of rotation

To execute an orientation movement, the following variables may be specified:

1. Orientation movement with specification of $\overleftrightarrow{O}_a$ and $\overleftrightarrow{O}_e$:
   - The start orientation tensor $\overleftrightarrow{O}_a$ rotates around an internally calculated axis of rotation $\vec{u}$ to the end orientation tensor $\overleftrightarrow{O}_e$. The following applies to the internally calculated angle of rotation β : $\beta \leqslant 180°$.

     The orientation movement is always carried out on the shortest path. It is independent of the special axis kinematics.



2. Orientation movement with specification of $\overleftrightarrow{O}_a$ and programmed axis of rotation $\vec{u}$ and angle of rotation β :
   - The start orientation tensor $\overleftrightarrow{O}_a$ is rotated around the programmed axis of rotation $\vec{u}$ at the angle β.
   - Orientation of the axis of rotation is programmed with "Cartesian components" or "polar coordinates" of the reference coordinate system with the aid of function ROTAX(..).
   - Movements of the orientation tensor are created with the function ROTAX(...) O(..).

     Example:

     using ROTAX(...) O(360), the start orientation tensor $\overleftrightarrow{O}_a$ rotates by exactly 1 rotation to its original position.
     The programmed angle of rotation is generally interpreted as incremental value.

   - The desired sense of rotation is generated by positive and negative angles of rotation.

The TCS coordinates $\vec{e}'_x, \vec{e}'_y, \vec{e}'_z$ in general describe a lateral cone surface during the orientation movement around the axis of rotation $\vec{u}$.

Exception: $\vec{u}$ stands parallel with $\vec{e}'_x$, $\vec{e}'_y$ or $\vec{e}'_z$. Then the cone surfaces turns into a circular surface.

Rotation speed of theorientation tensor:

The rotation speed of the orientation tensor depends on whether a TCP movement is taking place in addition to the orientation movement.

1. **TCP and orientation movement**:
   The programmed feedrate refers exclusively to the TCP movement. The orientation movement "follows" synchronously.
2. **Pure orientation movement**:
   The programmed feedrate is the angle speed of the coordinate of rotation movement around the axis of rotation $\vec{u}$.
   Movements of other axes without TCP share "follow" synchronously.
   If OMEGA is programmed for feedrate programming in addition to F, the angle speed corresponds to the OMEGA value.

Singularity of the orientation tensor movement:

Singularities occur in dependence on the axis transformation used. A singularity will occur for the 6 axis transformation type 3333301, for example, if the orientation tensor results in a position of $\pm 90°$ for the middle of the three rotary axes.
Discontinuous orientation movements at this location are bridged by **automatic insertion of an NC block**.

Working range coordinate programming

Relevant NC functions (orientation and orientation movement):

Coord(<i>): Activating working range coordinate programming. Axis addresses involved in the switchover to working range coordinates must not be programmed.
i: 1..5, number of the axis transformation to be activated.

Coord(0): Deactivating working range coordinates.

☞ **The position coordinates are programmed in Inches or mm, in accordance with G70/G71.**

The functions for orientation and orientation movement of the tool with **working range coordinates** are subdivided into

- Vector orientation and
- tensor orientation.

The Cartesian components of the orientation vector (ρ_x, ρ_y, ρ_z) are automatically standardized to 1 within the NC.
As a consequence, e.g. the specifications O(1,2,4), O(2,4,8) and O(0.5,1,2) are identical.

Vector orientation

The **orientation of the orientation vector** can be programmed using one of the following 5 alternatives:

phi<φ> theta<ϑ>: Orientation with polar angles φ, ϑ (polar angle names "phi" and "theta" are determined in MP 7080 00010 [4..5]). Programming is possible absolutely/incrementally and in degrees.
Example: G1 x10 y50 z30 phi90 theta90

O(<φ>,<ϑ>)
or
O(<ρ_x>,<ρ_y>,<ρ_z>): Orientation with the function O(...) and the polar angles φ, ϑ of the orientation vector (programming absolute in degrees) or with the Cartesian components ρ_x, ρ_y, ρ_z of the orientation vector (programming absolute, the normalization of the components to 1 is performed automatically within the NC).
Example: G1 x10 y50 z30 O(90,90)
G1 x10 y50 z30 O(0,1,0)

ROTAX($<u_x>,<u_y>,<u_z>$) O($<\beta>$):
or
ROTAX($<\varphi_u>,<\vartheta_u>$) O($<\beta>$):

ROTAX(..) defines the orientation of the axis of rotation around which the orientation vector rotates. The direction may be specified in Cartesian vector components u_x, u_y, u_z or in polar coordinates φ_u, ϑ_u. The programming of ROTAX(..) is only possible absolutely.

O(..) defines angle β, around which the start orientation vector rotates around the axis of rotation. Programming of O(..) is possible only incrementally and in degrees.

β may have freely defined values, i.e. several rotations are possible too.

Example: ROTAX(0,45) O(720)

The movement of the orientation vector is carried out as rotation of the orientation vector around a **programmed axis of rotation**, or if ROTAX(..) and O(..) have not been programmed, around an **internally calculated axis of rotation**.

Example: G1 x10 y50 z30 O(90,90)
O(100,80)
O(70.100)
...

Linear orientation movement

The **orientation of the orientation vector** can be programmed using one of the following 2 alternatives:

phi$<\varphi>$ theta$<\vartheta>$: Orientation with polar angles φ, ϑ (polar angle names “phi” und “theta” are determined in MP 7080 00010 [4..5]). Programming is possible absolutely/incrementally and in degrees.

Example: G1 x10 y50 z30 phi90 theta90

O($<\varphi>,<\vartheta>$)
or
O($<\rho_x>,<\rho_y>,<\rho_z>$): Orientation with the function O(...) and the polar angles φ, ϑ of the orientation vector (programming absolute in degrees) or with the Cartesian components ρ_x, ρ_y, ρ_z of the orientation vector (programming absolute, the standardization of the components to 1 is performed automatically within the NC).

Example: G1 x10 y50 z30 O(90,90)
G1 x10 y50 z30 O(0,1,0)

Tensor orientation (TCS orientation)

The column vectors $\vec{e}^{\,t}_x = \begin{pmatrix} o_{11} \\ o_{21} \\ o_{31} \end{pmatrix}$ $\vec{e}^{\,t}_y = \begin{pmatrix} o_{12} \\ o_{22} \\ o_{32} \end{pmatrix}$ $\vec{e}^{\,t}_z = \begin{pmatrix} o_{13} \\ o_{23} \\ o_{33} \end{pmatrix}$ of the orientation tensor do not have to be provided standardized to 1. Standardization to ”1” is performed automatically within the NC.

The **TCS orientation** can be programmed using one of the following 5 alternatives:

phi<φ> theta<ϑ> psi<ψ>:

Orientation with Eulerian angles φ, ϑ, ψ (Eulerian angle names “phi”, “theta” and “psi” are determined in MP 7080 00010 [4..6]. Programming is possible absolutely/incrementally and in degrees.
Example: G1 x10 y50 z30 phi90 theta90 psi45

Ox(<o_{11}>,<o_{21}>,<o_{31}>)
Oy(<o_{12}>,<o_{22}>,<o_{32}>)
Oz(<o_{13}>,<o_{23}>,<o_{33}>)
or
Ox(<φ_x>,<ϑ_x>)
Oy(<φ_y>,<ϑ_y>)
Oz(<φ_z>,<ϑ_z>):

Ox(..) defines the direction of the x coordinate of the TCS in the reference coordinate system.
The direction may be specified in Cartesian vector components Ox(<o_{11}>,<o_{21}>,<o_{31}>) or in polar coordinates Ox(<φ_x>,<ϑ_x>).

The definition applies analogously to the column vectors Oy(..) and Oz(..).
It is only permitted to program 2 of the three TCS coordinates. They do not need to stand perpendicular on each other because one of them is corrected internally to 90 degrees. Programming is only possible absolutely.
Example:
G1 x10 y50 z30 Ox(1,0,0) Oy(0,0.707,–0.707)

ROTAX(<u_x>,<u_y>,<u_z>) O(<β>)
or
ROTAX(<φ_u>,<ϑ_u>) O(<β>):

ROTAX(..) defines the orientation of the axis of rotation around which the orientation tensor rotates.
The direction may be specified in Cartesian vector components u_x,u_y,u_z or in polar coordinates φ_x,ϑ_x. The programming of ROTAX(..) is only possible absolutely.
O(..) defines angle β, around which the start orientation vector rotates around the axis of rotation. Programming of O(..) is possible only incrementally and in degrees.
β may have freely defined values, i.e. several rotations are possible too.

Example: ROTAX(0,45) O(720)
ROTAX(1,0,1) O(720)

☞ **The individual parameters are shown in detail in the DIN programming instructions of the PNC.**

Relevant MACODA parameters (working range coordinates):

The names of all six working range coordinates are defined for each channel in a MACODA parameter.

7080 00010: Designation of the working range coordinates:
[1..3]: names of the linear coordinates (position coordinates)
[4..6]: names of the orientation coordinates

7010 00030: strict coordinate classifications are assigned to the coordinates. The position coordinates are assigned the classifications 1, 2 and 3, the 3 orientation coordinates are assigned the classification 999 (neutral).

1030 00110 through 1030 00150:
the vector and tensor orientation can only be performed in case of an active axis transformation.
An axis kinematics is defined as axis transformation in the MACODA parameters.
Each of these MACODA parameters exists five-fold, so that it is possible to configure up to five axis kinematics.
For a description of the parameters, please refer to Chapter 5.8, Section “5 axis and 6 axis transformation” in the present manual.

Relevant CPL functions (working range coordinates):

CPOS(..): The CPOS function provides the value programmed last for a (synchronous) coordinate. The coordinate may be a working range coordinate or a pseudo coordinate.
Working range coordinates can only be addressed via their logical coordinate name or their logical coordinate index.
Pseudo coordinates can additionally be addressed via the physical axis number or the physical axis name of the assigned axis. Thus it is possible to call a pseudo coordinate with a strict index or name even if its logical arrangement on the channel may be modified due to an exchange of axes.
If a physical axis number or a physical axis name is given when calling CPOS, and this axis is not linked to a pseudo coordinate, a runtime error will occur.
Only coordinates of the own channel can be requested using CPOS.

Working range coordinate programming

AXO(..): The AXO function provides the G92 shift programmed last for a coordinate. With respect to the call parameters, AXO behaves in the same manner as CPOS.

WPOS(..): The WPOS function provides the current workpiece position (WCS) for a coordinate. With respect to the call parameters, WPOS behaves in the same manner as CPOS, with the extension that WPOS also allows for the access to coordinates not pertaining to the channel.

MPOS(..): The MPOS function provides the current machine zero point related position (ACS(0)) for an axis. The axis can be requested via its logical/physical name or its logical/ physical index. MPOS allows for access to axes not pertaining to the channel.
MPOS cannot be programmed with coordinate names, e.g. MPOS("X") (ACS(0) $\neq$ BCS, see Section 5.3).

PPOS(..): Cannot be used in case of working range coordinates. Substitute function: see CPROBE.

CPROBE(..): In case of working range coordinates, the CPROBE function replaces the PPOS function!
The precondition for CPROBE is that the probe inputs have been taken over by all axes of the channel (PPOS has the option of excluding individual axes from measurement via MACODA). The axis values taken over are converted to program coordinates with the aid of the transformation chain activated last. CPROBE supplies the value for one coordinate.
With respect to the call parameters, CPROBE behaves in the same manner as CPOS.

FXC(..): Calling FXC(i) until now supplies the total value of the table axis ZS, external axis ZS and program coordinate shift for an axis.
Since the program coordinate shift refers to coordinates, an incompatible change is necessary.
FXC(i) provides the total of the axis zero shifts for an axis.

COF(..),
DPC(..),
SCL(..): The functions refer exclusively to coordinates. Only coordinates of the own channel can be requested.

PROBE(..),
APOS(..),
SPOS(..): The PROBE, SPOS and APOS functions continue to provide axis values. No modification is required here. Only the probe value for an axis of the own channel can be checked using PROBE.

AXP(..): The AXP function needs to be extended so that coordinate names are expanded.

MCODS(..): In preparation for use with working range coordinates.

5.7.4 Assigning

Preconditions:

- Assigned axis transformation (see Section 5.8.1 and 5.8.2) must be available for the relevant machine.

Example:
Working range coordinate programming for a 6 axis transformation

6 axis machine with 6 axis transformation type **3333301** of the PNC. It allows TCP programming via three linear coordinates and the tool orientation (TCS) by programming the 3 Eulerian angles φ (phi), ϑ (theta) und ψ (psi). The orientation movement is executed by means of a coordinate of rotation movement.

1. Determining the working range coordinates in MP 7080 00010. Example:
   Names of the linear coordinates
   [1]: x
   [2]: y
   [3]: z
   Names of the orientation coordinates
   [4]: phi
   [5]: theta
   [6]: psi

2. Make sure that the parameters required for 6 axis transformation type 3333301 are assigned the matching values of the 6 axis machine in the MP 1030 00110 through MP 1030 00150.
3. Keep the number of the parameter block (1, 2 .. or 5) containing your transformation data (e.g. 3rd parameter block) in mind.
4. The parameters are taken over with system control reset.
5. Test the working range coordinate programming as follows:
   - Activate the working range coordinate programming with Coord(3). 3 represents the number of the 3rd parameter block). At the same time, the axis transformation 3333301 is activated. The previously valid axis field (X,Y,Z,A,B,C) is replaced by the coordinate field (x,y,z,phi,theta,psi).
   - Check the display of the operating mode “AUTOMATIC” or “MANUAL” to see whether the axis addresses have been replaced by the working range coordinates.
   - Test the reference pose (defined in MACODA) of the axes.
   - Program a suitable test program for working range coordinates (orientation of the tool, orientation movement, positioning in space, ...).
   - The application is completed when the functions have been tested and no runtime error has been reported.

5.7.5 Activating

Precondition:

- The application has been executed successfully.

1. The working range coordinate programming is activated using Coord(<i>) in the NC program."i" is the number of the axis transformation.
2. The previously valid axis names (X,Y,Z,A,B,C) are replaced by coordinate names (x,y,z,phi,theta,psi).
3. All subsequent G functions have to be programmed with working range coordinates if they are involved in the positioning or orientation of the tool (see below).
4. Depending on the type of activated axis transformation, the NC activates either the vector or tensor orientation.

Examples: transformations

- 5 axis kinematics type: 3232201
  Coord(4)
  G1 **x**50 **y**100 **z**20 **phi**45 **theta**30 ; programming in working
  range coordinates x,y,z,phi,theta

- 6 axis kinematics type: 3333301
  Coord(3)
  G1 x50 phi45 theta30 psi190
  Coord(0)

Typical NC functions using coordinates:

- NC functions causing a movement:
  G00, G01, G02, G03, G05, G10, G11, G12, G13, G32, G73, G200, G202, G203
- Plane selection and pole programming: G17, G18, G19, G20
- Chamfers and transition arcs: G34, G134, G234
- Path compensation: G40, G41

Typical NC functions using axes:

- Functions to influence the axis dynamics:
  G06, G14, G608, G114, G177, G594, G595
- Axis zero shifts:
  G54-G59, G154-G159, G254-G259, G160, G260, G360
- Positioning type: G151, ACP, ACN, DC
- Axis transfer: G510, ..., G513, G515, G516
- Axis coupling: G581

5.7.6 Deactivating

Precondition:

- Deactivating causes no movement, i.e. the tool may be in contact with the workpiece.

1. The working range coordinate programming is switched off using Coord(0). At the same time, the corresponding axis transformation is deactivated.
2. The coordinate field (x,y,z,phi,theta,psi) is invalid and is replaced by the axis field (X,Y,Z,A,B,C).
3. Programming using axis names is permitted again.
4. Programming of tool orientations or tool orientation movements are only possible through rotary axis programming.

Examples:

- 5 axis kinematics type: **3232201**

`Coord(4)`	
`G1 x50 y100 z20 phi45 theta30`	programming in working range coordinates
`Coord(0)`	
`G1 X50 Y100 Z20 B45 C30`	programming in axis coordinates

- 6 axis kinematics type: **3333301**

`Coord(3)`	
`G1 x50 phi45 theta30 psi190`	programming in working range coordinates
`Coord(0)`	
`G1 X50 Y100 Z200 B10 C20 A90`	programming in axis coordinates

5.8 Axis transformations

Function:

If the program coordinates are no longer parallel with the machine axes, the NC will at first be unable to determine axis setpoint values directly for the machine axes. In this case, axis setpoint values would have to be generated by a programming system and stored in the program. The disadvantage would be that the program would only be running on one machine with a certain tool. The PNC avoids this restriction by means of axis transformation.

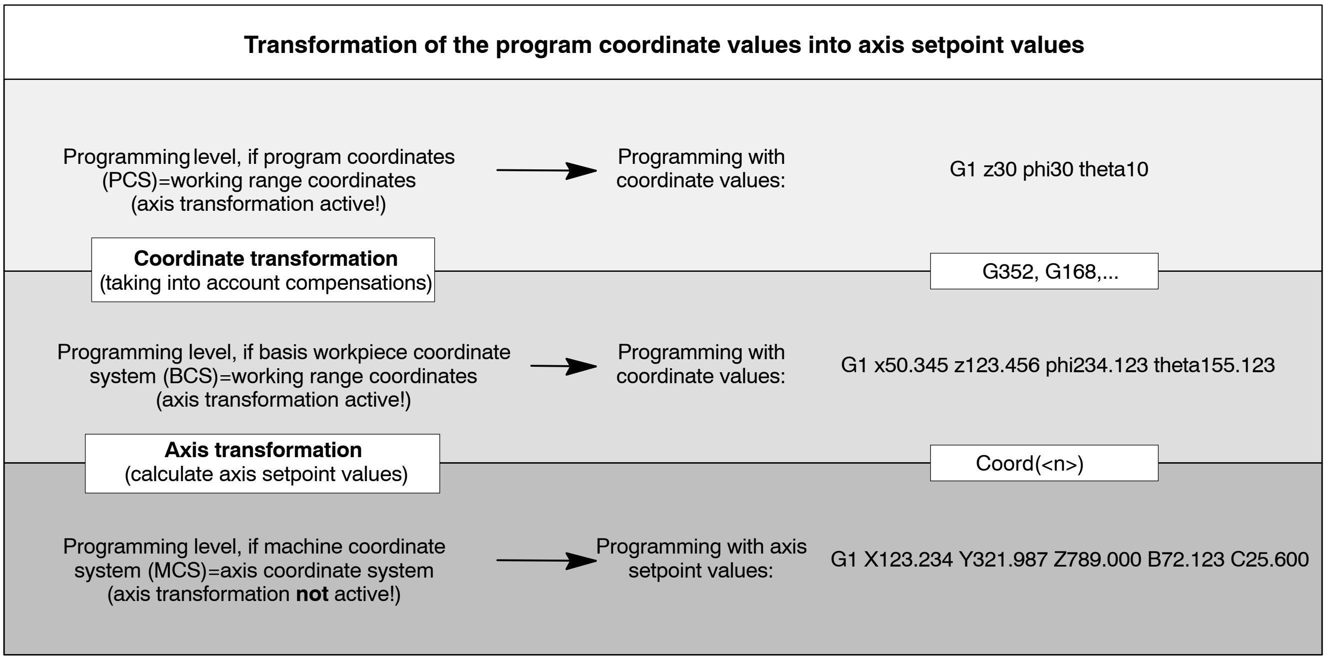
Program coordinates are interpreted as "working range coordinates". These coordinates refer to a machine-independent coordinate system positioned and oriented anywhere in space.
These working range coordinates are converted into the real axis system of the machine using a machine-specific axis transformation which takes into account or defines the geometric-kinematic relationship of machine and tool. In this process, the axis transformation calculates the shares of all real axes needed to approach each position in space.

For example, up to 5 axes may be needed to machine an "inclined plane" in order to ensure plane milling of a surface. In this case, only the block "G1 x100 phi 100 theta 45" is programmed with the transformed coordinates x, phi and theta. These have an effect on the linear axes X, Y and Z as well as on the rotary axes B and C.

The program coordinates have to be "processed" internally for the axis transformation.

1. If the programmed coordinates relative to the program coordinate system are shifted and/or rotated by compensations, the compensations are at first left out of account using a "coordinate transformation". This results in working range coordinates which refer to the basis workpiece coordinate system (BCS).
2. These working range coordinates are the initial value for the axis transformation which uses them to calculate the actual axis setpoint values.
   The axis transformation is also referred to as "reverse transformation".

Axis transformations



The way axis setpoint values are created from working range coordinates using axis transformation shows that the working range coordinates may have individually different effects on the physical axis configuration with each axis configuration. This can be realized by different settings of the axis transformations in MACODA.

At present, it is possible to store 5 different parameter blocks of an axis transformation per machine in MACODA **simultaneously** and to call these using Coord(<i=1..5>).

The PNC knows the following axis transformations:

- 5 axis transformation
- 6 axis transformation
- polar coordinate transformation

All axis transformations are distinguished through a type identification (MP 1030 00110). The type identification is a seven-digit number, where the digits have the following meaning:

Digit	1	2	3	4	5	6 and 7
	$\mathbf{K_l}$	$\mathbf{K_r}$	$\mathbf{A_l}$	$\mathbf{A_r}$	**O**	**NN**

Digit	Abbr.	Meaning	Options
1	K_l	Number of position coordinates	**1..3**: (e.g. x, y, z used in the NC programming)
2	K_r	Number of orientation coordinates	**1..3**: (e.g. phi, theta, psi used in the NC programming)
3	A_l	Number of linear axes (physically existing)	**0..**:limited by the number of system axes

Axis transformations

Digit	Abbr.	Meaning	Options
4	A_r	Number of axes of rotation (physically existing)	**0..**:limited by the number of system axes
5	O	Identification of the supported orientation movement	**0**: No orientation movement **1**: Linear orientation movement in the axes of rotation **2**: Vector rotation (coordinate of rotation movement) of the orientation vector **3**: Tensor rotation of the orientation (TCS).
6 and 7	NN	Consecutive numbering for internal purposes	**01 .. 99**: The number is type-related

The type identification shows which functions have an effect when the axis transformation is **selected**:

- The axis transformation with number **i** (from the **i**th parameter block in MACODA) is activated using the Coord(<i>) command if the configuration in MACODA has no incorrect entries.
- If the activated axis transformation supports the **vector or tensor orientation**, the corresponding NC functions for the orientation movement are activated.
- Programming of **coordinate names** is activated, i.e. the axes are switched over to coordinates.
- The axes involved in the axis transformation may not be programmed any more in case of active axis transformation.

  Example:

Coordinate names:	x	y	z	phi	theta	psi	U	V
Axis names:	X	Y	Z	A	B	C	U	V

  The axis transformation determines how many of the maximum 6 working range coordinates can be programmed.
  Working range coordinates consist of max. 3 translational “position coordinates” and of max. 3 rotary “orientation coordinates”, and their names can be freely set in MACODA (in the following, “x”, “y” und “z” will be used for position coordinates and “phi”, “theta” und “psi” for orientation coordinates).
  In the example, U and V are additional channel axes not involved in the axis transformation.

- As a result of switching over to axis transformation, the workpiece coordinate display jumps from ”axis positions” to ”coordinate values”. The axis names of the axis field change into coordinate names, as shown in the example above.

5.8.1 5 axis transformation

With the aid of 5 axis transformation, program coordinates are turned into axis positions of the axes involved.

Axis configuration

The 5 axis transformation realized in the PNC includes:

- 3 linear coordinates (e.g. x, y, z)
- 2 orientation coordinates ϑ, φ (e.g.. theta, phi)
- 3 linear axes (e.g. X, Y and Z)
- 2 axes of rotation (e.g. B and C)

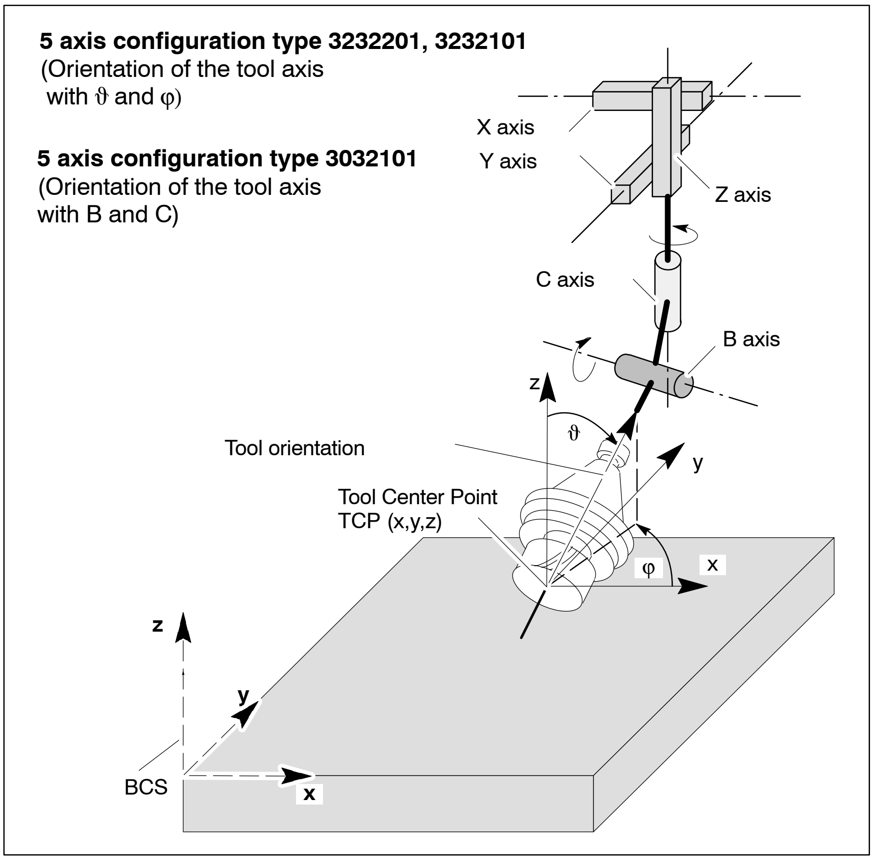
There are 3 types of 5 axis transformation:

- **Type 3232201**
  The initial values are the linear coordinates x, y, z and the orientation coordinates ϑ (theta) and φ (phi). it supports the vector orientation movement (rotation of the orientation vector $\vec{\rho}$) around an axis of rotation fixed in space, large circular movement of the vector tip. If a different value is programmed for ϑ and/or φ, the tool orientation will change.
  The orientation always rotates around the tool tip (TCP, tool center point). This means, it is assumed that the TCP is fixed and the tool axis rotates around this fixed point.
  All 5 axes may be involved in case of orientation changes. The orientation always points in direction of the tool holder.
  In addition, the orientation vector $\vec{\rho}$ can be programmed as spline. An online compensation for φ and ϑ is not possible.

- **Type 3232101**
  The initial values are the linear coordinates x, y, z and the orientation coordinates φ (phi) und ϑ (theta). It supports the orientation movement (straight line on an imaginary $\varphi - \vartheta$ plane).
  Both polar coordinates can also be programmed as spline. An online compensation is possible for all coordinates.
  The axis kinematics is identical with type 3232201. However, it does not support vector orientation, i.e. a constant φ (phi) is expected for all orientation movements.
  Switching this type on using Coord(<n>) activates the "Linear orientation movement" function.

Axis transformations

- **Type 3032101**
  supports a linear movement of the two axes of rotation. This transformation is basically identical with Type 3232201, however, it has **no** programmable orientation coordinates.
  The tool is aligned by direct programming of the axes of rotation B and C. The orientation takes place linear in the rotary axis positions.

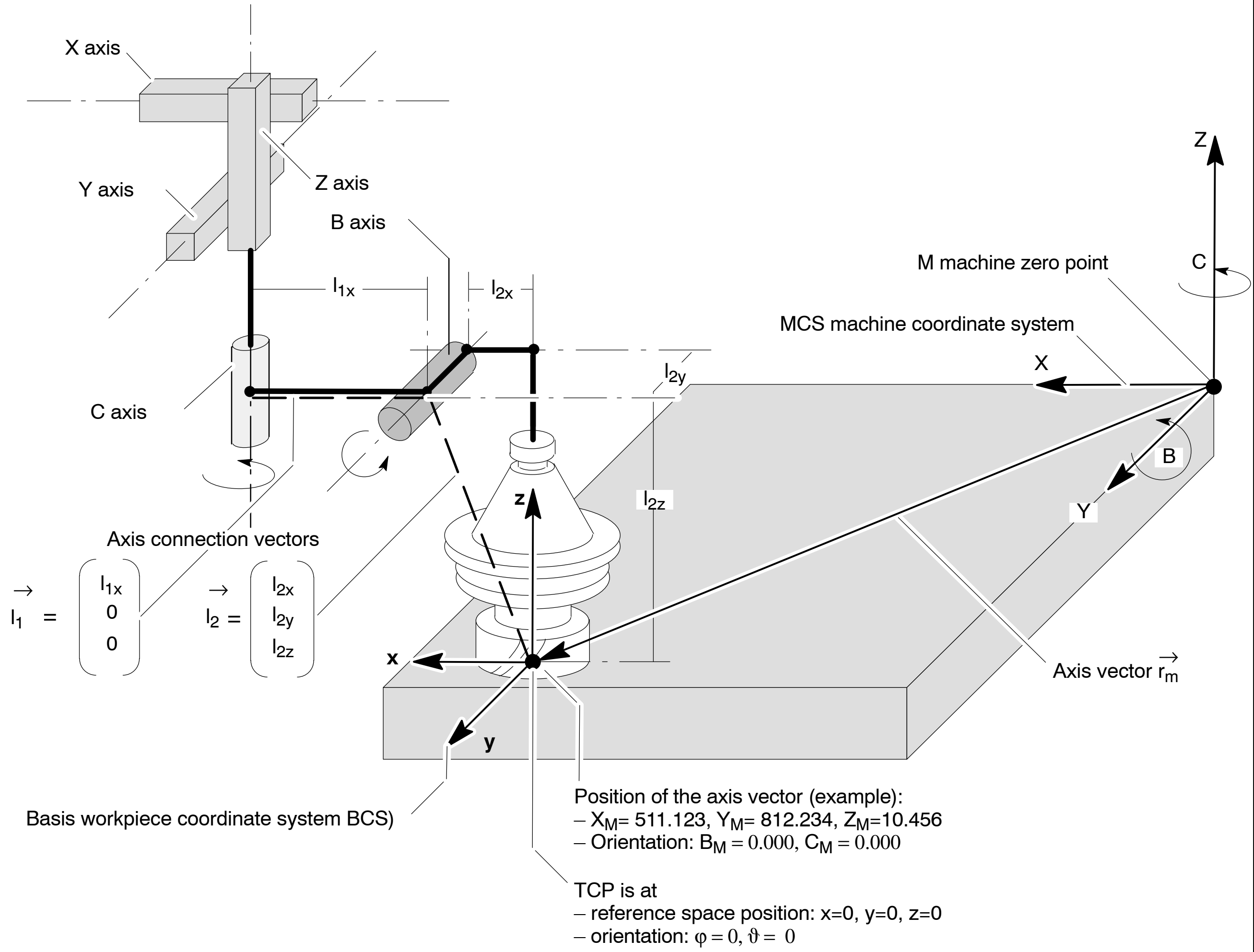


Axis transformations

Reference pose

In reference pose, the 5 axis configuration must have a certain position to establish a relationship between MCS and BCS.
This is exactly the case when the tool center point (TCP) is set at space position x, y, z=0 and the tool axis orientation corresponds to polar coordinates ϑ, φ = 0 degrees.

Reference pose of the 5 axis configuration



- In the figure above, l_{1x}, l_{2x}, l_{2y} und l_{2z} < is 0. The direction of the vectors is opposite (negative) to the BCS coordinate direction.
- The tip (TCP) of the rotation symmetrical tool (cutter or laser) is standing on the origin of the basis workpiece coordinate system (BCS) and the tool symmetry axis is aligned along the z coordinate. Axis C rotates the B axis body and the corresponding tool around the z coordinate.
- Axis B rotates the tool around the y coordinate of the BCS. The arrows at the axis bodies specify the positive direction of rotation. The linear axes (X, Y, Z) define a right-handed axis coordinate system positioned parallel with the BCS.

The **axis vector** $\vec{r}_m$ corresponds to the distance between the M machine zero point and the reference space position (zero point of the basis workpiece coordinate system).
The angles the rotary axes have to rotate in order to reach the reference orientation of ϑ, φ = 0 in the reference space position x,y,z = 0 are also relative to the M machine zero point.

The **axis connection vectors** $\vec{l}_1$ and $\vec{l}_2$ define the geometry of the 5 axis kinematics in the **reference pose** (see figure above):

- $\vec{l}_1$ is the axis connection vector from the C axis to the B axis.
  It **has to be** parallel with the X axis and therefore has no components in the y and z direction:

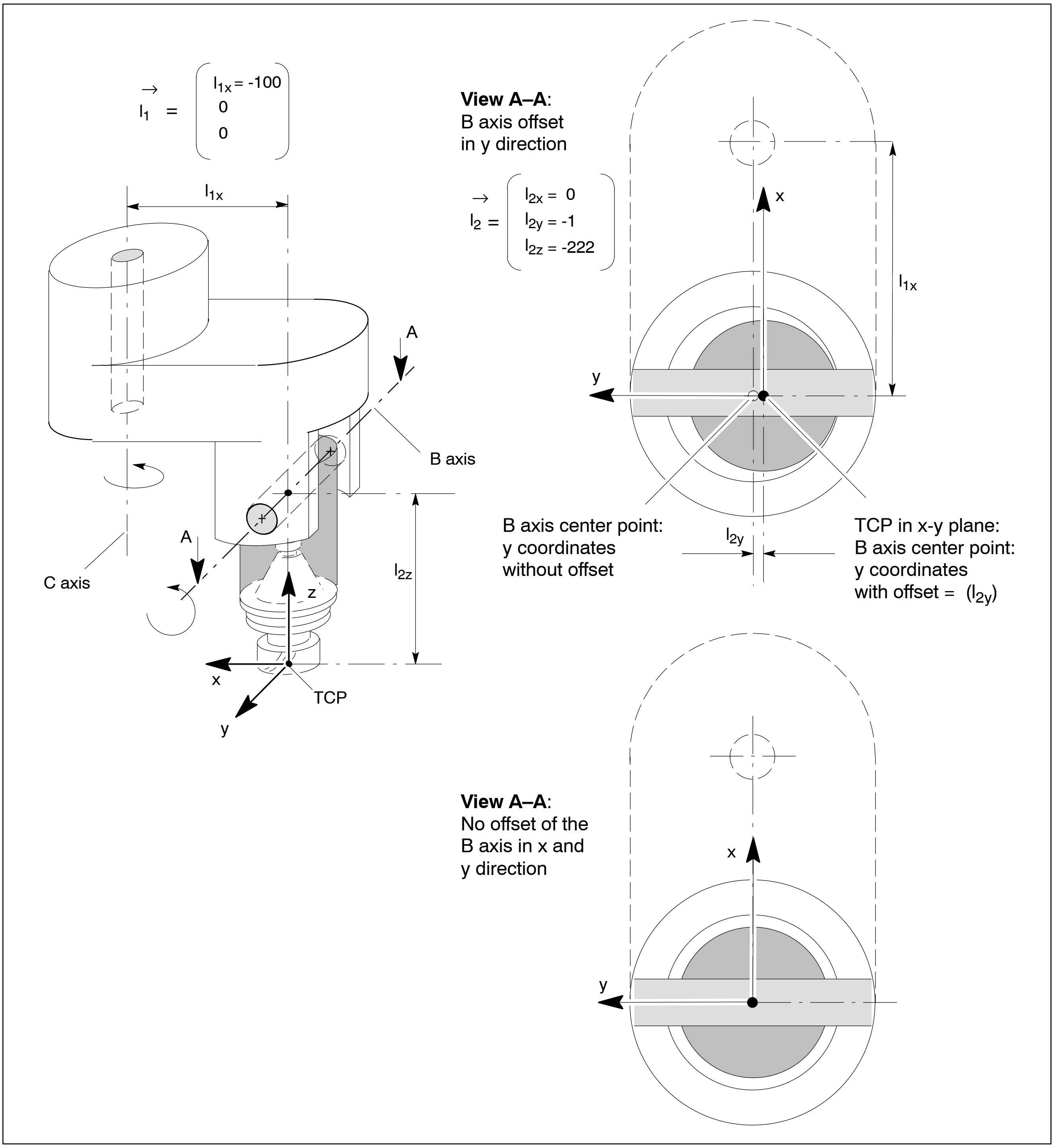
  $$\vec{l}_1 = \begin{bmatrix} l_{1x} \\ 0 \\ 0 \end{bmatrix}$$

- $\vec{l}_2$ is the continued vector of $\vec{l}_1$ to the zero point of the BCS.
  $\vec{l}_2$ may have up to 3 components unequal to 0. As a rule, l_{2z} is the distance from the B axis to the TCP. If the TCP and the center point of the B axis (= end point $\vec{l}_1$) are offset from each other in the x–y plane, the offset may be compensated with l_{2x} and l_{2y}. Therefore, the connection vector $\vec{l}_2$ is as follows:

  $$\vec{l}_2 = \begin{bmatrix} l_{2x} \\ l_{2y} \\ l_{2z} \end{bmatrix}$$

Axis transformations

Example: 5 axis rotating swiveling head with and without B axis offset



Special properties

- Feedrate: having activated the 5 axis transformation, there is a switchover to working range coordinate programming. The programmed feedrate (F) refers to the programmable position coordinates only, i.e. the F word is used to program the path speed of the tool center point (TCP).
  Additional orientation and pseudo coordinate movements do not change this path speed.
  Orientation and pseudo coordinate movement is guided along synchronously, i.e. the end position is reached simultaneously for all coordinates. The movement of the orientation and pseudo coordinates carried along, may, however, lead to an additional limitation of the path kinematics (maximum path speed and acceleration) because the limit values of all axes involved in the movement are monitored.
- Axes of rotation B and C may be endless axes as well as rotary axes.

Relevant NC functions:

Coord(<i>): Using i=1..5, one of the five possible 5 axis transformations defined in MP 1030 00110 is activated.

Coord(0): The active transformation is deactivated.

G0....: The NC functions or G codes programmable with working range coordinates may be programmed in case of active 5 axis transformation (see DIN Programming instructions).

Relevant MACODA parameters (MP):

1030 00110 through 1030 00150: Definition of the axis kinematics as axis transformation. Each of these MACODA parameters exists five-fold, so that it is possible to configure up to five axis kinematics.

☞ **All values to be entered refer to the reference pose.**

1030 00110 : Transformation type:
3232201 or 3032101 has to be entered for the 5 axis transformation.

1030 00120 : System axes of the transformation (1–64):
Defines the system axes involved in the transformation.
Please note for both types of transformation:

[1] System axis number of X
[2] System axis number of Y
[3] System axis number of Z
[4] System axis number of C
[5] System axis number of B
[6..8] not relevant

1030 00130: Reference pose axis positions:
Defines the distances of the reference space position to the M machine zero point for the linear axes.
For the axes of rotation, the respective angles they traverse from the M machine zero point to the reference space position are set.
[1] Position of the 1st linear axis
[2] Position of the 2nd linear axis
[3] Position of the 3rd linear axis
[4] Position of the 1st rotary axis (farthest away from the tool (TCP))
[5] Position of the 2nd rotary axis (at the tool)
[6..8] not relevant

1030 00140 : Length and angle parameters:
Defines the length vectors $\vec{l}_1$ and. $\vec{l}_2$
All values in mm.
[1] l_{1x}
[2] no meaning
[3] no meaning
[4] l_{2x}
[5] l_{2y}
[6] l_{2z}
[7..8] no meaning.

Axis transformations

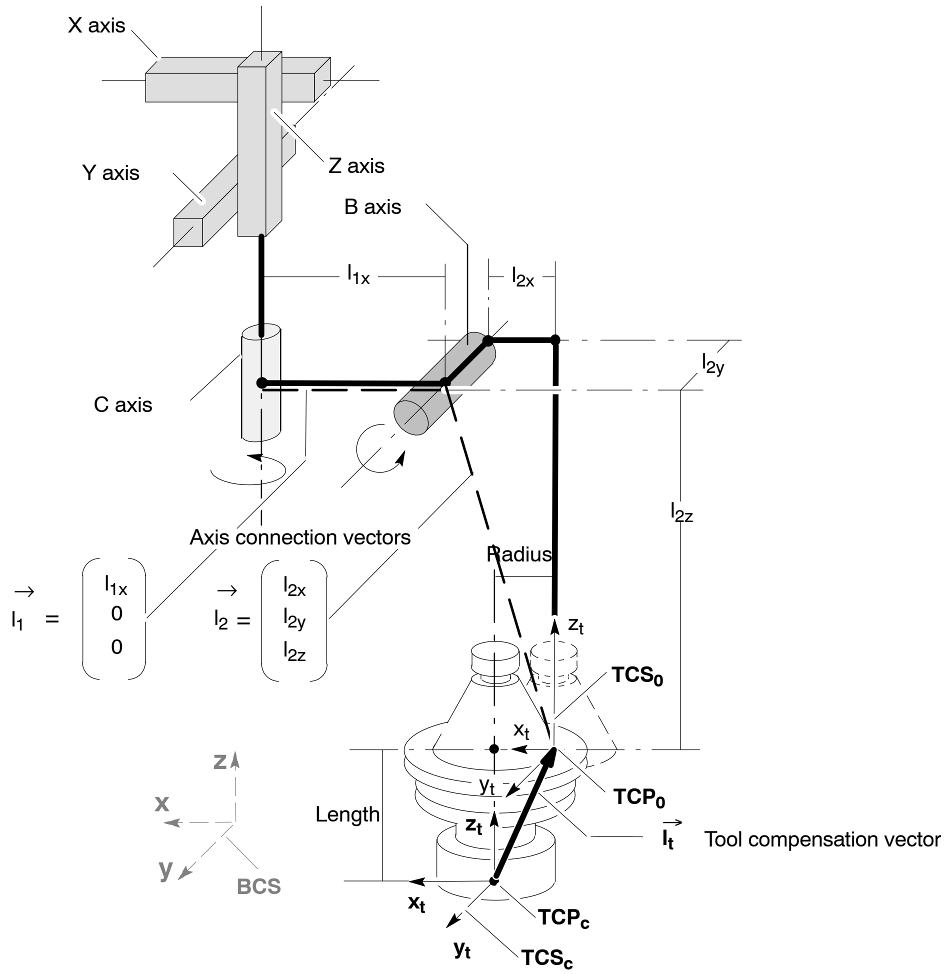
Tool compensation for the 5 axis transformation

All 5 axis transformations support tool compensations. The position of the TCS tool coordinate system with regard to tool compensations is explained below.

- In the **reference pose** the TCS_0 and the BCS are identical. The tool symmetry axis runs parallel with the BCS z coordinate. In the TCS_0 tool coordinate system, no tool compensation is active.
- In this context, a **tool compensation** (length, radius) means a spatial shift of the TCP along a compensation vector $\vec{l_t}$ from TCS_c to TCS_0 (positive L3 value extends the tool).

Tool compensation as spatial shift of the TCP along the vector $\vec{l_t}$

(Type 3232201, 3032101)



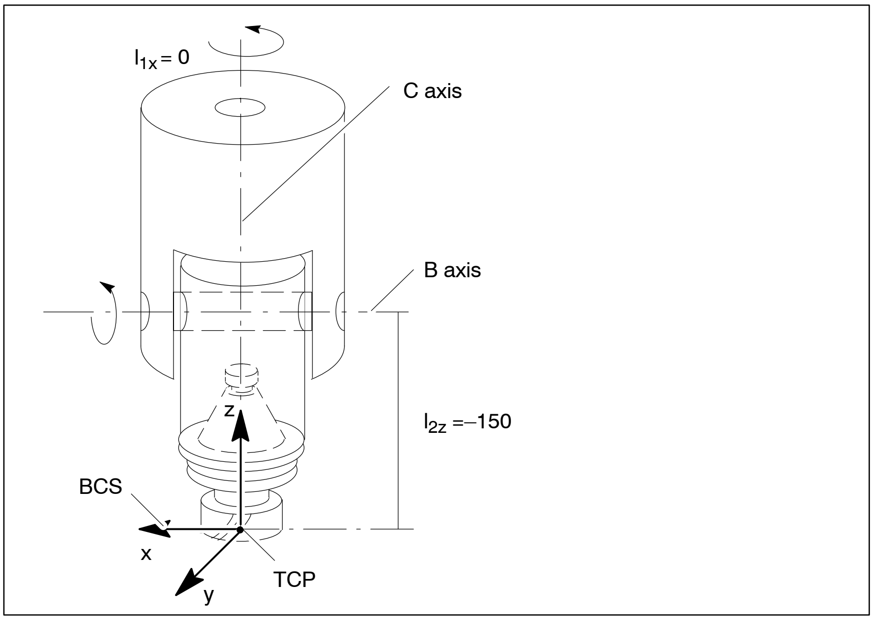
Assigning the 5 axis transformation

Transformation types 3232201 and 3032101 are used as examples.

☞ **All values to be entered refer to the reference pose.**

1. Set value 3232201 as transformation type in MP 1030 00110 (no. 1) and value 3032101 in MP 1030 00110 (no. 2).
2. The system axis numbers must be entered in MP 1030 00120 (no. 1 and 2). Please make sure that the rotary axes are entered in the order of their appearance from the tool holder to the tool tip.
3. Appropriate vectors $\vec{l_1}$ and $\vec{l_2}$ have to be determined. This can be done by measuring or on the basis of the construction drawings. Length and offset values have to be set in MP 1030 00140 [1,4,5,6] both for no. 1 and no. 2.
4. Establish the axis zero position: Approach reference pose. The axis positions in the reference pose are set in parameter 1030 00130 [1..5] both for no. 1 and no. 2.
5. Take over parameter using system control reset.
6. Test the 5 axis transformation in an NC program for functionality using Coord(1) and Coord(2). If no runtime error occurs, the parameters are consistent.

Example:5 axis transformation for a swiveling rotating milling head.



Settings in one out of five MACODA parameter blocks:

1030 00110 Transformation type: 3232201 or 3032101

1030 00120 System axes of the transformation (1-164):
[1] 1 corresponds to axis X
[2] 2 corresponds to axis Y
[3] 3 corresponds to axis Z
[4] 5 corresponds to axis C
[5] 4 corresponds to axis B

1030 00130 Reference pose axis positions:
[1] 0
[2] 0
[3] 0
[4] 0
[5] 0

1030 00140 Length and angle parameters:
[1] 0
[4] 0
[5] 0
[6] –150

Activating the 5 axis transformation

Precondition:

- Configured 5 axis kinematics in MP 1030 00110 through MP 1030 00150.
- All NC functions and G codes used in the program must be checked for their applicability with the selected transformation using the table below.

1. Activating the transformation in the NC program using Coord(<i>).

   Example: 5 axis transformation type: 3232201

   ```
   Coord(4)
   G1 x50 y20 z40 phi45 theta30
   Coord(0)
   ```

Deactivating the 5 axis transformation

1. Deactivate a 5 axis or 6 axis transformation in the NC program using Coord(0).

5.8.2 6 axis transformation

Axis configuration

6 axis transformation is used in machines with tools that

- have to execute translational and rotary path movements in space simultaneously, and in addition to the path movement
- execute orientation movements which align the tool relative to every point of the path.

With the aid of 6 axis transformation, program coordinates are turned into axis positions of the axes involved.

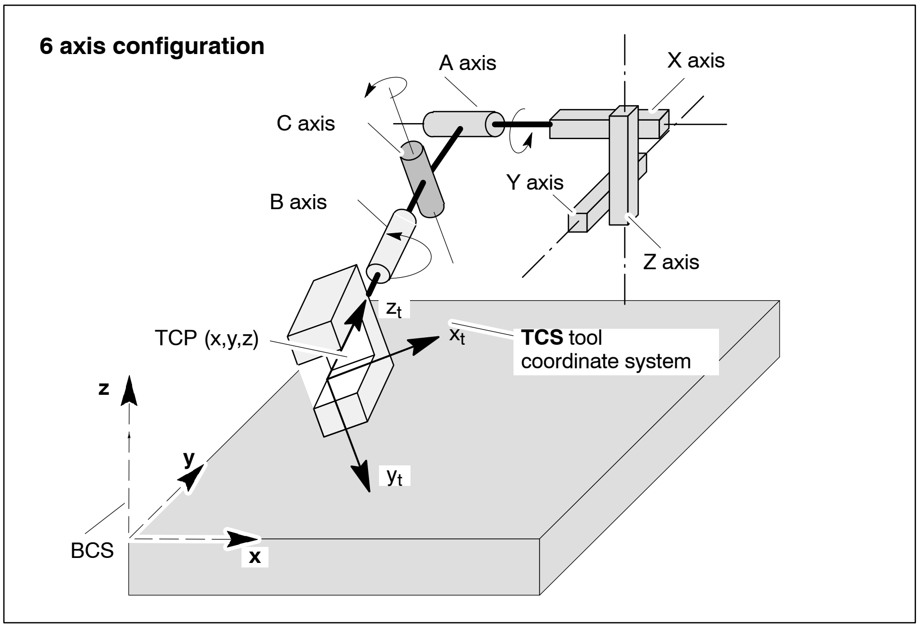
The 6 axis transformation realized in the PNC includes:

- 3 linear coordinates (e.g. x, y, z)
- 3 orientation coordinates ϑ, φ, ψ (e.g. theta, phi, psi),
- 3 linear axes (e.g. X, Y and Z)
- 3 axes of rotation (e.g. A, B and C)

There are 2 versions of 6 axis transformation:

- **Typ 3333301** allows TCP programming via three linear coordinates and the tool orientation (TCS) by programming the 3 Eulerian angles φ (phi), ϑ (theta) and ψ (psi) (for information on Eulerian angles, see Section 5.5).
  The orientation movement is performed as TCS rotation around an axis of rotation fixed in space.

- **Type 3033101** supports TCP programming via three linear coordinates and tool orientation by programming the three axes of rotation. The orientation movement is executed linearly in the rotary axis positions.



Axis transformations

Type and position of the axes of rotation

The axes of rotation are arranged one after the other on the tool and stand on each other perpendicularly. In the reference pose, they rotate the tool around one of the linear basis coordinates x, y or z in positive sense of rotation.
The positions of the axes of rotation are referred to as α_i, α_j, α_k . The indexes i, j and k (1, 2 or 3) specify the type of axis.

The type is defined as follows:

Axis of rotation type	Explanation	Axis classification in MACODA 7010 00030
1	The axis of rotation rotates around x. Its position is referred to as α_1.	100
2	The axis of rotation rotates around y. Its position is referred to as α_2.	200
3	The axis of rotation rotates around z. Its position is referred to as α_3.	300

Order of the axes of rotation relative to the tool:

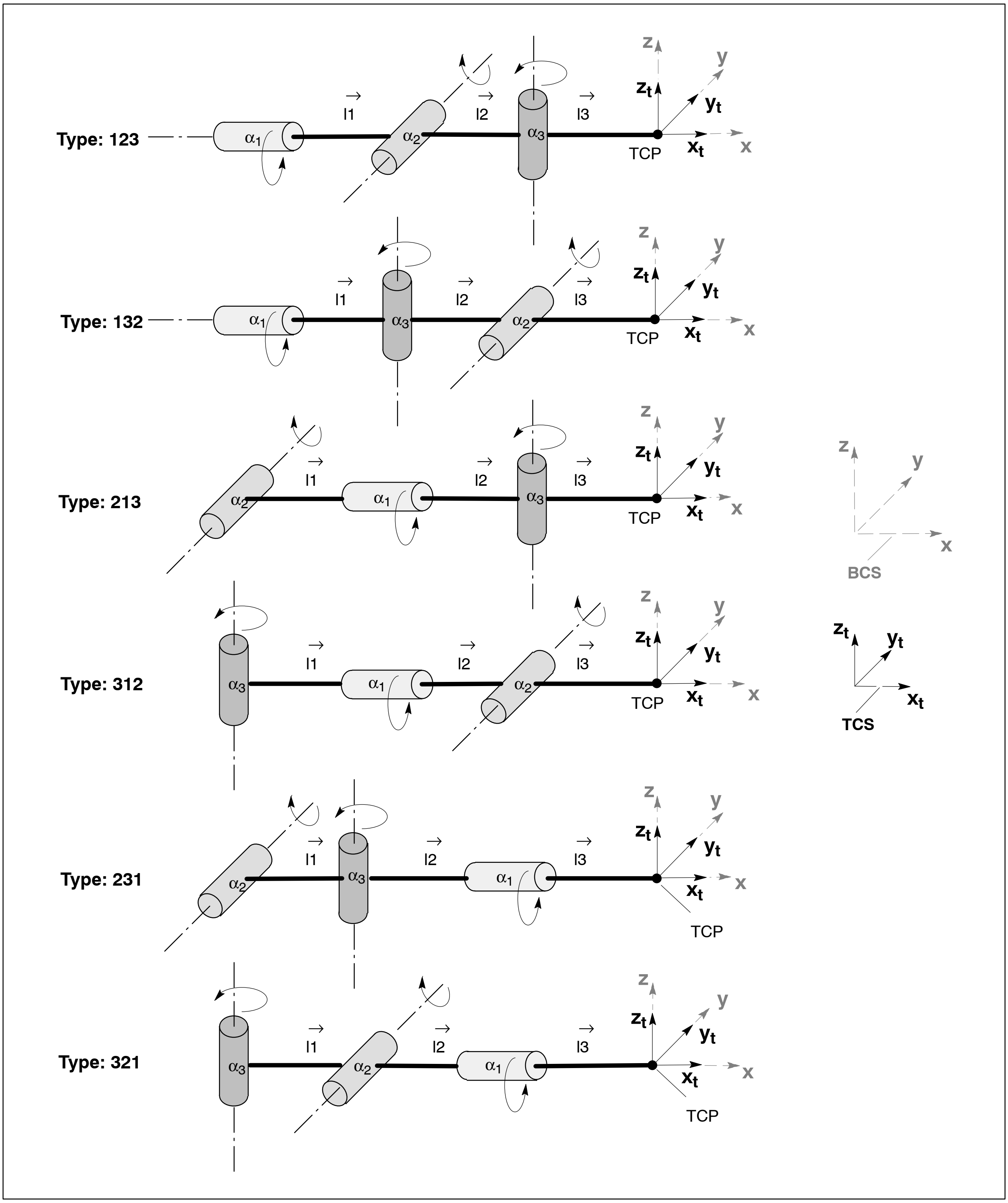
The axes of rotation type 1, 2 or 3 can be arranged on the tool in **six** different orders.

This results in the following **subconfigurations:**

Subconfiguration type	Position of the 1st axis of rotation (farthest away from the TCP)	Position of the 2nd axis of rotation	Position of the 3rd axis of rotation (close to the TCP)
123	α_1	α_2	α_3
132	α_1	α_3	α_2
213	α_2	α_1	α_3
231	α_2	α_3	α_1
312	α_3	α_1	α_2
321	α_3	α_2	α_1

Example: Type 132 = the first axis of rotation rotates around the x coordinate of the BCS, the second axis of rotation around the z coordinate and the third axis of rotation around the y coordinate.

Subconfigurations of the axes of rotation arrangement:



Axis transformations

Axis connection vectors

The geometry to the axis kinematics is defined by the three axis connection vectors $\vec{l}_1$, $\vec{l}_2$ and $\vec{l}_3$. The vectors may have freely defined x, y and z components in relation to the BCS.

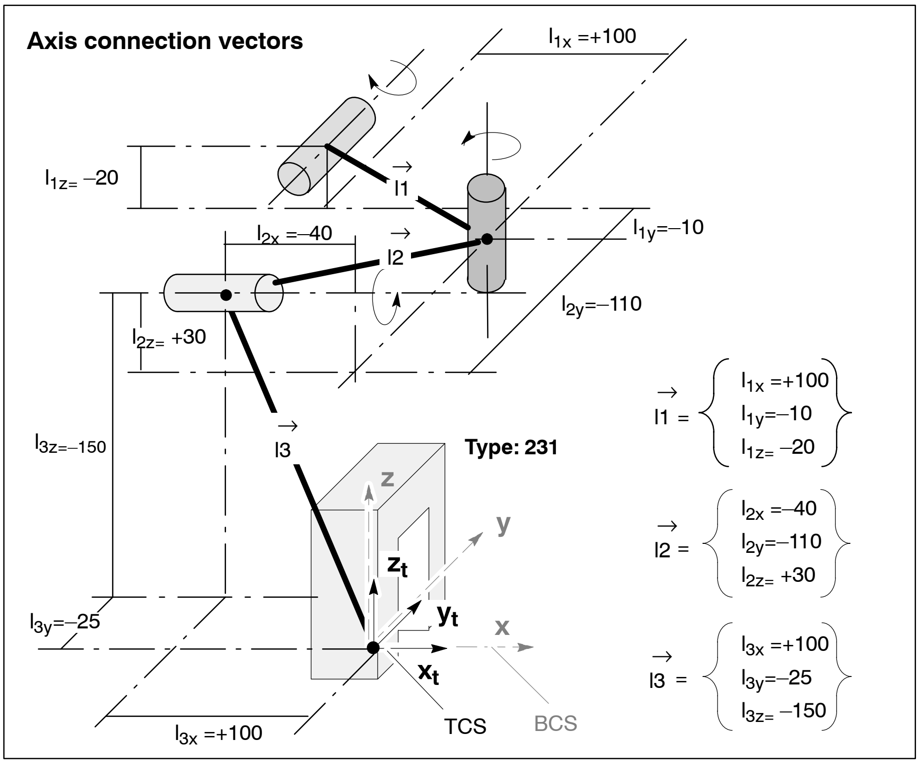
The connection vector $\vec{l}_i$ (i=1,2,3) is as follows:

$$\vec{l}_i = \begin{bmatrix} l_{ix} \\ l_{iy} \\ l_{iz} \end{bmatrix}$$

$\vec{l}_1$ is located between the 1st and 2nd axis of rotation. The 1st axis of rotation is the rotary axis located farthest away from the tool (TCP). It is followed by the 2nd axis of rotation which is connected with the 3rd axis of rotation via $\vec{l}_2$. $\vec{l}_3$ connects the 3rd axis of rotation with the tool (TCP).

The l_{ix}, l_{iy} and l_{iz} components of the axis connection vectors are set in MP 1030 00140.

Location of the axis connection vectors and their x, y, and z components:



Axis transformations

Reference pose

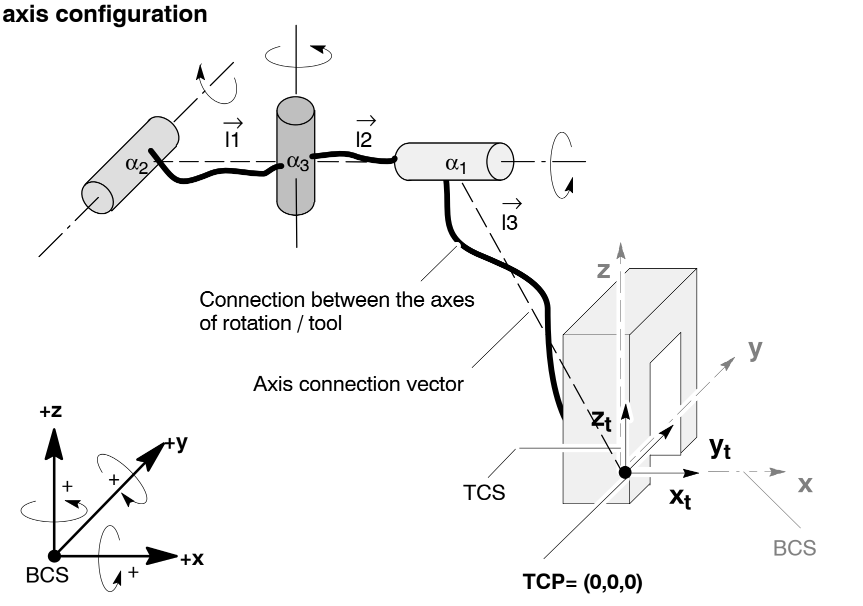
The **reference pose** is significant for the configuration parameters of the axis kinematics. The reference pose is characterized by the following properties:

- The TCP is at **space position (0,0,0)** in the BCS (zero point of the basis workpiece coordinate system).
- The coordinates (x_t, y_t, z_t) of the tool coordinate system (TCS) are **parallel** or **anti-parallel** with the coordinates of the basis workpiece coordinate system.
- The axes of rotation are positioned so that they rotate around x, y and z of the BCS in accordance with their respective type. They are arranged **one after the other** on the tool and stand on each other **perpendicularly**.

The sense of rotation of the axes of rotation is mathematically positive if one looks from one coordinate vertex in the direction of the coordinate origin and the sense of rotation is counter-clockwise.

The linear axes (X, Y, Z) define a right-handed axis coordinate system positioned parallel with the BCS.

Reference pose of the 6 axis configuration



Axis transformations

Reference orientations of the tool coordinate system (TCS)

The **orientation** of the TCS (x_t, y_t, z_t) relative to the BCS (x, y, z) in the reference pose can be configured with MP 1030 00150.

The following conditions apply to the reference orientations of the TCS:

- the main directions (x_t, y_t, z_t) of the TCS have to be parallel or anti-parallel with the main directions (x, y, z) of the BCS.
- The main directions of the TCS and BCS have to result in a right-handed right-angle coordinate system.

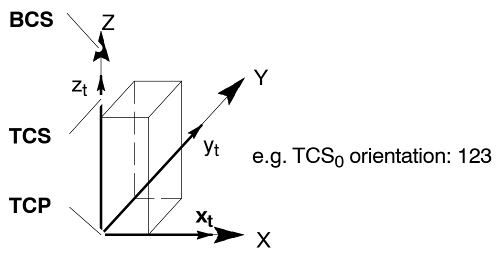
The presetting of MP 1030 00150 (reference orientation of the tool coordinate system) defines a TCS_0, which coincides with the orientation of the BCS in the reference pose of the axis transformation.
[1] 1
[2] 2
[3] 3

Deviating from this presetting, it is possible to define an orientation for the $\mathbf{TCS_1}$ in MP 1030 00150 which occupies one of the 24 reference orientations in relation to the BCS.

To give a better **overview**, the 24 possible orientations of the TCS_1 are subdivided as follows:

The x_t coordinate of the TCS_1 is parallel or anti-parallel with one of the 3 main directions x, y or z of the BCS. This results in 8 different TCS_1 orientations for **each** of the 3 main directions.

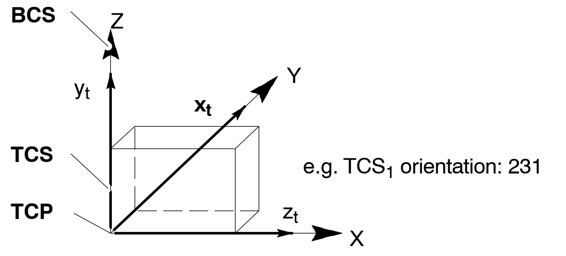
$\mathbf{x_t}$ is located along the **x coordinate** of the BCS:



TCS main direction x_t —> x coordinate of the BCS

			Reference orientation of the tool coordinate system (TCS) in MP 1030 00150		
xt [1] is located along the BCS coordinate**:**	yt [2] is located along the BCS coordinate:	zt [3] is located along the BCS coordinate:	[1]	[2]	[3]
x	y	z	**1**	**2**	**3**
x	z	–y	1	3	–2
x	–z	y	1	–3	2
x	–y	–z	1	–2	–3
–x	z	y	–1	3	2
–x	–y	z	–1	–2	3
–x	y	–z	–1	2	–3
–x	–z	–y	–1	–3	–2

$\mathbf{x_t}$ is located along the **y coordinate** of the BCS:

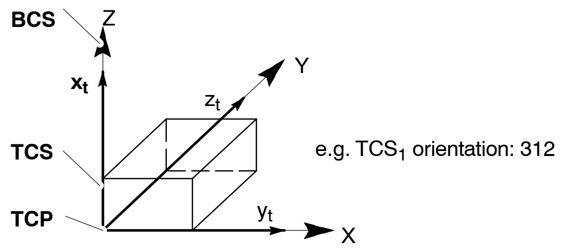


TCS main direction x_t —> y coordinate of the BCS

Reference orientation of the tool coordinate system (TCS) in MP 1030 00150

xt [1] is located along the BCS coordinate**:**	yt [2] is located along the BCS coordinate**:**	zt [3] is located along the BCS coordinate:	[1]	[2]	[3]
y	z	x	**2**	**3**	**1**
y	x	–z	2	1	–3
y	–x	z	2	–1	3
y	–z	–x	2	–3	–1
–y	x	z	–2	1	3
–y	–z	x	–2	–3	1
–y	z	–x	–2	3	–1
–y	–x	–z	–2	–1	–3

$\mathbf{x_t}$ is located along the **z coordinate** of the BCS:



TCS main direction x_t —> z coordinate of the BCS

Reference orientation of the tool coordinate system (TCS) in MP 1030 00150

xt [1] is located along the BCS coordinate**:**	yt [2] is located along the BCS coordinate:	zt [3] is located along the BCS coordinate:	[1]	[2]	[3]
z	x	y	**3**	**1**	**2**
z	y	–x	3	2	–1
z	–y	x	3	–2	1
z	–x	–y	3	–1	–2
–z	y	x	–3	2	1
–z	–x	y	–3	–1	2
–z	x	–y	–3	1	–2
–z	–y	–x	–3	–2	–1

Relevant NC functions:

Coord(<i>): Using i=1..5, one of the five possible 6 axis transformations defined in MP 1030 00110 is activated.

Coord(0): The active transformation is deactivated.

G0....: The NC functions or G codes programmable with working range coordinates may be programmed in case of active 6 axis transformation (see DIN Programming instructions).

Properties:

- Feedrate: in case of active 6 axis transformation, the programmed feedrate (F) refers to the programmable position coordinates only, i.e. the F word is used to program the path speed of the tool center point (TCP).
  Additional orientation and pseudo coordinate movements do not change this path speed. The orientation and pseudo coordinate movement is guided along synchronously, i.e. the end position is reached simultaneously for all coordinates.
  The movement of the orientation and pseudo coordinates carried along, may, however, lead to an additional limitation of the path kinematics (maximum path speed and acceleration) because the limit values of all axes involved in the movement are monitored.
- Axes of rotation A, B and C may be endless axes as well as rotary axes.

Relevant MACODA parameters (MP):

1030 00030: Axis classification of axes of rotation must be set in accordance with the explanations given on page 5–91.

1030 00110 through 1030 00150: Definition of the axis kinematics as axis transformation. Each of these MACODA parameters exists five-fold, so that it is possible to configure up to five axis kinematics.

☞ **All values of the following parameters to be entered refer to the reference pose.**

1030 00110 : Transformation type:
3033101 or 3333301 has to be entered for the 6 axis transformation.

1030 00120 : System axes of the transformation (1-64):
Defines the system axes involved in the transformation.
Please note for both types of transformation:

[1] System axis number of the 1st linear axis
[2] System axis number of the 2nd linear axis
[3] System axis number of the 3rd linear axis
[4] System axis number of the 1st rotary axis (farthest away from the tool (TCP))
[5] System axis number of the 2nd rotary axis
[6] System axis number of the 3rd rotary axis (at the tool)
[7, 8] not relevant

From the order of rotary axes designated in parameters [4], [5] and [6] and the axis classifications of the rotary axes, the NC establishes the **subconfiguration**.

1030 00130: Reference pose axis positions:
The 6 axis positions resulting from the reference pose have to be entered here:

[1] Position of the 1st linear axis
[2] Position of the 2nd linear axis
[3] Position of the 3rd linear axis
[4] Position of the 1st rotary axis (farthest away from the tool (TCP))
[5] Position of the 2nd rotary axis
[6] Position of the 3rd rotary axis (at the tool)
[7, 8] not relevant

1030 00140 : Length and angle parameters:
Defines the axis connection vectors $\vec{l}_1$, $\vec{l}_2$ and $\vec{l}_3$ by specifying the l_{ix}, l_{iy} and l_{iz} components.
All values in mm.
[1] l_{1x}
[2] l_{1y}
[3] l_{1z}
[4] l_{2x}
[5] l_{2y}
[6] l_{2z}
[7] l_{3x}
[8] l_{3y}
[9] l_{3z}
[10.0.16] no meaning.

1030 00150 : Reference orientation of the tool coordinate system:
Defines the orientation of the TCS_1 in relation to the BCS in the reference pose.
[1] Direction of x_t
[2] Direction of y_t
[3] Direction of z_t
[4..8] not relevant

Values: ± 1, $\pm 2, \pm 3$. The orientation of the TCS **must** result in a right-handed coordinate system.

Parameter 1030 00150 is not relevant for type 3033101.

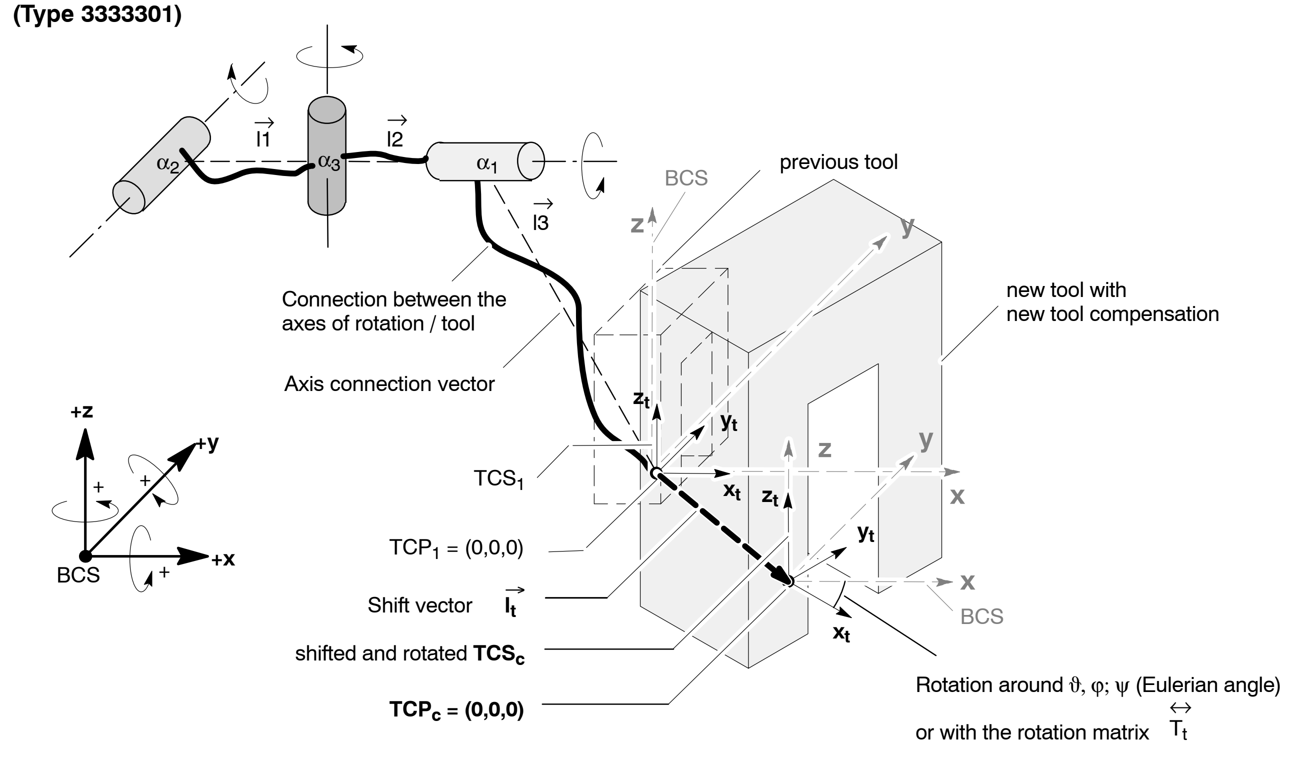
Axis transformations

Tool compensation for the 6 axis transformation type 3333301

The **TCS$_1$** can be shifted and rotated using tool compensations. A tool coordinate system **TCS$_c$** is thus created:

- The shift (compensation vector) of the **TCS$_c$** relative to the **TCS$_1$** is defined by $\vec{l}_t$.
- The orientation of the **TCS$_c$** in the coordinate system **TCS$_1$** is defined by the rotation matrix $\overleftrightarrow{T}_t$ or in the same manner by the three Eulerian angles ϑ, φ, ψ. The compensation vector and the Eulerian angles can be set e.g. using the CPL command “TC”.

Effect of tool compensation on TCS$_1$ —> TCS$_c$ (Type 3333301)



In this case, the tool compensation has the same effect as replacing the configuration vector $\vec{l}_3$ by $\vec{l}_t + \vec{l}_3$.

Axis transformations

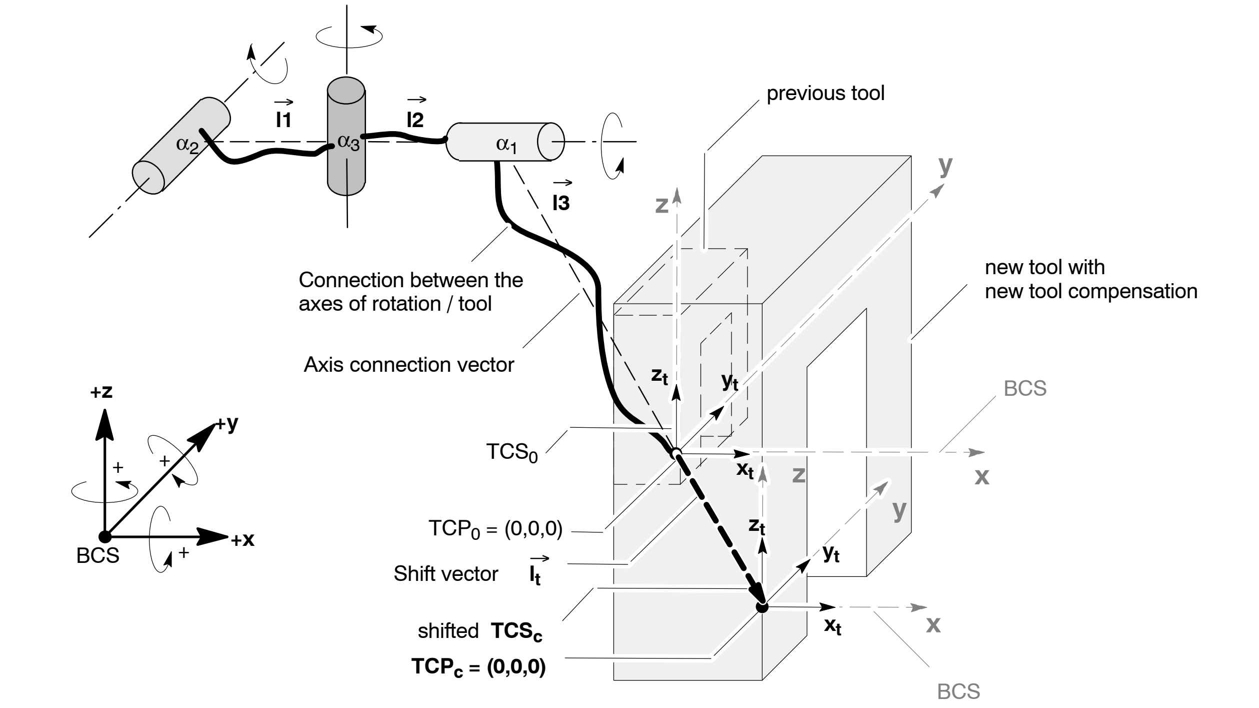
Tool compensation for the 6 axis transformation type 3033101

Using tool compensations, the $\mathbf{TCS_1}$ is freely shifted in space (here always $TCS_1=TCS_0$). A tool coordinate system $\mathbf{TCS_c}$ is thus created:

- The shift (compensation vector) of the $\mathbf{TCS_c}$ relative to the $\mathbf{TCS_0}$ or $\mathbf{TCS_1}$ is defined by $\vec{l}_t$.

☞ **A rotation of the TCS as with type 3333301 is not possible.**

Effect of tool compensation on TCS_0 —> TCS_c (Type 3033101)



In this case, the tool compensation has the same effect as replacing the configuration vector $\vec{l}_3$ by $\vec{l}_t + \vec{l}_3$.

Axis transformations

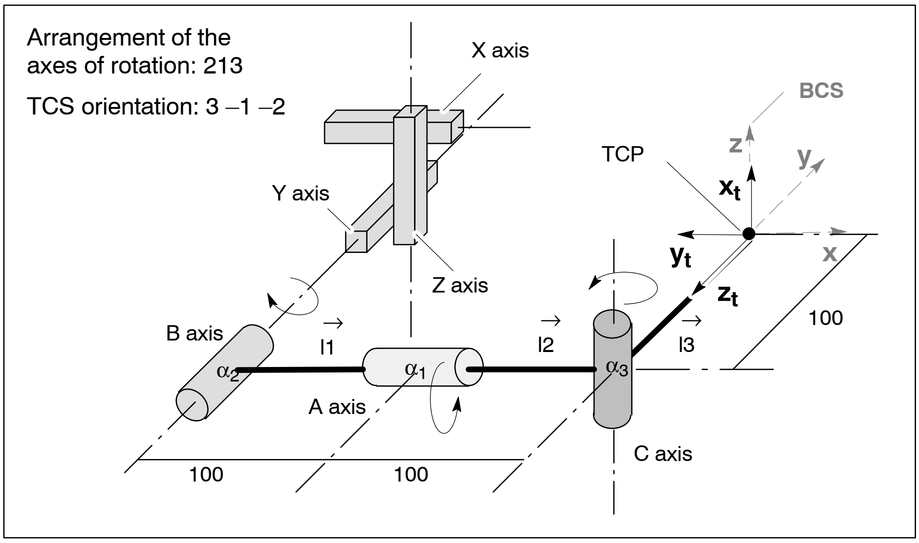
Assigning the 6 axis transformation

Transformation types 3333301 and 3033101 are used as examples.

☞ **All values to be entered refer to the reference pose.**

1. Set value 3333301 as transformation type in MP 1030 00110 (no. 1) and value 3033101 in MP 1030 00110 (no. 2).
2. The system axis numbers must be entered in MP 1030 00120 (no. 1 and 2).
   Please make sure that the 1st rotary axis is located farthest away from the tool (TCP) and that the 3rd rotary axis has the shortest distance to the TCP.
3. Components for vectors $\vec{l_1}$, $\vec{l_2}$ and $\vec{l_3}$ have to be determined appropriately. This can be done by measuring or directly from the construction drawings. Length and offset values have to be set in MP 1030 00140 [1..9] both for no. 1 and no. 2.
4. Establish the axis zero position: Approach reference pose. The axis positions in the reference pose are set in parameter 1030 00130 [1..5] both for no. 1 and no. 2.
5. The reference pose of the TCS is entered in parameter 1030 00150 for no. 1. Type no. 2 (type=3033101) requires no TCS definition.
   Care must be taken that the values result in a right-handed coordinate system. A left-handed coordinate system generates a runtime error.
6. Take over parameter using system control reset.
7. Test the 6 axis transformation in an NC program for functionality using Coord(1) and Coord(2). If no runtime error occurs, the parameters are consistent.

Example: Machine axis configuration X, Y, Z, A,B,C with the axis classifications A: 100, B: 200, C: 300 and the 6 axis kinematics 213 (arrangement of the axes of rotation):



Settings in **one** out of five MACODA parameter blocks:
1030 00110 Transformation type: 3333301 or 3033101

1030 00120 System axes of the transformation (1-64):
[1] 1 Axis X
[2] 2 Axis Y
[3] 3 Axis Z
[4] 5 Axis B
[5] 4 Axis A
[6] 6 Axis C

1030 00130 Reference pose axis positions:
[1] 0; for axis X
[2] 0; for axis Y
[3] 0; for axis Z
[4] 0; for axis B
[5] 0; for axis A
[6] 0; for axis C

1030 00140 Length and angle parameters:
[1] 100
[2] 0
[3] 0
[4] 100
[5] 0
[6] 0
[7] 0
[8] 100
[9] 0

1030 00150 Reference orientation of the tool coordinate system**:**
[1] 3 ;x_t is located along z
[2] –1;y_t is located along –x
[3] –2;z_t is located along –y

Activating the 6 axis transformation

Precondition:

- Configured 6 axis kinematics in MP 1030 00110 through MP 1030 00150.
- All NC functions and G codes used in the program must be checked for their applicability with the selected transformation using the table below.

1. Activating the transformation in the NC program using Coord(<i>).
   Example: 6 axis transformation type: 3333301

```
Coord(5)
G1 x50 y20 z40 phi45 theta30 psi190
Coord(0)
```

Deactivating the 6 axis transformation

1. Deactivate a 6 axis transformation in the NC program using Coord(0).

5.8.3 Polar coordinate transformation/end face machining

This function allows for the coordinate programming (x,y) of a contour on the end face of a workpiece under the following global preconditions:

- the workpiece is moved by a rotary or endless axis which rotates around the tool feed coordinate.
- the tool (or the workpiece itself) is moved by a linear axis which runs parallel with the workpiece clamping surface. This may be an axis along a Cartesian main direction or an axis parallel with the main direction.

The system axes on which the polar coordinate transformation is to have effect can be parametrized taking into account the above mentioned marginal conditions. Therefore, we will use the following constellation example for the descriptions below:

- **linear axis involved:**

System axis number:	1
Drive function type:	1 (drive moves rotary/linear axis)
Phys. axis designation:	X
Channel assignment:	1 (is assigned to channel 1)
Axis movement type:	1 (linear axis)

- **rotary axis involved:**

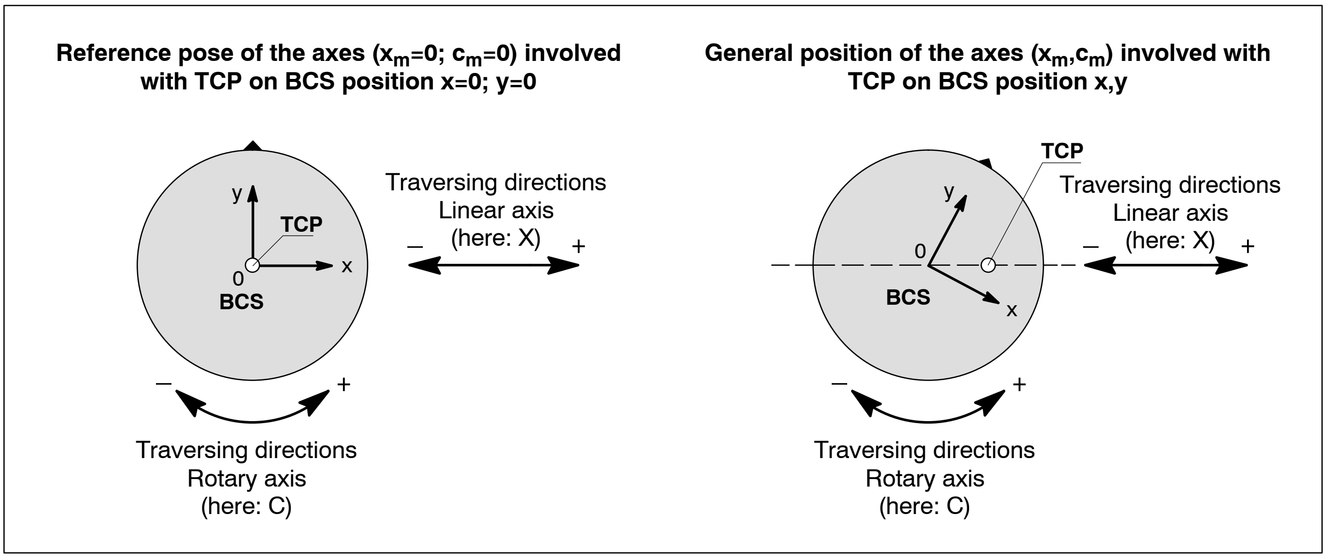
System axis number:	7
Drive function type:	3 (spindle/C axis)
Phys. axis designation:	C
Channel assignment:	0 (auxiliary axis, has to be changed to C axis mode and taken over into channel 1 prior to activating polar coordinate programming)
Axis movement type:	2 (endless axis)

The polar coordinate transformation establishes a relationship between the coordinates x and y in the basis workpiece coordinate system (BCS) and the corresponding machine coordinates of the axes involved (here: x_m, c_m). This way, every contour point can be transformed into a corresponding rotary and linear axis position pair under the following preconditions:

- The values relevant to polar coordinate transformation have to be parametrized as axis kinematics correctly in MP 1030 00110 through 1030 00140. See ”Relevant MACODA parameters” below.
- The axes involved (here: X,C) must be located on the same channel when the polar coordinate transformation is activated.
- The zero point of the BCS coordinate axes x and y is located on the axis of rotation of the rotary/endless axis involved (here: C). The rotary/endless axis rotates around the tool feed axis of the BCS (here: z).

Polar coordinate transformation/end face machining

- The linear axis (here: X) moves horizontally parallel with the workpiece clamping surface. The tool contact point moves on a straight line, which is parallel with the linear axis traversing direction. Its extension runs through the axis of rotation of the rotary/endless axis involved.
- The BCS with the coordinate axes x and y is permanently connected to the end face of the workpiece. When the workpiece rotates, the BCS rotates as well.
- In the reference pose of the system axes involved the TCP (tool center point) has to be located on the BCS position x=0, y=0.
- The axis classification of the axes involved must be specified correctly. See ”Relevant MACODA parameters” below.

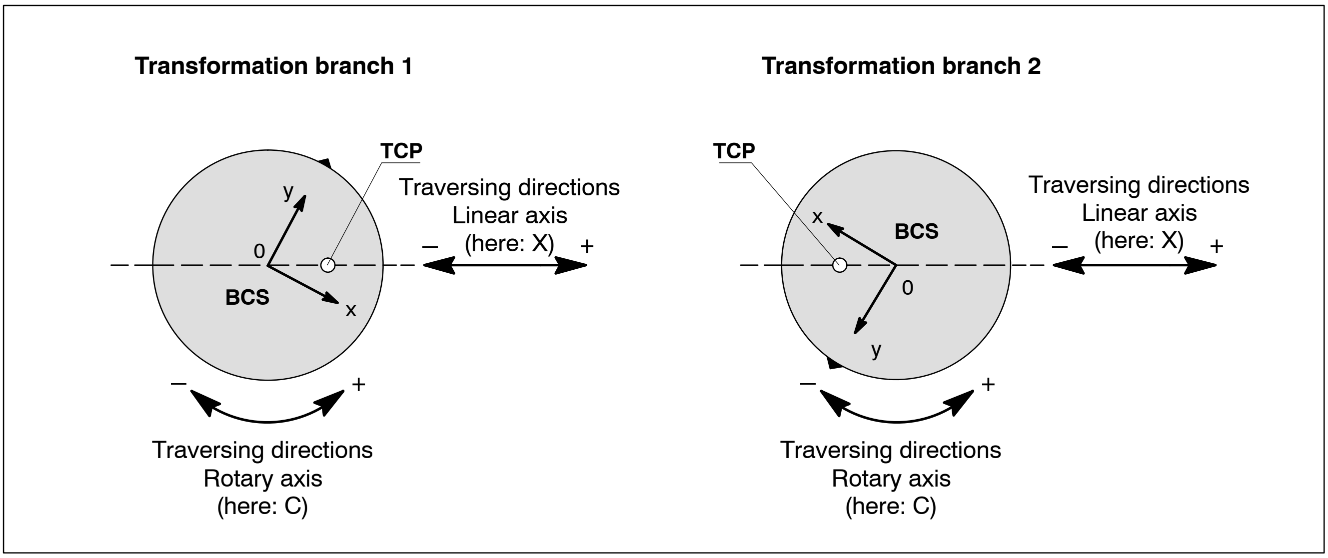


Reference pose and general position of the machine axes

Polar coordinate transformation/end face machining

Transformation branches and automatic branch change

Every TCP position, with the exception of the BCS zero point, can be described by two different machine coordinate pairs of the axes involved (see figure below). Therefore, two ”transformation branches” have been implemented in the control unit to calculate the transformation.



Identical position of the TCP in the BCS; can be described by 2 different machine coordinate pairs

Which branch is used for the internal transformation calculations, is decided when the polar coordinate transformation is activated. An automatic transition between the two branches is then possible only when the zero point of the BCS coordinate axes x and y is traversed

The **automatic branch change** is of advantage only if the BCS zero point (x=0, y=0) is to be traversed. If only one transformation branch had been implemented in the control unit, the principle would require that an intermediate block be inserted to rotate the rotary/endless axis by 180 degrees. Due to the automatic branch change, this 180 rotation is superfluous.

There are cases, however, where the automatic branch change is not desirable. The automatic branch change has to be prohibited, e.g. if the linear axis only has a limited traversing range which is not sufficient to machine the entire end face.
In this event, the NC automatically creates an intermediate block rotating the rotary/endless axis by 180 degrees. The required traversing range of the linear axis can thus be reduced to half the range.

Whether an automatic branch change is allowed or not can be configured via MP 1030 00140.

Special cases for path movements through the axis of rotation of the rotary/endless axis involved

There are two different cases: The axis of rotation of the rotary/endless axis involved can be traversed

1. without change of direction of the tool path (tangentially) or
2. with change of direction of the tool path (non-tangentially).

Ref. 1.: in case of permitted branch change, the control unit automatically switches over the branch. A 180-degree rotation of the rotary/endless axis involved is thus not necessary.

In case of prohibited branch change, the NC automatically divides the block into 3 block segments:
1st block segment: path movement up to BCS coordinate 0,0.
2nd block segment: 180-degree rotation of the rotary/endless axis involved.
3rd block segment: residual path movement from 0,0 to the end point.

Ref. 2.: if the tool path changes its direction at the BCS coordinate 0,0, this results in an unsteadiness; the rotary/endless axis would "jump" at this point.

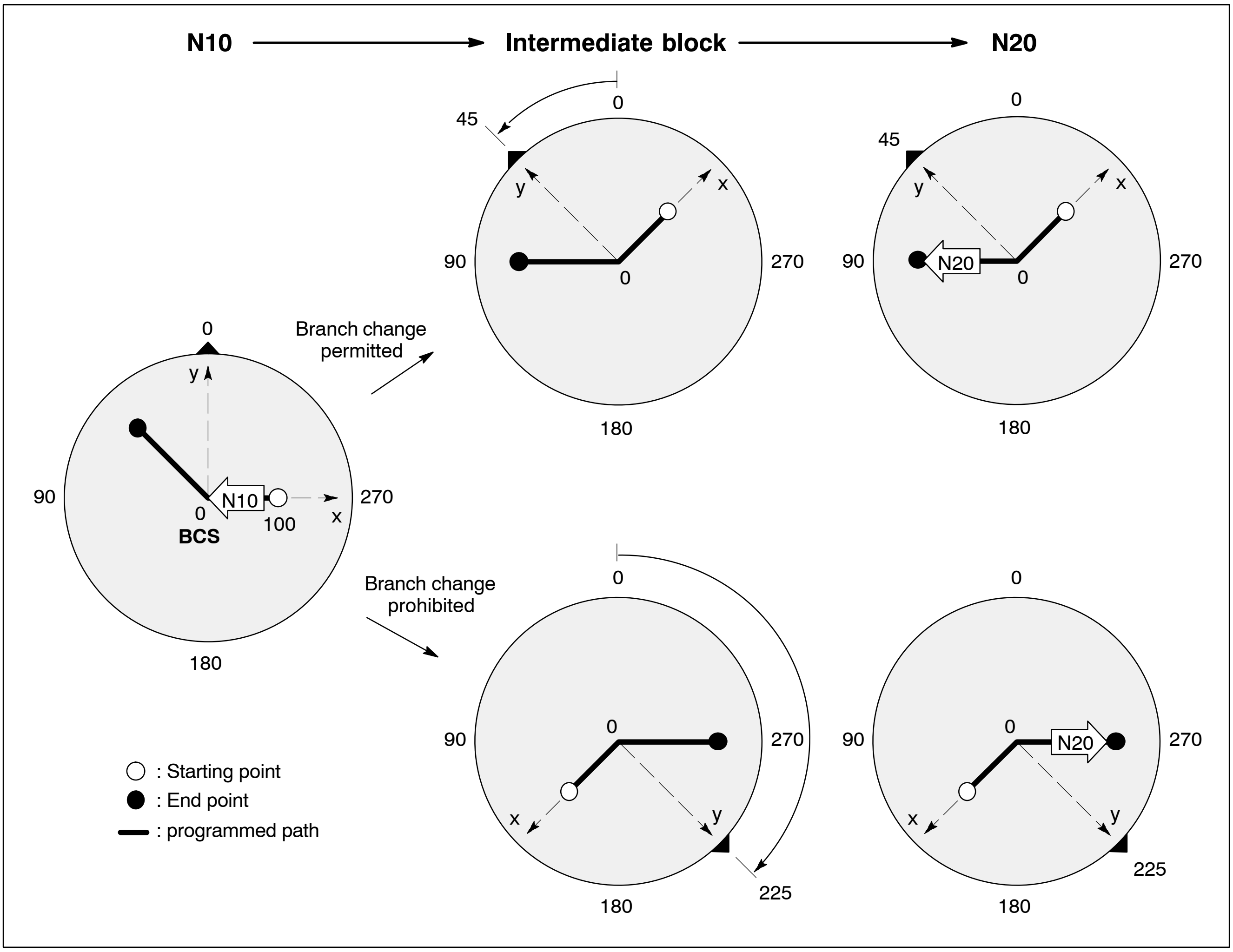
The polar coordinate transformation monitors the setpoint path to detect this effect and automatically inserts an intermediate block, if required, which rotates the rotary/endless axis correspondingly.
When creating the appropriate intermediate block, the NC also takes into account a possibly allowed branch change.

Example for 2.:

```
N05  x100 y0
N10  x0
N20  x–100 y100
```

Between the blocks N10 and N20, the NC rotates the rotary axis from position 0 degrees to 45 degrees before it continues the TCP movement via N20. A change of the transformation branch takes place in this context.
If the automatic branch change were prohibited, the NC would rotate the rotary axis to position 225 degrees. In this context, please refer to the figure below.

Polar coordinate transformation/end face machining



Creation of an intermediate block at block transition in the BCS zero point

Special cases for path movements close to the axis of rotation of the rotary/endless axis involved

- Tool paths which pass, start or end at the axis of rotation of the rotary axis (BCS coordinate: x=0; y=0) involved at a distance of ≦ 0.01 mm are led through the BCS zero point compulsorily by the NC.
  For paths leading past this range, please note:
  - the block is divided into 2 block segments. In case of a circle, two partial circles with the same radius as the original circle will be generated, if possible.
  - the first block segment ends, the second block segments starts at the zero point.

- Tool paths which pass the axis of rotation of the rotary axis involved at a distance of > 0.01 mm, principally require a reduction of the max. path speed and path acceleration. The reduction is greater, the closer the path approaches the 0.01 mm limit.
  To prevent the reduction from becoming effective along the entire path (creeping movement over the whole block), the path is divided into block segments. This results in a strong reduction only for the block segments which come close to the zero point.

Relevant MACODA parameters (MP):

1030 00110 : Axis transformation type.
2011001 has to be entered for polar coordinate transformation.

1030 00120 : System axes of transformation.
Defines the system axes involved in the polar coordinate transformation.
[1] System axis number of the linear axis
[2] System axis number of the rotary/endless axis
[3]...[8] not relevant

1030 00130: Reference pose axis positions.
Defines the system axis position at reference pose.
[1] Position of the linear axis
[2] Position of the rotary/endless axis
[3]...[8] not relevant

1030 00140 : Length and angle parameters.
Determines if the linear axis can traverse the BCS zero point.
Please note: x_m: current system axis position
x_{mref}: system axis position in the BCS zero point.
[1] 0: yes
+1: no. Pos. positions allowed only ($x_m – x_{mref} \geqq 0$)
–1: no. Neg. positions allowed only ($x_m – x_{mref} \leqq 0$)
[2]...[16] not relevant

☞ **As MACODA allows for the parametrization of max. 5 kinematics, it is also possible to configure several polar coordinate transformations.**

7010 00030: Classification for logical axes.
Defines the functional relevance of the **logical axes** involved in the polar coordinate transformation. Therefore, please refer to the MACODA manual.
Since the polar coordinate transformation takes into account both positive and negative directions of movement of the axes, it may be necessary to parametrize the respective axis classification value with a sign.
Please note:

- Moves in the reference pose of the rotary/endless axis of the TCP along +x in case of positive traversing direction of the linear axis involved, then the axis classification value of the linear axis is positive, otherwise it is negative.
- If a TCP position in the first BCS quadrant results in a negative axis position of the rotary/endless axis involved, the axis classification value of the rotary/endless axis involved is positive, otherwise it is negative.

7080 00010: Designation of the coordinates:
Determines the names used for the coordinates of a channel in case of active axis transformation in the part program.

Assigning

1. Determine the necessary axis classifications of the axes involved and enter the data in MP 7010 00030.
2. Set value 2011001 as transformation type under a free transformation parameter block (e.g. no. 1) in MP 1030 00110.
3. Enter the system axis numbers of the axes involved in MP 1030 00120.
4. Approach the reference poses of both axes (position x=0, y=0). Store the resulting machine axis positions in MP 1030 00130.
5. Determine in MP 10300 0140 whether the linear axis can traverse the axis of rotation of the rotary axis involved.

CAUTION
Danger of damage to the machine or the workpiece!
Do not initiate a system control reset during machining.

---

6. Initiate a system control reset (DIAGNOSTICS ▶ RESET FUNCTIONS ▶ SYSTEM CONTROL RESET).
7. Program command COORD(1) in an NC program.
   If no runtime error occurs, the parameters of the polar coordinate transformation are consistent.

Activating

Precondition:

- The axis kinematics has been assigned correctly.

1. Program COORD(1).
   The polar coordinate transformation has been configured in transformation parameter block 1 (see above under “Assigning”)

Deactivating

1. Program COORD(0).
   A presently active axis transformation/polar coordinate transformation is switched off.

Polar coordinate transformation/end face machining

Notes:

6 Precision / Dynamics

6.1 Maximum axis step change

Function:

Determines the maximum difference permitted for machining axes (synchronous axes) between two consecutive speed setpoint values in the interpolator clock pulse at the block transition.

Thus it is possible to

- prevent the control unit from providing speed setpoint values at the block transitions during the execution of part programs in operating mode "Automatic" which would dynamically overload the connected drive (see also "Shape" function, page 6–7 ff.)
- reduce the machining time for a tool, if G8 (path slope, see page 6–11) is active.

Restrictions:

- The acceleration at the block transitions generated by the maximum axis step change (⇒ MP 1010 00011) is also dependent on the interpolator clock pulse that has been set (⇒ MP 9030 00001, NC cycle time) (a=v/t).

  If, e.g. the interpolator clock pulse is increased (by reducing the NC cycle time) without adapting the maximum axis step change correspondingly, the resulting higher acceleration may, under certain circumstances, no longer be carried out correctly. On the other hand, a reduction of the interpolator clock pulse (by increasing the NC cycle time) may have the effect that the achievable drive dynamics is not fully utilized.

  If you change the interpolator clock pulse, it is therefore necessary
  - to adapt the maximum axis step change in MP 1010 00011, or
  - limit the acceleration that can be generated at the block transition via MP 1010 00012. In this case, the control unit automatically **reduces** the maximum axis step change, if required.

Relevant MACODA parameters (MP):

1010 00011:	Maximum axis step change
1010 00012:	Maximum axis step change acceleration
7030 00310:	Minimum angle for calculation of the maximum axis step change

Relevant G functions:

G8: Path slope on.
During contour machining, the control unit attempts to generate a speed as constant as possible within the magnitude of the programmed feedrate at the block transitions, too. For details, please refer to page 6–11). Please note that G8 and the

”Shape” function (see page 6–7 ff.) cannot be active simultaneously.

G108: ”Contour-dependent feedrate reduction” on.
The control unit monitors the deceleration path available.

G228: Block transition without deceleration.
Prevents the path speed at the block transition from being reduced on account of the maximum axis step change below a defined contour knee angle.
For this purpose, it is possible to influence the basic setting of the minimum angle in MP 7030 00310 program-controlled via K address (G228 K<*angle*>).

G328: Precision programming (see page 6–4).
Reduction of the path speed at the block transition.

6.1.1 Assigning

1. Enter the maximum permitted axis step change accelerations for your application into MP 1010 00012. In the calculation, take into account both the mechanical data limits as well as the expected average load.
2. Calculate the resulting maximum axis step change v=a*t (where t corresponds to the interpolator clock pulse that has been set) for every machining axis (⇒ MP 9030 00001).
   Enter the calculated values into MP 1010 00011.
3. Adapt MP 7030 00310 (minimum angle for calculation of the maximum axis step change).

CAUTION
Danger of damage to the machine or the workpiece!
Do not initiate a system control reset during machining.

---

4. Initiate a system control reset (DIAGNOSTICS ▶ RESET FUNCTIONS ▶ SYSTEM CONTROL RESET).

6.1.2 Activating

1. Program G8.
   With active G8, the maximum axis step change is automatically taken into account during the execution of part programs in the operating mode ”Automatic”.

6.1.3 Deactivating

1. Program G9.

6.2 Path acceleration limitation

Function:

Makes sure that a parametrizable positive/negative acceleration is not exceeded on any of the programmed paths.
Although this may result in a slightly longer processing time of a part program, it prevents a creeping dynamic overloading of the drives. This is an advantage in particular when the machining process is frequently carried out near the limits of load and dynamics.

Relevant MACODA parameters (MP):

7030 00210: Path acceleration.
Max. admissible positive path acceleration on the programmed contour.

7030 00220: Path deceleration.
Max. admissible negative path acceleration on the programmed contour.

6.2.1 Assigning

1. Enter the desired acceleration values into MP 7030 00210 and MP 7030 00220.

☞ **Values > 0 have to be entered for both parameters.**

CAUTION
Danger of damage to the machine or the workpiece!
Do not initiate a system control reset during machining.

---

2. Initiate a system control reset (DIAGNOSTICS ▶ RESET FUNCTIONS ▶ SYSTEM CONTROL RESET).

6.3 Precision programming

Function:

In certain applications, a **controlled** ”slurring” of contour transitions may be expressly desired (e.g. when machining freeform surfaces).

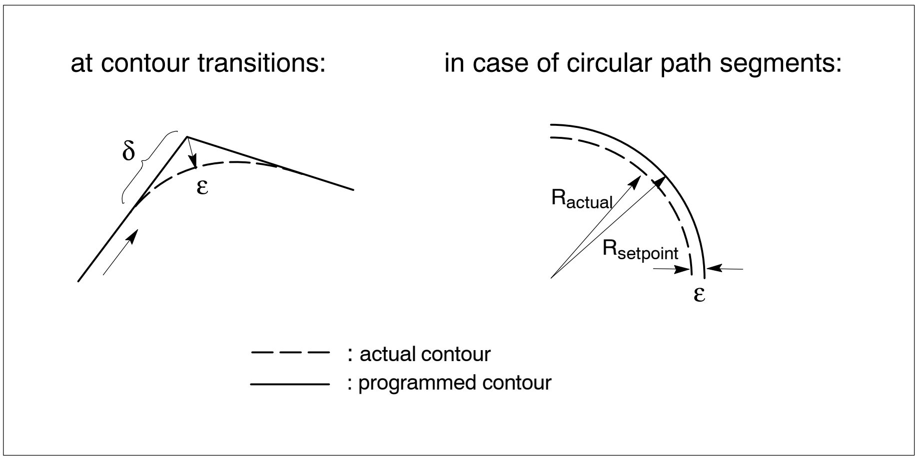
Therefore, ”precision programming”

1. limits
   - the contour error (ε) or
   - the path overtravel (δ) at contour transitions

   and

2. in case of circular path segments (circles, helical, helicalN)
   - the radius error (ε)

   to a programmed value (in mm or inch; depending on active G71 or G70, respectively).



For this purpose, the control unit calculates the maximum path speed for each contour transition or circular path segment that may not be exceeded, so that the specified limit value is observed.

Restrictions:

- The ”Precision programming” function is only effective for contour transitions if G8 or G108 are active. It is always effective for circular path segments.
- The ”Exact positioning” function has to be switched off (G62 is active).
- To calculate the max. path speed, only the controller parameters of the first axis on the channel are used. Therefore, all axes involved in the path have to be parametrized identically with regard to the dynamics.
- To calculate the max. path speed, the control unit assumes there is a linear relationship between speed and overtravel. The influence of acceleration operations on the current overtravel are not taken into account.

- If ”Feed forward” (traversing without following distance error; feed forward control) is active in the drive, the control unit will only take into account SERCOS parameter P-X-0500 to calculate the admissible path speed in connection with Servodyn drives. For this purpose, the control unit reads the parameter value from the drive during SERCOS phase start-up and stores it.
  If values > 99% are entered for P-X-0500, a following distance error of 1% of the following distance error without feed forward control is assumed.
  In the case of non-Servodyn drives, the control unit calculates the admissible path speed assuming that the overtravel is reduced by half in case of active feed forward control (in the drive).

Relevant MACODA parameters (MP):

8003 00001: Contour deviation tolerance with G328, precision programming

Relevant G functions:

G328 **or**

G328 EPS<*Contour error*> **or**

G328 DIST<*Path overtravel*>

Precision programming ON in every case.
If **only G328** is programmed, the control unit limits the contour/radius deviation ε at the contour transitions or in case of circular path segments. The value set in MP 8003 00001 serves as limit value.
If **G328 is programmed in connection with EPS**, the control unit limits the contour/radius deviation ε at the contour transitions or in case of circular path segments, respectively, to the value programmed after EPS.
If **G328 is programmed with DIST**, the control unit limits the path overtravel δ at contour/radius transitions to the value programmed after DIST. When traversing circular path segments, the control unit now limits the radius deviationε in accordance with MP 8003 00001.

DIST and EPS must not be programmed together in one block.

G329: Precision programming OFF.

Relevant SD functions:

SD(328,1) provides the maximum permitted contour/radius error ε programmed last on the active channel as rounded integer value.
Alternative programming: SD(328).

SD(328,2) provides the maximum permitted path overtravel δ programmed last on the active channel as rounded integer value.

SDR(328,1)	the same as SD(328,1), however return value in real format. Alternative programming: SDR(328).
SDR(328.2)	the same as SD(328.2), however return value in real format.

If the ”Precision programming” function has been switched off (G329 is active), all functions provide value 0.

6.3.1 Assigning

1. Enter the maximum permitted contour/radius deviation ε into MP 8003 00001.

CAUTION
Danger of damage to the machine or the workpiece!
Do not initiate a system control reset during machining.

2. Initiate a system control reset (DIAGNOSTICS ▶ RESET FUNCTIONS ▶ SYSTEM CONTROL RESET).

6.3.2 Activating

Preconditions:

- Presetting for contour/radius deviation ε in MP 8003 00001 has been parametrized correctly.
- G8 or G108 are active
  (only required if ”Precision programming” is to be used at contour transitions. The function is always effective for circular path segments.)
- All axes involved in the path have been parametrized identically with regard to the dynamics.
- The ”Exact positioning” function is switched off (G62 is active).

1. Program G328 according to the syntax described.

6.3.3 Deactivating

1. Program G329.

6.4 Shape

Function:

Using the ”Shape” function, it is possible to lessen or limit the jerk in case of changes in speed. This reduces the peak load of mechanics and drive components and may, when operating near the limit, help avoid servo errors.

Thus ”Shape” can, for example, be used for positioning operations involving high accelerations (e.g. punching application) and/or high loads to minimize vibration on the machine.

Two types of functions are available in connection with ”Shape”:

- Jerk-limited velocity profile generation with path shape (G108): splits up jumps occurring during path acceleration generated by contour-dependent feedrate reduction into several interpolation cycles and thus lessens the jerk (see also Section 6.7).
- Point-to-point movement using Shape (G408): splits up jumps occurring during path acceleration into several interpolation cycles specified via programming and thus lessens the jerk.
- Axis-by axis programmable Shape (G608): limits the maximum jerk of machining axes. If several axes are involved in the interpolation, the control unit makes sure that the respective jerk limit value r_{max} (see Section 6.4.1) is not exceeded on any of the axes involved.
  As opposed to the ”Point-to-point movement using Shape” where the *desired* number of interpolation cycles is specified, the permitted maximum number of interpolation cycles is specified here. The control unit automatically calculates the required data on the basis of the travel paths and the permitted maximum interpolation cycles.
  If G608 is programmed without axis addresses, the permitted maximum number of interpolation cycles from MP 1003 00008 is used.

☞ **Servodyn drives also provide a ”Shape” function (see drive parameters P-0-0033, P-0-0513, P-0-0514, P-0-0526).**
It can be activated additionally at the drive end, if required.

Restrictions:

- You should not use ”Shape” for contour machining.
- ”Point-to-point movement using Shape” (G408), ”Axis-by-axis programmable Shape” (G608) or ”Path slope” (G8, G108) cannot be used simultaneously.
- In comparison with deactivated ”Shape”, the interpolation time per point-to-point movement in case of activated ”Shape” increases by the additionally necessary interpolation cycles because the movement becomes more ”smooth”.

Shape

Relevant MACODA parameters (MP):

1003 00008: Order of Shape for axis.
Defines the default settings for G608 axis by axis.

1010 00001: Maximum axis acceleration.

Relevant G functions:

G9: Shape/path slope off.

G408: Point-to-point movement using Shape.
The SIN and LIN parameters determine how an acceleration jump is split up.

- LIN x: linear splitting up of the jump into x interpolation cycles (x: 2 to 41).
- SIN x: SIN^2-shaped splitting up. Strictly specified values are admissible for x only (see DIN Programming Instructions).

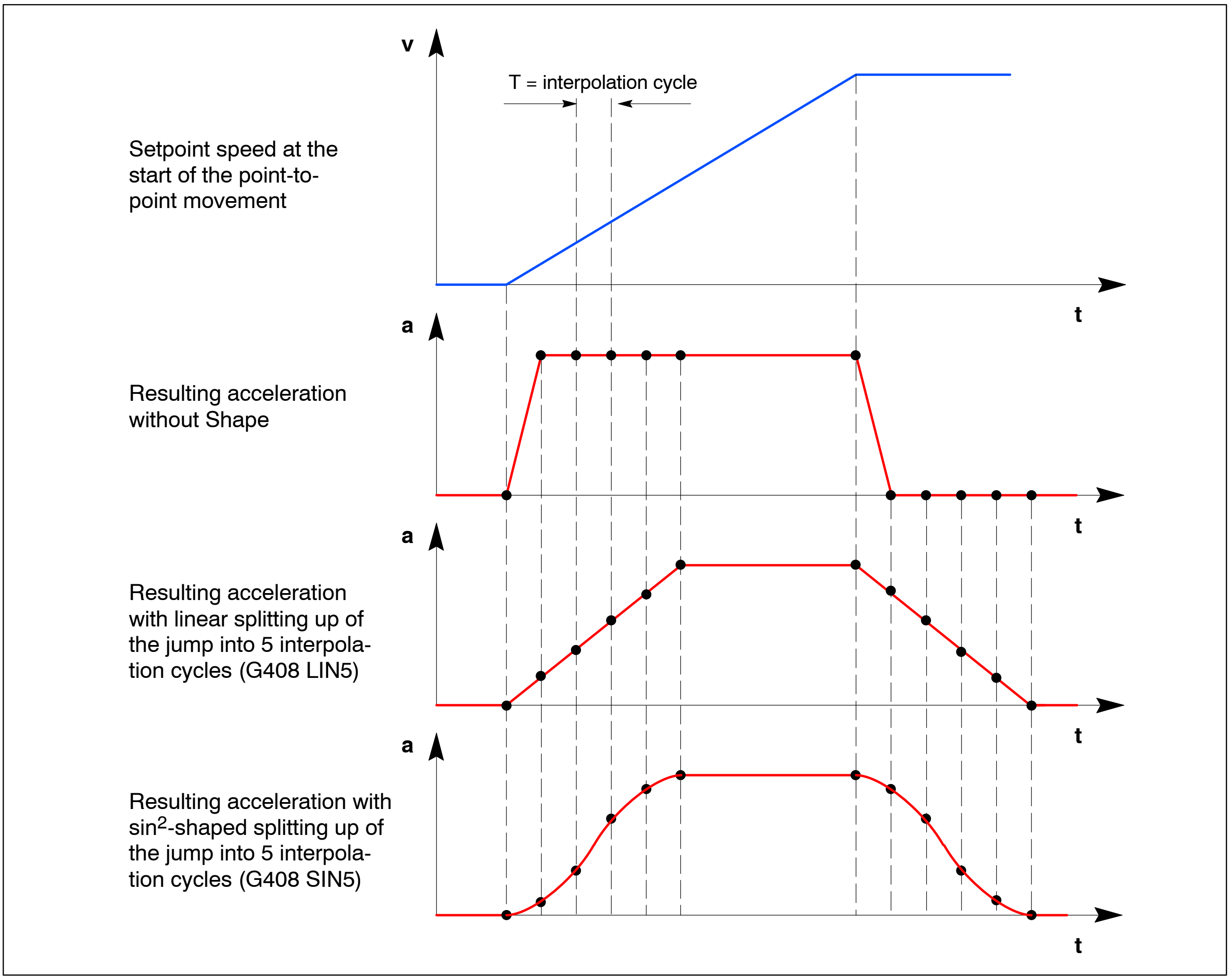
"LIN 2" acts without programmed parameters.

G608: Axis-by axis programmable Shape.
The default settings are programmed via MP 1003 00008.

G6: Acceleration programming.
Supersedes MP 101000001 temporarily.

☞ **The G8, G9, G108, G408 and G608 G functions form a group and therefore deselect each other mutually.**

Shape



Effects of the SIN and LIN parameters in case of G408

6.4.1 Assigning

☞ **Parametrizing is only necessary when using ”Axis-by-axis programmable Shape” (G608)!**

1. Adapt the Shape orders of the individual axes in MP 1003 00008. The values entered correspond to the maximum number of interpolation cycles which the PNC may use to split up an acceleration jump on the respective axis when the axis is traversing alone.

   The values in MP 1003 00008 are used for all axes whose axis addresses have **not** been programmed in a G608 block.

Shape

If the maximum permitted ”jerk” is known, use the following equation to convert from ”Jerk” to ”Shape order”.

$$S = 1000 * a_{max} / (r_{max} * T_{ipo})$$

Please note:

S Shape order. Please note for the calculated value: $1 \leq S \leq 21$. The calculated value has to be entered as integer (round up/off!) into the respective individual parameter of MP 1003 00008.

a_{max} Parametrized maximum axis acceleration (value from MP 1010 00001).

r_{max} Maximum permitted ”jerk” of the axis in the unit of measure m/s^3.

T_{ipo} Parametrized ”NC cycle time” (value from MP 9030 00001).

Example:

max. permitted jerk:	500 m/s^3
maximum axis acceleration:	10 m/s^2
NC cycle time:	4 ms
Required input value:	5

CAUTION
Danger of damage to the machine or the workpiece!
Do not initiate a system control reset during machining.

2. Initiate a system control reset (DIAGNOSTICS ▶ RESET FUNCTIONS ▶ SYSTEM CONTROL RESET).

6.4.2 Activating

1. Program G408 using the LIN or SIN parameter to activate the function ”Point-to-point movement using Shape”
   -or-
   program G608 to activate the ”Axis-by-axis programmable Shape” function.

☞ **G608 can be programmed alone or with the ”Axis address” parameters and the ”Shape order” value (e.g. G608 X4).**
See Section 6.4.1 for an equation for converting ”Jerk” into ”Shape order”.

6.4.3 Deactivating

1. Program G9.

6.5 Path slope

Function:

During machining at feedrate, the function tries to keep the feedrate speed as constant as possible, even at the block transitions. For this purpose, the function always looks at the programmed contour and the available deceleration path 1 block in advance.
If it detects an unsteady contour transition (corner), the feedrate speed is reduced taking into consideration the maximum axis step change (see page 6–1) only to the extent that is necessary for going around the corner due to the available drive dynamics.

This way, you achieve

- higher quality contour transitions and in particular
- shorter machining times

because it is no longer necessary to decelerate down to path speed v=0 before every contour transition.

☞ **Decelerating to v=0 is performed at the end of each programmed path.**

Restrictions:

- ”Path slope” acts only on the machining axes (synchronous axes).
- Decelerating to v=0 is performed after each G0 block.
- After a G200 block, decelerating to v=0 is only performed if G61 or G163 is active.
- MP 7060 00110 (number of blocks in preparation) must at least be assigned the value ”3”.
- If the travel path of the last two programmed blocks is too short at high path speeds, the drive dynamics may be insufficient to be able to decelerate correctly to v=0. See also ”Contour-dependent feedrate reduction” on page 6–14 ff.
- In case of active ”path slope”, the execution time of the block has to be longer than the execution time of auxiliary functions (incl. acknowledgment) programmed in one and the same block.
  The execution time is for the most part determined by the programmed travel path as well the setpoint feedrate.

Relevant G functions:

G8: ”Path slope” on.

G9: ”Path slope” off.
Decelerating to v=0 is always performed at block transitions.

☞ **The G8, G9, G108, G408 and G608 G functions form a group and therefore deselect each other mutually.**

6.5.1 Activating

1. Program G8.

6.5.2 Deactivating

1. Program G9.

6.6 Look-ahead function

Function:

Using the ”Look-ahead” function, the PNC is capable of examining the programmed path across several machining blocks ahead of the machining operation.
Only then is it possible to detect potentially difficult path properties and to react accordingly.

The look-ahead function is the basis for all functions

- performing a contour-dependent change of the path speed and
- influencing a programmed contour, e.g. by automatically inserted additional contour segments.

For the look-ahead function to work properly, it is necessary to hold a sufficient number of blocks in block preparation. This allows the control unit to utilize the block information it needs to calculate the different strategies in advance.

For machining channels, at least 10 blocks should be reserved for this purpose. If many small contour segments have to be processed subsequently, it may be necessary to increase the number of blocks.
For CPL channels, 3 blocks are normally sufficient for the look-ahead function because the CPL channel performs no contour machining.

Determine the number of blocks in the block preparation (SAV) per assigned channel using configuration parameter 7060 00110. With the PREPNUM function, the number of blocks can be temporarily restricted via the part program.
The standard presetting for channel 1 and 2 is 30 blocks each, for the CPL channel it is 3 blocks. For additional machining channels, at least 30 blocks should be reserved, too.
If the presetting of MP 7060 00110 may be critical or limiting for functions using the look-ahead feature, a note to this effect will be included in the respective function description.

☞ **Please note that the look-ahead function requires approx. 8 k class 2 system memory per block!**

★ Before you increase the number of blocks, verify that sufficient memory is available.
DIAGNOSIS ▶ CONTROL DIAGNOSIS ▶ MEMORY

☞ **To take over changes in MP 7060 00110, perform a control reset.**

Relevant MACODA parameters (MP):

7060 00110: Number of blocks in preparation.

Relevant special functions:

PREPNUM: Limits the maximum number of blocks processed in block preparation. For details, see ”DIN Programming Instructions“.

6.7 Jerk-limited velocity profile generation

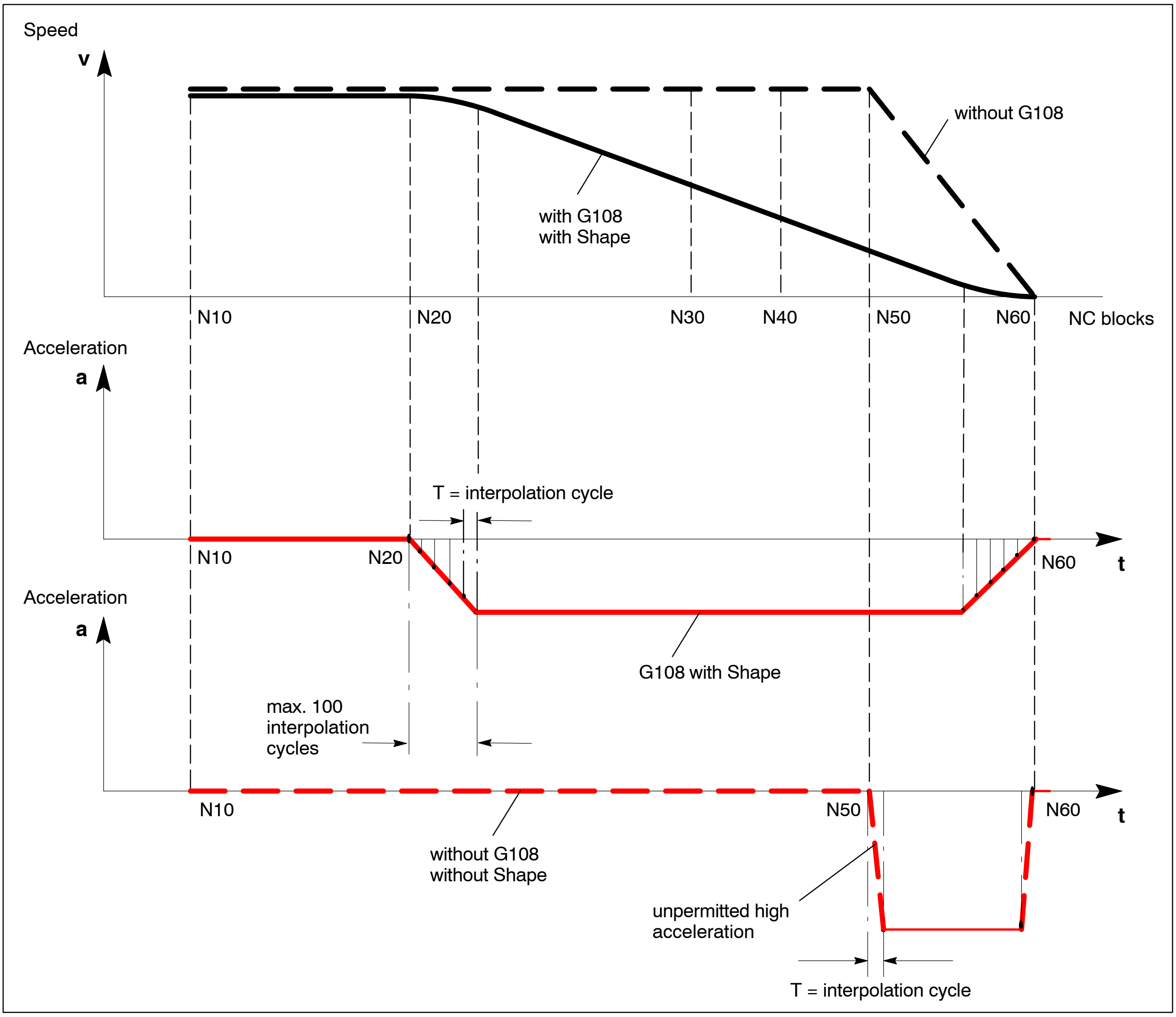
Function:

The **jerk-limited velocity profile generation** is a “contour-dependent feedrate reduction” with optional Shape functionality.

It determines

- by means of **look-ahead function** (advance interpolation), if the distance from the end point of the current block to the end point of the next block is sufficient to decelerate from the current path speed to zero. If the distance is not sufficient, the following blocks are included in the feedrate determination, limited by the maximum number of blocks entered in the MACODA parameters. This serves to avoid unpermitted high accelerations during block changes.
- a velocity profile by the **Shape** functionality (subsequent interpolation) in the path operation: The turning points in the **velocity profile** determined for the contour-dependent feedrate reduction generate jumps in the path acceleration. These jumps are split up into several interpolation cycles and thus smoothed. The number of cycles is programmable and limited to maximum 100 interpolation cycles.

Jerk-limited velocity profile generation



Restrictions:

- The ”contour-dependent feedrate reduction” acts only on the machining axes (synchronous axes).
- A PREPNUM active in the part program limits the jerk-limited velocity profile generation with the effect that fewer blocks than have been defined in the relevant MACODA parameters are available.
- If the look-ahead range is set too low, this may result in an unnecessary reduction of the path speed.
- Within the deceleration ramp, path speeds which differ block by block will be effective only if the difference exceeds a fixed tolerance range (±15%).

Relevant G functions:

G108 {Shape<Shape order>}
Switch ”Contour-dependent feedrate reduction” on (optional with Shape)

If G108 is programmed without the optional Shape, the order set in MP 7050 00320 becomes effective.

Relevant NC function:

Shape<Shape order>

The following applies:

<Shape order> Number of interpolation cycles (optional), programmable number 0..100.
Without programming, MP 7050 00320 shall apply

☞ **The G8, G9, G108, G408 and G608 G functions form a group and therefore deselect each other mutually.**

Relevant MACODA parameters (MP):

7050 00320: Default order for Shape, if G108 is programmed without Shape parameter.
0: Presetting
0...100: Number of interpolation cycles

7050 00399: Syntax of path slope options
(for system administrators only)
The syntax for the 2 existing path slope options can be changed under this parameter.
G8, G08, G008: Path slope (see Section 6.5)
G108 : Look-ahead (contour-dependent feedrate reduction)

7060 00110: Required total number of blocks for look-ahead range, **including** interpolation on the channel.
0: Presetting
1 ...1000: Number of blocks

The interpolator requires max. the number of blocks set or programmed as Shape order (number of interpolation cycles) (1 cycle = 1 block).
If the interpolated NC blocks are longer, the blocks required by the interpolator is reduced accordingly. All NC blocks not required by the interpolation, are available to the look-ahead functionality.

7060 00120: Maximum number of blocks for the look-ahead range
This maximum value is only relevant if MP 7060 00110 is configured sufficiently large **and** a corresponding portion in parameter 7060 00130 is assigned to the jerk-limited velocity profile generation.
1: Presetting
0...1000: Number of blocks

Jerk-limited velocity profile generation

7060 00130: Percentage of blocks for the look-ahead range
Applies to the groups velocity profile, buffered blocks, spline and others.
100: Presetting
0 .. 100: Percentage share

The individual parameter [1] of this parameter specifies the percentage share of the "total number of blocks for look-ahead range" (7060 00110) available to the jerk-limited velocity profile generation.

Relevant special functions:

PREPNUM: Limits the maximum number of blocks processed in block preparation. Has priority over the values set in the relevant MACODA parameters. For details on PREPNUM, see "DIN Programming Instructions".

6.7.1 Assigning

Precondition:

Please note that the function requires further blocks (maximum up to Shape order) from the interpolator in addition to the set number of blocks for the look-ahead range. These blocks have to be taken into account in MACODA parameter 7060 00110 "Number of blocks in preparation", too!

1. Determine the number of blocks for look-ahead and Shape in MP 7050 00320, 7060 00110, 7060 00120 and 7060 00130.
   When specifying the number of blocks, take into account other functions as well (e.g. spline online calculation), if required.
   If the number of interpolation cycles is not sufficient, the ratio in 7060 00130 or the total number in 7060 00110 have to be changed.
2. If any changes have been made in the relevant MACODA parameters, System control reset has to be initiated.
3. Before you increase the number of blocks, verify that sufficient memory is available.
   DIAGNOSIS ▶ CONTROL DIAGNOSIS ▶ MEMORY

6.7.2 Activating

CAUTION
Danger of damage to the machine or the workpiece!
Do not initiate either a control reset or a system control reset during machining, because the NC blocks prepared relative to the look-ahead function will not be executed anymore.

1. Program G108 or G108 SHAPE.

6.7.3 Deactivating

1. Program G9.

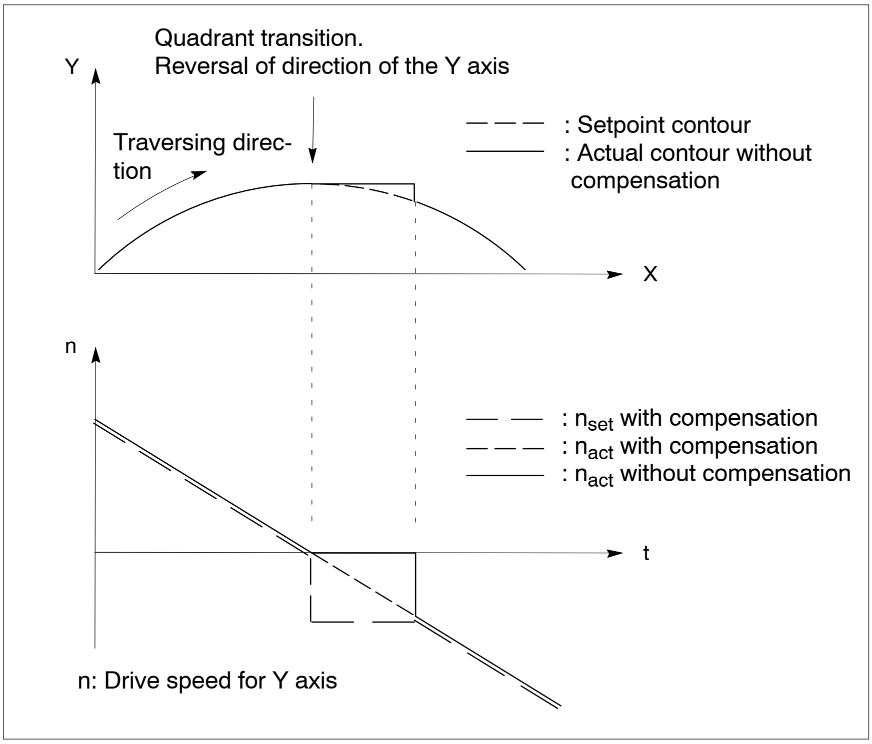
6.8 Circle compensation at the quadrant transition

Function:

Reduces the impulse-type contour distortions caused by static friction on circular paths following the quadrant transitions.

Circle errors at the quadrant transitions are generated, among others, by the reversal of direction of an axis involved in a circular path. One axis moves at maximum speed at the transition point, while the other axis comes to a stop. When accelerating in the new traversing direction, the static friction holds back the axis at the turning point at first. This results in an overtravel which is only reduced in a jerk when the static friction limit is exceeded.

For this reason, the drive is able to add a triangle-shaped speed pulse independently in case of a reversal of direction of an axis, which is reduced starting at a certain pulse level in a certain number of scanning steps (drive function group ”Circle compensation”, see ”Servodyn Parameter Manual”)



To commission this drive function, it is necessary to:

- traverse a circular path with a radius that is typical of the machine (determined by you) at different speeds and establish the pulse levels required for optimum compensation.
  The value pairs of **speed** and required **pulse level** thus established can be stored in the Servodyn drive in SERCOS parameters P-0-0539 (speed table speed pulse) and P-0-0540 (pulse table speed pulse) (max. 20 value pairs).
- store the **circle radius** used in P-0-0537 (reference radius path speed).

Circle compensation at the quadrant transition

- determine the **pulse area** empirically and save it in P-0-0538 (pulse area speed pulse).
  The pulse area (= pulse level * scanning steps) remains nearly constant for differing speeds and therefore needs to be determined for one single speed only.

The pulse level required for compensation and the number of scanning steps needed are dependent on radius and feedrate speed (path speed). Because the drive knows neither the radius nor the path speed of a circle segment that is currently being traversed, this information is taken into account by the NC, it is standardized and the result is transmitted to the drive in the form of the ”standardized circle feedrate” (P-0-0536 standardized path speed). For this purpose, SERCOS parameter P-0-0536 has to be included in the configuration list of the MDT telegram (S-0-0024).

Calculation of the ”standardized circle feedrate” in the NC:

$$V_{stand} = V_{path} \times \sqrt{(P\text{-}0\text{-}0537 / R)}$$

V_{stand}: standardized circle feedrate.
The value is calculated separately for the axes involved in the circle path (main and second axis), it is only active in the course of the circle path and is set to zero at the end of the circle.

V_{path}: current path speed.

R: programmed radius.

P-0-0537: reference radius.
It is read from the drives involved by the NC (in SERCOS phase 4).

On the basis of the standardized circle feedrate, the drive is now able to determine the pulse level required for compensation from the value pairs (speed, pulse level; P-0-0539, P-0-0540) established during commissioning. If no value pair is available for the current path speed, the required pulse level is calculated by the drive via linear interpolation.

From the pulse level, the drive finally determines the number of scanning steps which are necessary for the linear reduction of the compensation speed pulse (scanning steps = pulse area / pulse level).

Restrictions:

- Active axis transformations are not permitted.
- The axes of the program coordinate system (PCS) must run parallel with the axes of the basis coordinate system (BCS), i.e. shifts in positive or negative direction are allowed whereas rotations are not!

Circle compensation at the quadrant transition

Relevant SERCOS parameters:

P-0-0510: Position options.
Bit 4 switches the drive's own circle compensation (function "Friction compensation with speed pulse") (bit 4 = 1) on.

P-0-0536: Standardized path speed.
It is transmitted cyclically to the drive by the NC.

P-0-0537: Reference radius path speed.
The NC needs it to calculate the standardized circle feedrate.

P-0-0538: Pulse area speed pulse.
The NC needs it to calculate the required scanning steps.

P-0-0539: Speed table speed pulse.
Speeds used to determine the pulse levels entered in P-0-0540.

P-0-0540: Pulse table speed pulse.

S-0-0024: Configuration list MDT.
Determines which SERCOS parameters the drive expects in the freely configurable telegram.

S-0-0015 Telegram types parameter.

6.8.1 Assigning

Precondition:

- Use of Servodyn drives with software version V47 (or higher).
- The relevant SERCOS parameters (see above) are correctly set in the drive and saved in the EEPROM or will be transferred by the control unit to the drive during SERCOS initialization (for information on SERCOS, see page 4–67 ff.).

1. Integrate SERCOS parameter P-0-0536 into S-0-0024 and make sure that the "freely configurable telegram" has been selected in S-0-0015.
2. Select a radius typical of the machine (e.g. 100 mm) and enter the value in P-0-0537 (unit: mm, without decimal places).
3. Enter all path speeds in P-0-0539 (unit: mm/min; without decimal places) you wish to determine the pulse level for (e.g. in the range 100 mm/min to 5000 mm/min). Specify the path speeds (max. 20 values) in ascending order separated by commas. In P-0-0540, the pertaining pulse levels will be entered later (also separated by commas) in the unit rpm (with 3 decimal places).
4. Create a part program, where a full circle with the previously specified radius (see item 2.) and the first path feedrate included in P-0-0539 (see item 3.) has been programmed. Start and end point should **not** be located on a quadrant transition.
5. Make sure that P-0-0536 includes the value 0.
6. Set bit 4 in P-0-0510 (position options) to "1". This will activate the circle compensation in the drive.

Circle compensation at the quadrant transition

7. Save all SERCOS parameters in the EEPROM of the drive or enter the SERCOS parameters mentioned until now and their values into the relevant SERCOS files of the NC. The SERCOS files are transferred by the control unit to the drive during SERCOS initialization (for information on SERCOS, see page 4–67 ff.).

CAUTION
Danger of damage to the machine or the workpiece!
Do not initiate a control RESET during machining.

8. Start up the control unit and the drives.
9. Use an axis oscilloscope to record the actual positions of the axes involved for the part program that was previously created. When you specify the scanning time, please note that the smaller you set the scanning time, the shorter the maximum possible recording duration will be.
   After the recording process, actuate the "Calculate curve" softkey and enter the data of the setpoint circle ("Enter circle data" softkey). If you use the "Circle deviation" softkey, the axis oscilloscope will show you the circle deviation and the quadrant transitions on the display.
   In the current state, you should be able to detect a circle error at the quadrant transition.
10. Change the pulse area (P-0-0538; max. 3 decimal places) and the pulse level pertaining to the current path speed (P-0-0540; unit: rpm, 3 decimal places) via SERCOS monitor until a new axis oscilloscope recording shows no circle deviations worth mentioning.
    When that is the case, leave the pulse area that has been determined constant in the following process steps.
11. Now change the path feedrate (path speed) in the part program to the next value stored in P-0-0539 and record the sequence of the part program again as described in item 9.
12. Change the pulse level pertaining to the path speed (P-0-0540; unit: rpm, max. 3 decimal places) via SERCOS monitor until a new axis oscilloscope recording shows no circle deviations worth mentioning.
13. Repeat items 11. through 12. for the remaining speed values stored in P-0-0539.
14. Save all SERCOS parameters in the EEPROM of the drive or enter the data of parameters P-0-0538 and P-0-0540 determined until now into the relevant SERCOS files of the NC. The SERCOS files are transferred by the control unit to the drive during SERCOS initialization (for information on SERCOS, see page 4–67 ff.).

CAUTION
Danger of damage to the machine or the workpiece!
Do not initiate a control RESET during machining.

15. Start up the control unit and the drives.

6.8.2 Activating/deactivating

The NC activates the function if

- ”P-0-0536” has been entered in S-0-0024 (configuration list MDT),
- the ”freely configurable telegram” has been selected in S-0-0015 (telegram types parameter) and
- NC and drives perform a new start-up.

The NC deactivates the function if

- ”P-0-0536” has been removed from S-0-0024 (configuration list MDT) and
- NC and drives perform a new start-up.

6.9 Dynamic limit values

Function:

Serves to limit the speed and acceleration of connected drives within the control unit.
This way, situations within the NC which may lead to the dynamic overload of individual drives can be avoided in advance. For this purpose, the NC automatically reduces the path speed or path acceleration so that the defined limitations of all axes involved will not be exceeded.

Possible causes for dynamic overload:

- path feedrate too high.
  At a specified path feedrate, the required speeds of individual axes involved in the path may increase considerably.
- drive dimensions too small.
  Drives may be dynamically overloaded by peak loads.
- incorrect drive parametrization.
  Limitations in the drive (via SERCOS parameter) have not been parametrized correctly.
- unfavorable programming.
  e.g. many short machining blocks with upslope and downslope.

Possible effects of dynamic overload:

- contour distortions
- overtemperature in the drive
- increased machine wear
- servo error.

Restrictions:

- All setpoint values generated by the drive independently cannot be influenced via the limit values defined by the control unit (e.g. in case of ”Interpolation in the drive”, ”Drive-controlled referencing” or ”Halt” of the drive after the occurrence of an error)! In these cases, an appropriate parametrization within the drive is required.

Relevant MACODA parameters (MP):

1005 00001:	Maximum axis speed.
1010 00001:	Maximum axis acceleration.
1010 00011:	Maximum axis step change (in connection with G8)
1010 00012:	Maximum axis step change acceleration (relevant for changes in the interpolator cycle)
1040 00012:	Max. spindle speed of the gear range.
1040 00031:	Max. spindle acceleration.

Relevant G functions:

G6: Acceleration programming ON.
Supersedes the acceleration values defined in 1010 00001 with programmed or previously stored values.

G7: Acceleration programming OFF.
The values stored in MACODA are valid again.

G206: Saves the currently active acceleration values of all axes.

Relevant SERCOS parameters:

S-X-0091: Bipolar speed limit value.

S-0-0138: Acceleration bipolar.

P-0-0004: Halting mode with drive off.

6.9.1 Assigning

1. Make sure that the relevant SERCOS parameters have been parametrized correctly.
   The values of the SERCOS parameters should normally be set according to the maximum dynamic capacity of the respective drive. In this context, it is necessary to take the coupled axis mechanics as well as the maximum possible load into consideration sufficiently!
   If you have made any changes, save all SERCOS parameters in the EEPROM of the drive or enter their values into the relevant SERCOS files of the NC. The SERCOS files are transferred by the control unit to the drive during SERCOS initialization (for information on SERCOS, see page 4–67 ff.).
2. For axes:
   Configure MP 1005 00001 and MP 1010 00001. The limit values set in the drive must not be exceeded.
   For spindles:
   Configure MP 1040 00012 and MP 1040 00031. The limit values set in the drive must not be exceeded, taking into account the transmission ratio.
3. Configure MP 1010 00011 and MP 1010 00012. The speed and acceleration values set in item 2. must not be exceeded.

CAUTION
Danger of damage to the machine or the workpiece!
Do not initiate a control RESET during machining.

---

4. Start up the control unit and the drives.

7 Operation

7.1 Control reset

Function:

The function available as a standard resets the PNC to a precisely defined state. It can be initiated via softkey or interface signal.
We distinguish between two different ”control reset types”:

”Control reset”:	only has an effect for the channel where the function is initiated.
”System control reset”:	has an effect throughout the system.

”Control reset” initiates the following actions:

- Deceleration to standstill of all machining axes of the channel.
- Deselecting the active part program on the channel (in this context, also note the ”Automatic program selection” function on page 7–28 ff.).
- Resetting error states on the channel
- Calling the ”Power-up condition” function and interpretation of the init string contained in MP 7060 00020 (see page 3–6 ff.).
  All functions programmed ahead of the text ”#SysRes” are taken into account in this context.

”System control reset” initiates additional actions:

- Deceleration to standstill of all auxiliary axes and spindles in the system.
- Calling of functions which are programmed in MP 7060 00020 behind the text ”#SysRes”.
- Transfer of changed configuration parameter values.
  If the transfer of individual parameters is only possible by means of control unit start-up, the PNC will display a blinking ”?” in the NC status line. You can then call up detailed information via the info key.

Relevant interface signals:

NC-I0.0: ”System control reset” (global interface).
L/H signal edge initiates the function.

NC-I1.1: ”Control reset” (channel interface).
L/H signal edge initiates the function.

NC-O14.2: ”Control reset executed” (channel interface).
The signal is not set if the channel is not in power-up condition and a program has been selected or is active.
It will be set as long as the channel is in power-up condition and no program has been selected yet.
In case of an L/H signal edge, the control unit has executed the control reset. The PLC then has to reset to NC-I1.1.

7.1.1 Activating

CAUTION
Initiating ”Control reset” or ”System control reset” during machining may cause damage to the machine or workpiece!

- **Calling ”Control reset” via softkey:**
  AUTOMATIC ▶ CHANNEL CONTROL RESET.
- **Calling ”System control reset” via softkey:**
  DIAGNOSTICS ▶ RESET FUNCTIONS ▶ SYSTEM CONTROL RESET.
- **Calling ”Control reset” via interface signal:**
  L/H signal edge at NC-I1.1 of the respective channel interface.
- **Calling ”System control reset” via interface signal:**
  simultaneous L/H signal edge at NC-I1.1 of all assigned channel interfaces.
  – or –
  L/H signal edge at NC-I0.0 of the global interface.

7.2 Jogging in machine coordinates

Function:

Permits manual movement of axes in positive and negative direction.

The length of the traversing path can be influenced by

- the duration of the set interface signal ”Manual+” / ”Manual–”
- 4 incremental steps (1000, 100, 10, 1 increments) permanently stored in the system
- a user-defined incremental step (x increments).

In the ”manual, jog” mode it is possible to move all axes existing in the NC. In case of synchronous axes, this mode has a direct effect on the machine coordinates (MCS).

The individual options are first selected via the axis interface signals ”Manual feed/incremental bit x” (NC-I1.0 to NC-I1.3).
Subsequently, the traversing movement can be initiated via the axis interface signals ”Manual+” or ”Manual–”

It is possible to store the travel speeds separately for each of the three options mentioned above, using the configuration parameters.
The value used to accelerate to setpoint speed can be set jointly for all options via configuration parameters.

Preconditions:

- SERCOS communication has been established with the respective drive and is working properly.
  In this context, refer to the description of the following signals:
  NC-I8.x / 9.x: ”Drive inhibit” (axis interface)
  NC-I10.x / 11.x: ”Drive off” (axis interface)
  NC-O14.0: ”SERCOS system ready” (axis interface)
  NC-O14.1: ”Drive ready” (axis interface)
  NC-O14.2: ”Drive under control” (axis interface)
  NC-O15.0: ”Axis inhibited (test)” (axis interface)
- ”Manual” or ”Jog” mode has to be activated.
  In case of synchronous axes, note that their operating mode, in contrast to auxiliary axes, is not influenced by their axis interface but by the channel interface.
  The operating mode of a channel can be set either at the operating panel or by the PLC (binary coded via channel interface). The values permitted are specified using NC-I3.0 ”Mode selection by PLC” (channel interface). In case of auxiliary axes, their operating mode can be specified exclusively via the axis interface.
  In this context, refer to the description of the following signals:
  NC-I2.0: ”Mode selection bit 0” (channel interface:
  NC-I2.1: ”Mode selection bit 1” (channel interface)
  NC-I2.2: ”Mode selection bit 2” (channel interface)
  NC-I2.3: ”Mode selection bit 3” (channel interface)
  NC-I3.0: ”Mode selection by PLC” (channel interface)
  For auxiliary axes:
  NC-I1.6: ”Mode selection bit 0” (axis interface)
  NC-I1.7: ”Mode selection bit 1” (axis interface)

Jogging in machine coordinates

- The axis interface signals ”Handwheel assignment bit x” (NC-I2.0 and NC-I2.1) must not be set.
- The axis interface signal ”Feed inhibit” (NC-I12.x / NC-I13.x) must not be set.

Relevant MACODA parameters (MP):

1005 00003:	Manual feed slow.
1005 00004:	Manual feed medium.
1005 00005:	Manual feed fast.
1005 00006:	Manual feed rapid.
1005 00007:	Manual feed for defined steps.
1005 00008:	Manual feed for variable steps.
1010 00002:	Jog acceleration
1015 00001:	Axis resolution.
1015 00002:	Variable increment step.

Relevant interface signals:

NC-I0.0:	”Manual+” (axis interface).
NC-I0.1:	”Manual–” (axis interface).
NC-I0.3:	”Incremental step in inch” (axis interface).
NC-I1.0:	”Manual feed/ incremental bit 0” (axis interface).
NC-I1.1:	”Manual feed/ incremental bit 1” (axis interface).
NC-I1.2:	”Manual feed/ incremental bit 2” (axis interface).
NC-I1.3:	”Manual feed/ incremental bit 3” (axis interface).

Relevant CPL functions:

MCODS(...) see MCDOS (27,..) in the CPL Programming Manual

Jogging in machine coordinates

7.2.1 Assigning

1. Use the table below to set the relevant configuration parameters:

Jog mode [1]	Length of the traversing path [2]	effective speed	effective acceleration
Rapid	dependent on the duration of the set interface signal ”Manual+” / ”Manual–”	MP 1005 00006	MP 1010 00002. If assigned ”0”, the value from MP 1010 00001 is applicable.
Fast		MP 1005 00005	
Medium		MP 1005 00004	
Slow		MP 1005 00003	
x increments	per L/H edge at ”Manual+” or ”Manual–”: MP 1015 00002 * MP 1015 00001	MP 1005 00008	
1000 increments	per L/H edge at ”Manual+” or ”Manual–”: 1000 * MP 1015 00001	MP 1005 00007	
100 increments	per L/H edge at ”Manual+” or ”Manual–”: 100 * MP 1015 00001		
10 increments	per L/H edge at ”Manual+” or ”Manual–”: 10 * MP 1015 00001		
1 increment	per L/H edge at ”Manual+” or ”Manual–”: 1 * MP 1015 00001		

(1) Selection via axis interface signals ”Manual feed/incremental bit ”x” (NC-I1.0 to NC-I1.3).

(2) If the signal ”Inch incr. step” (NC-I0.3) is set at the axis interface of a linear axis, the PNC will multiply the value entered in MP 1015 00001 by the factor 2.54.

2. Initiate a system control reset (DIAGNOSTICS ▶ RESET FUNCTIONS ▶ SYSTEM CONTROL RESET).

7.2.2 Activating

1. If you wish to jog synchronous axes, select ”Jog mode” on the operating panel:
   MANUAL ▶ JOG MODE

☞ **Selection via the operating panel is not possible if the signal ”Mode selection by PLC” (NC-I3.0) has been set at the channel interface.**
In this case, the PLC must provide the ”Manual” operating mode via the channel interface signals ”Mode selection bit x” (NC-I2.0 to 2.3).

2. If you wish to jog asynchronous axes, make sure that the ”Manual” mode is provided at the axis interface of the respective axis via the signals ”Mode selection bit x” (NC-I1.6 and 1.7).

Jogging in machine coordinates

3. Make sure that the axis interface signals ”Handwheel assignment bit ”x” (NC-I2.0 and NC-I2.1) have not been set.
4. Select the desired jog mode.
   Selection via axis interface signals ”Manual feed/incremental bit x” (NC-I1.0 to NC-I1.3).

☞ **Please note that the signal ”Inch incr. step” (NC-I0.3) influences the length of the traversing path in the increment modes!**
If it has been set, the PNC multiplies the value entered in MP 1015 00001 by the factor 2.54!

5. Initiate the traversing movement of the relevant axes via the axis interface signals ”Manual+” or ”Manual–”.

7.3 Jogging in workpiece coordinates

Function:

Using this function, you can jog coordinates/pseudo coordinates in workpiece coordinates (WCS) and in Z direction in tool coordinates (TCS). The **"Manual, jogging in workpiece coordinates"** mode is available for this purpose.

The following **coordinates** may be jogged:

- with axis transformation switched off: all pseudo coordinates (axes)
- with active 5 axis transformation: all linear orientation coordinates, the TCS Z direction and the pseudo coordinates (axes).

The coordinate/pseudo coordinate to be jogged is selected by the PLC via NC block specification (see program module -B04SATZV) using the JogWCSSelect.. NC function (see also "Relevant NC functions").
As an alternative, the coordinates can be selected by any part program, e.g. a CPL program.

Only one coordinate/pseudo coordinate can be moved at any given point in time.

The "Manual, jogging in workpiece coordinates" mode must be selected prior to jogging:

- directly by the PLC (operating mode 14) when mode selection by PLC is active
- via the operator interface using the "Jog" softkey if it has been initialized in MACODA 6001 00030 for jogging worpiece coordinates.

Jogging options:

- **Continuous jogging**: for jogging, the maximum positive or negative end point of the selected coordinate is specified. The movement is stopped when you release the jog key.
  The maximum end point of a coordinate is calculated by the NC on the basis of the axis end switches, taking into account the active transformation.
- **Incremental jogging**: When the jog key is actuated, the control unit will check whether it is possible to traverse the selected path increment.
  The instruction will be ignored if it is not possible to traverse the selected path increment.

Jogging in workpiece coordinates

Differences to the ”Manual, jog” mode (see Section 7.2):

In contrast to the “Jogging in workpiece coordinates” it is possible to move all axes existing in the NC in the ”Manual, jog” mode. When synchronous axes are jogged, this has a direct effect on the machine coordinates (MCS).
If a 5 axis transformation is active, however, it is practically impossible to move the transformed z coordinate (e.g. diagonally in space) in machine coordinates. In this case, up to 3 machine axes (X, Y, Z) have to be jogged simultaneously to be able to traverse the z coordinate.

Restrictions:

- Select the **5 axis transformation with linear orientation** (MP 1030 00110 ”Axis transformation type” = 3232101 or 3032101), even if the 5 axis transformation with vector orientation (MP 1030 00110 = 3232201) was active previously.
  In this context, the name of the part program (initialization program), which selects an axis and/or coordinate transformation in connection with CPL commands, can be entered in MP 7500 11000.
  This program will **always** be executed at the start of the first jog movement after selection of the operating mode.

Relevant MACODA parameters (MP):

1010 00002:	Jog acceleration
6001 00030:	Configuration of the jog mode softkey
7050 01000:	Jogging WCS: manual feeds
7050 01010:	Jogging WCS: feedrate and incremental step, system axis number
7050 01020:	Jogging WCS: resolution of an increment
7050 01030:	Jogging WCS: variable increment step
7050 01100:	Jogging WCS: program name for the standard configuration

Relevant NC functions:

JogWCSSelect JWSCHAN<Channel no.> JWSCOORD<Coordinate no.> {JWSFEED<F value>} {JWSSTEP<Increments>}

Selection of coordinate for jogging in workpiece coordinates

Jogging in workpiece coordinates

The following applies:

JWSCHAN<Channel no.>	Number of the channel the coordinate was selected for.
JWSCOORD<Coordinate no.>	1...8: Number of the coordinate 9: TCS Z direction: • the TCS Z direction exists in case of active 5 axis transformation only. • a compensation in TCS-Z direction is converted into a movement of the linear working range coordinates (x,y,z).
{JWSFEED<F value>}	**Optional,** **Default:** corresponds to the feedrate speed of the axis interface (see MP 705001020). Unit: mm/min or degrees/min (G71), or inch/min or degrees/min (G70)
{JWSSTEP<Increments>}	**Optional,** **s**election of incremental jogging and simultaneous specification of the step size in increments. **Default:** incremental and continuous jog mode of the axis interface (see MP 7050 01020).

☞ **JWSSTEP may only be programmed together with JWSFEED.**

Relevant interface signals:

Channel NC-I0.6:	Workpiece coordinates manual +
Channel NC-I0.7:	Workpiece coordinates manual –
Channel NC-I2.0 .. 2.3:	Mode selection

7.3.1 Assigning

1. Create the initialization program for switching over into the 5 axis transformation with linear orientation (if required), when jog mode is selected.
2. Edit the program module -B04SATZV for the NC block specification to select a coordinate for jogging.
3. Link the relevant interface signals.
4. Enter the values for the relevant MACODA parameters:
   - MP 7050 0100 – MP 7050 01030
   - If you wish to release the jog mode via softkey, enter the value 2 or 3 in MP 6001 00030.

Jogging in workpiece coordinates

- The acceleration capacity of the jog movement results from the values in MP 1010 00002 ”Jog acceleration”.
- The initialization program specified in MP 7050 01100 must be stored in the search path.

7.3.2 Activating

1. Select the ”Jogging in workpiece coordinates” mode via PLC or softkey.
2. Select the desired coordinate (specification of the JogWCSSelect NC function, e.g. by the PLC via NC block specification).
3. Select the “speed range” or the “step size” (see also MP 7050 01020).
4. The movement starts when the channel interface NC-I0.6 or NC-I0.7 has been set.

Example:

1010 00002	[1] [2] .. [64]	Jog acceleration for the axes relevant for this channel
6001 00030	[1] 3	”Jog” softkeys exist
7050 01000	[1] .. [6]	Set jog feedrate values as required
7050 01010	[1] 1	The step size is derived from the 1st system axis
7050 01020	[1] 1.000	1 increment corresponds to 1 µm
7050 01030	[1] 2000	The variable increment step is 2000 incr.
7050 01100	[1] *Initstri* [2] *ngForJog* [3] *WCS* [4]	The initialization program is called: InitstringForJogWCS

Example for an initialization program ” InitstringForJogWCS”:

```
10   IF @AXTRAFO%=1 THEN
N20  Coord(2)                ;The axis transformation 2 must be of
                             type 3232101 or 3032101
30   ELSE
N40  Coord(0)
50   ENDIF
```

7.3.3 Deactivating

- The continuous movement is stopped by releasing the respective jog key when the channel interface NC-I0.6 or NC-I0.7 has been reset.
- Leaving the ”Jogging in workpiece coordinates” mode terminates every jog movement.
- Channel control reset and system control reset discontinue an active jog movement.

7.4 Handwheel

Function:

You can connect up to two digital handwheels to manually traverse the axes in manual mode. The communication takes place via CAN-Bus (only PNC-R) or PROFIBUS-DP, a mixed operation is permitted, too.

The length of the traversing path can be influenced by:

- 4 incremental steps (1000, 100, 10, 1 increments) permanently stored in the system
- a user-defined incremental step (x increments).

The incremental steps are always interpreted as evaluation factor for the scale division of the handwheel. Moving the handwheel by 1 scale mark, for example, will move the respective axis by the corresponding number of increments.
The step size can be switched over via the axis interface signals ”Manual feed/incremental step bit x” (NC-I1.0 to NC-I1.3).

The travel speed is derived by approximation from the rotation speed of the handwheel and limited to the rapid mode of the axis.

Preconditions:

- Module osa switch (machine operating panel with CAN-Bus), or module osa switch dp (machine operating panel with PROFIBUS-DP), or handwheel with direct PROFIBUS-DP connection.
- The number of pulses per revolution is adjusted.
  - osa switch: The digital handwheel has to provide 100 pulses per revolution (e.g. Messrs. Euchner, HKG 100X100A05). No further adjustment.
  - osa switch dp: The pulses per revolution of the handwheel are always multiplied by four. This value must be set in the MP 9060 00001.
    (e.g. 9060 00001 = 400 for a handwheel with 100 pulses per revolution).

- Correct connection:
  - osa switch: handwheel 1 has been connected correctly at female D shell connector ”St3”. The machine operating panel is connected to X51 of the osa master module via ”St2”.
  - osa switch dp: handwheel 1 has been connected correctly at female D shell connector ”St3”. The machine operating panel is connected to X71 of the PNC-P plug-in card or to X51 of the module “osa dc I/O” or “osa dc I/O ana” via ”St2” and configured via WinDP.
  - Handwheels which are connected without machine operating panel directly via PROFIBUS-DP have to transmit values to the PNC representing the counter reading with a sign, not incremental values.

Handwheel

For information on the connection of the machine operating panels, please refer to the manuals ”PNC-R Connectivity Requirements”, ”PNC-P Connectivity Requirements” and“PLC Project Planning”.

CAUTION
Damage to the electrical/electronic equipment possible!
Switch off system before you pull out or plug in connecting lines or before you install a second logic card!

- For communication via PROFIBUS-DP, the handwheel counter reading has to be transmitted to the NC cyclically via the APS module “B14HANDW”.
- SERCOS communication has been established with the respective drive and is working properly.
  In this context, refer to the description of the following signals:
  NC-I8.x / 9.x: ”Drive inhibit” (axis interface)
  NC-I10.x / 11.x: ”Drive off” (axis interface)
  NC-O14.0: ”SERCOS system ready” (axis interface)
  NC-O14.1: ”Drive ready” (axis interface)
  NC-O14.2: ”Drive under control” (axis interface)
  NC-O15.0: ”Axis inhibited (test)” (axis interface)
- ”Manual” or ”Jog” mode has to be activated.
  In case of synchronous axes, note that their operating mode, in contrast to auxiliary axes, is not influenced by their axis interface but by the channel interface.
  The operating mode of a channel can be set either at the operating panel or by the PLC (binary coded via channel interface). The values permitted are specified using NC-I3.0 ”Mode selection by PLC” (channel interface). In case of auxiliary axes, their operating mode can be specified exclusively via the axis interface.
  In this context, refer to the description of the following signals:
  NC-I2.0: ”Mode selection bit 0” (channel interface:
  NC-I2.1: ”Mode selection bit 1” (channel interface)
  NC-I2.2: ”Mode selection bit 2” (channel interface)
  NC-I2.3: ”Mode selection bit 3” (channel interface)
  NC-I3.0: ”Mode selection by PLC” (channel interface:
  For auxiliary axes:
  NC-I1.6: ”Mode selection bit 0” (axis interface)
  NC-I1.7: ”Mode selection bit 1” (axis interface)
- The axis interface signal ”Feed inhibit” (NC-I12.x / NC-I13.x) must not be set.

Relevant MACODA parameters (MP):

1005 00002: Rapid traverse rates of axes.
1015 00001: Axis resolution.
1015 00002: Variable increment step.
9060 00001: Number of increments per revolution.
9060 00002: Field bus assignment (PNC-R only).

Relevant interface signals:

NC-I0.3: ”Incremental step in inch” (axis interface).
NC-I0.5: ”Handwheel direction of rotation” (axis interface)
NC-I1.0: ”Manual feed/ incremental bit 0” (axis interface).
NC-I1.1: ”Manual feed/ incremental bit 1” (axis interface).
NC-I1.2: ”Manual feed/ incremental bit 2” (axis interface).
NC-I1.3: ”Manual feed/ incremental bit 3” (axis interface).
NC-I2.0: ”Handwheel assignment bit 0” (axis interface)
NC-I2.1: ”Handwheel assignment bit 1” (axis interface)

7.4.1 Assigning

1. Use the table below to set the relevant configuration parameters:

Incremental step (1)	**Length of the traversing path** (2)	**effective speed**
x increments	per scale mark on the handwheel: MP 1015 00002 * MP 1015 00001	dependent on the rotation speed of the handwheel. Is always limited to the value in MP 1005 00002 however.
1000 increments	per scale mark on the handwheel: 1000 * MP 1015 00001	
100 increments	per scale mark on the handwheel: 100 * MP 1015 00001	
10 increments	per scale mark on the handwheel: 10 * MP 1015 00001	
1 increment	per scale mark on the handwheel: 1 * MP 1015 00001	

(1) Selection via axis interface signals ”Manual feed/incremental bit x” (NC-I1.0 to NC-I1.3).

(2) If the signal ”Inch incr. step” (NC-I0.3) is set at the axis interface of a linear axis, the PNC will multiply the value entered in MP 1015 00001 by the factor 2.54.

2. Initiate a system control reset (DIAGNOSTICS ▶ RESET FUNCTIONS ▶ SYSTEM CONTROL RESET).

7.4.2 Activating

1. If you wish to move synchronous axes using the handwheel, select ”Jog mode” on the operating panel:
   MANUAL ▶ JOG MODE

☞ **Selection via the operating panel is not possible if the signal ”Mode selection by PLC” (NC-I3.0) has been set at the channel interface.**
In this case, the PLC must provide the ”Manual” operating mode via the channel interface signals ”Mode selection bit x” (NC-I2.0 to 2.3).

2. If you wish to traverse asynchronous axes via handwheel, make sure that the ”Manual” mode is provided at the axis interface of the respective axes via the signals ”Mode selection bit x” (NC-I1.6 and 1.7).
3. Select the desired incremental step.
   Selection via axis interface signals ”Manual feed/incremental bit x” (NC-I1.0 to NC-I1.3).
4. Select the desired handwheel.
   Selection via axis interface signals ”Handwheel assignment bit x” (NC-I2.0 to NC-I2.1).
5. Specify using the NC-I0.5 axis interface signal how the control unit is to interpret the sense of rotation of the handwheel.
   NC-I0.5 has not been set:
   Turning the handwheel clockwise results in a positive traversing movement.
   NC-I0.5 has been set:
   Turning the handwheel clockwise results in a negative traversing movement.

☞ **Please note that the signal ”Inch incr. step” (NC-I0.3) influences the length of the traversing path!**
If it has been set, the PNC multiplies the value entered in MP 1015 00001 by the factor 2.54!

6. Initiate a traversing movement by turning the handwheel.

7.4.3 Deactivating

☞ **An axis can only be moved using the jog key (jog axis) once the assigned handwheel has been deactivated!**

1. Reset the respective axis interface signal ”Handwheel assignment bit x” (NC-I2.0 to NC-I2.1):
   Deactivate handwheel 1: reset NC-I2.0.
   Deactivate handwheel 2: reset NC-I2.1.

7.5 Online correction with handwheel (in workpiece coordinates)

Function:

The “online correction” is used with active or inactive part program for the online

- correction of a **position** or **orientation** using handwheel in the workpiece coordiante system (WCS) of a channel.
- traversing of the **position of the tool longitudinal axis** in TCS Z direction of the tool in the tool coordinate system (TCS) (no tool compensation!).

Please note especially that, with active axis transformation of several coordinates, the position/orientation of an axis/coordinate can be changed **while the axis is moving** by a compensation coming from the handwheel.

☞ **In the ”Manual” (jog mode) and ”Manual approaching the reference point” modes, it is not possible to perform an online correction (for more information, please refer to Handwheel mode in Section 7.4).**

DANGER
Danger of collision with the workpiece in case of online correction!
Danger of damage to persons and equipment by the breaking of tools and workpieces!
Activate the online correction with great care, in particular in case of active program and moving axes!

Once the correction has been activated via handwheel, it remains effective until it is deactivated. This means that the correction also has an effect on the subsequent programmed positions in an active program. In this case, the programmed positions are offset by the value of the correction.

Example:

The position of the tool is compensated online in TCS Z direction using the handwheel. In the next block, a position in X, Y is approached. In this block, the tool tip is offset in tool longitudinal axis by the value of the online correction.

The online correction of the handwheel has an effect on the following coordinates/axes (examples of coordinate and axis identifiers x, y, z, theta, phi).

Online correction

Transformation	**Axes (pseudo coordinates)**	**WCS linear coordinates**	**WCS orientation coordinates**	**TCS coordinates**
none	all	–	–	–
5 axis transformation with rotary axis programming	all	x, y, z	–	Z direction
5 axis transformation with vector orientation	all	x, y, z	–	Z direction
5 axis transformation with linear orientation	all	x, y, z	theta, phi	Z direction

The active correction values of an activated online correction are transformed when activating, deactivating and changing an

- axis transformation
- coordinate transformation (inclined plane).

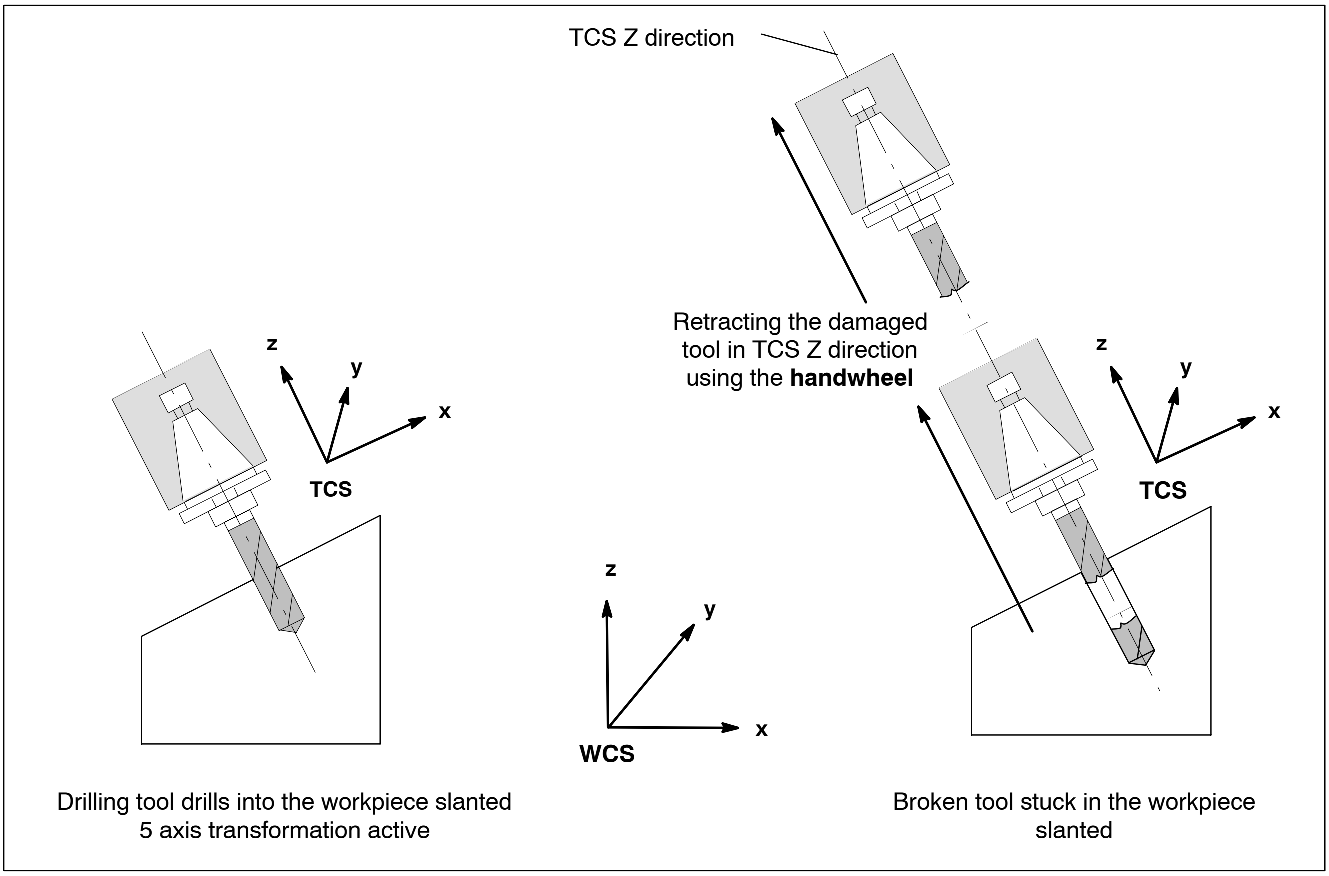
A connection which may still exist at this point in time between coordinate/axis and handwheel is disbanded, i.e. the online correction is deselected.

In case of axis transfer (see Section 4.9), the correction memory is deleted and the online correction is deselected, if the axis transferred is linked with the handwheel.
An axis just taken over always has an empty online correction memory.

Application:
A total of up to 2 digital handwheels can be operated at the PNC-P (PROFIBUS-DP) or the PNC-R (PROFIBUS-DP, CAN). In addition, 1 handwheel can be connected per SERCOS drive.

The online correction can be used, for instance in a 5 axis transformation, to retract a damaged drilling tool stuck in the workpiece diagonally in TCS Z direction with the aid of the handwheel. In order to continue working at the same position with a new tool, the handwheel is used to “traverse back“ on the old online correction until the new tool has the old TCS Z value (zeroset first, so that the different tool length corrections are taken into account, if applicable).

Online correction



Restrictions:

- An online correction is not possible in the operating modes ”Manual” (jog mode) and ”Manual approaching the reference point”.
- Machine-oriented absolute position:
  G76 traverses to an offset position, i.e. the online correction is not calculated back.
- The CPL functions PPOS and CPROBE do not take into account the correction values of the online correction.
- Probe
  G75 measures the correct position. Use the CPL function **PROBE** to read this measuring value.
- Measuring at the dead stop
  G375 measures the correct position. Use the CPL function **PROBE** to read this measuring value.
- Limit switch
  The NC does **not** check whether a coordinate position generated by the online correction exceeds the software limit switches. For this purpose, activate the “limit switch check” in the SERCOS drive.

Online correction

Relevant MACODA parameters (MP):

7050 00900: Online correction in workpiece coordinates (WCS) available

7050 00910: Online correction WCS: sensor selection, device group

7050 00910: Online correction WCS: sensor selection, number of the device

7050 00914: Online correction WCS: sensor selection, identification

7050 00920: Online correction WCS: sensor increments –> mm factor
The following increments are permitted:
1 increment –> 0.001 mm
10 increments
100 increments
1000 increments

7050 00921: Online correction WCS: sensor increments –> mm divisor
The following increments are permitted:
1 increment –> 0.001 mm
10 increments
100 increments
1000 increments

7050 00926: Online correction WCS: incremental step, system axis number

7050 00940: Online correction WCS: max. speed in mm/min

7050 00941: Online correction WCS: max. acceleration in m/sec^2

For connection of the handwheel via PROFIBUS-DP to the PNC-P card or via PROFIBUS-DP/CAN to the PNC-R hardware (osa master):

9060 00001: Number of increments per revolution
100: CAN (only PNC-R)
400: PROFIBUS-DP

060 00002: Selection of the interface at the PNC-R:
1: CAN
2: PROFIBUS-DP

Online correction

Relevant NC functions:

HWOCON OCONCH<Channel no.> OCCOORD<Coordinate no.> [OCSTEP<Increments>]

Activating the online correction via PLC or via switching functions or from any channel.

HWOCON OCCOORD<Coordinate no.> [OCSTEP<Increments>]

Activating the online correction on the own channel.

The following applies:

<Channel no.> Number of the channel the online correction was selected for.

<Coordinate no.> 1...8: Number of the coordinate
9: TCS Z direction:
- the TCS Z direction exists in case of active 5 axis transformation only.
- a correction in TCS Z direction is converted into a movement of the linear working range coordinates (x,y,z).

<Increments> Specifying the incremental step
Default:
Incremental step from the axis interface
(see MP 7050 00926)

HWOCOFFOCOFFCH<Channel no.>

Deactivating the online correction via PLC or via switching functions or from any channel.

HWOCOFF **Deactivating** the online correction on the own channel

The following applies:

<Channel no.> Number of the channel the online correction was selected for.

HWOCDEL **Deactivating** the online correction **and deleting** the correction values

Relevant interface signals:

at the channel interface

NC-I2.6: Online correction enable
NC-I2.7: Online correction direction of rotation

Online correction

Relevant SERCOS parameters:

The following parameters are only relevant if the digital handwheel is connected directly at the SERCOS drive:

S-0-0016:	Configuration list DT. Determines which SERCOS parameters are to be transmitted to the NC cyclically. The SERCOS parameter specified in MP 7050 00714 has to be contained here (e.g. P-0-0553 "Position actual value 2 incremental").
S-0-0115:	Position encoder type (external encoder): rotation encoder
S-0-0117:	Rotation encoder 2 resolution (external encoder). Only necessary if encoder type "rotation encoder" has been set in S-0-0115.
S-0-0257:	Pulse multiplication-2 (external encoder).
P-0-0552	Evaluation external encoder 1: activated

Relevant CPL functions:

MCODS(...) see MCODS (27,..) in the CPL Programming Manual

7.5.1 Assigning

1. Connect the handwheel with the PNC.
2. Enter the relevant values into MP 7050 00900 through 7050 00941.

   Example:
   Online correction via handwheel 1 on channel 1

7050 00900	[1] 1	Online correction enable
7050 00910	[1] 3	Device group 3: digital handwheel with PROFIBUS-DP or CAN
7050 00912:	[1] 1	Handwheel no.: 1
7050 00914:	[1]	for 7050 00910 [1] =3 no meaning
7050 00920:	[1] 1.0	1 handwheel increment –> 0.001 mm
7050 00921:	[1] 1000.0	
7050 00926:	[1] 1	The incremental step is derived from the 1st system axis
7050 00940:	[1] 100.0	The maximum speed of the correction is limited to 100 mm/min.
7050 00941:	[1] 99.999	No additional limitation of acceleration

3. Link the channel interface signals NC-I2.6 and NC-I2.7 in the PLC.
4. Configure the relevant SERCOS parameters when the handwheel is directly connected to the drive.

7.5.2 Activating

1. Selecting the online correction:
   - in the operator interface in the “Manual” main menu using PREPARING ▶ ONLINE CORRECTION, or
   - via the PLC by entering the NC function HWOCON
2. Enabling the function by setting the channel interface signal NC-I2.6.

☞ **When the online correction is enabled on a channel, the correction performed on that channel is automatically deactivated.**

7.5.3 Deactivating

The online correction can be deactivated in different ways:

- **Interrupting:**
  Canceling the interface signal NC-I2.6 ”Online correction enable”.
- **Switching off:**
  The NC function HWOCOFF switches the online correction off.
- **Deleting:**
  The NC function HWOCDEL switches the online correction off and deletes the correction memory of this channel.
- **Deactivating via operator interface:**
  Using the softkey ONLINE CORRECTION in the “Manual” main menu, it is possible to switch the online correction on, over and off. In addition, online correction values may be deleted. For detailed information, please refer to the manual ”Operating instructions, standard operator interface”.
- **Control unit start-up**
  The online correction is inactive following a start-up of the control unit. The online correction memories are deleted.
- **Control reset**
  - Has no direct effect on online correction.
  - The contents of the correction memories is maintained.
  - An online correction is only switched off if
    – an axis transfer (G510..G513), or
    – a changing of the coordinate transformation (inclined plane) or
    – a changing of the axis transformation is performed in the init string (MP 7060 00020).
  - The online correction values are only deleted if HWOCDEL has been entered in the init string (MP 7060 00020).

7.6 Override

Function:

Influences the specified

1. path speeds within a channel and
   speeds of the machining axes in manual mode
2. speeds of auxiliary axes
3. spindle speeds.

For this purpose, the absolute (e.g. programmed) values are continuously converted within the control unit in dependence on separate, currently set percentage values and become effective on the machine.

The respective percentage value results from the signals which

- are active at the channel-related digital interface
  (for speeds according to item 1. above)
- are active at the axis-related digital interface
  (for speeds according to item 2. above)
- are active at the spindle-related digital interface
  (for speeds according to item 3. above)

The signals can be converted to percentage values as follows:

- via ”Step-related value”:
  32 individual parameters of a MACODA parameter are assigned percentage values (value range: 0 to 150%). The control unit selects one of these individual parameters in dependence on the active bit pattern (width: 5 bits = max. 32 steps).
  This kind of conversion may, for instance, be used in connection with override potentiometers which are designed as digital rotary switches.
- via ”Direct input”:
  the control unit interprets the active bit pattern directly as binary number (width: 16 bits) and converts it with a weighting of 0.01% into a percentage number with 2 decimal positions.
  Thus it is possible to enter override values in the range from 0.00 to 655.35% with a resolution of 0.01%.

Restrictions:

- ”Direct input” with override values is only possible at the channel-related digital interface. Override values for auxiliary axis speeds or spindle speeds can only be set using ”Step-related input”.
- The response time following a change of override values depends on the PLC cycle time (guide value: approx. 20 ms)

Override

Relevant MACODA parameters (MP):

7030 00010:	Feed override for channel. Provides a channel-specific definition of the maximum permitted override value and the override values for the ”Step-related input”. Under channel ”0”, the respective values are parametrized globally for all auxiliary axes in the system. If the override values for the ”Step-related input” are all assigned ”0”, the control unit switches over to ”Direct input”.
1003 00100:	Override for auxiliary axis. Determines whether the override function for auxiliary axes is to be activated or deactivated.
1040 00041:	Spindle override in 32 steps. Provides a spindle-specific definition of the maximum permitted override value and the override values for the ”Step-related input”.

Relevant G functions:

G63:	Feedrate 100%. Deactivates the override function on the current channel.
G66:	Feedrate 100% off. Activates the override function on the current channel.

Relevant interface signals:

at the channel interface

NC-I1.2:	”1^{st} Override 100%”.
NC-I4.x:	”Override bit x”; weighting: 2^x (for step-related input).
NC-I5.x:	”Override bit x”; weighting: 2^{x+8} (for direct input: NC-I4.x and NC-I5.x act jointly).
NC-O18.3:	”Override 0%” .
NC-O18.4:	”Override 100%” .
NC-O20.7:	”G63 active”.

at the axis interface

NC-I0.2:	”Axis override 100%” .
NC-I4.x:	”Override bit x”; weighting: 2^x .
NC-O15.1:	”Axis Override 0%”.
NC-O15.2:	”Axis Override 100%”.

at the spindle interface

NC-I2.x:	”Override bit x”; weighting: 2^x .
NC-I3.3:	”Spindle override 100%”.
NC-O15.3:	”Spindle override 100%”.
NC-O17.2:	”Spindle override 0%”.

Relevant CPL functions:

SD(2,..)	read-only access to potentiometer data on the active channel
MCODS(...)	see ”Potentiometers” under MCDOS in the CPL Programming Manual

7.6.1 Assigning

...for path feed and machining axes in manual mode

1. Adapt MP 7030 00010 (feedrate override for channel) for each assigned machining channel (from channel 1 on) as follows:
   Individual parameter [1]:
   Enter maximum override value to which an input via channel interface is restricted (in %).

   Individual parameters [2] to [33]:
   Enter override values for the ”Step-related input” (value range: 0 to 150%).
   If ”Direct input” is desired, assign the value ”0.00” to [2] through [33].

2. Verify that all channel-related interface signals are correctly generated or evaluated by the PLC and transmitted to the corresponding channel interface of the control unit.

CAUTION
Danger of damage to the machine or the workpiece!
Do not initiate a control reset during machining.

3. Initiate a control reset.

...for auxiliary axes

1. Adapt MP 7030 00010 (feedrate override for channel) for channel 0 as follows:
   Individual parameter [1]:
   Enter maximum override value to which an input via axis interface is restricted (in %).

   Individual parameters [2] to [33]:
   Enter override values for the ”Step-related input” (value range: 0 to 150%).

2. Verify that all axis-related interface signals are correctly generated or evaluated by the PLC and transmitted to the corresponding axis interface of the control unit.
3. Release the override function for the desired auxiliary axes in MP 1003 00100. For this purpose, assign the value ”00” to the relevant individual parameters (no release: enter ”–01”).

CAUTION
Danger of damage to the machine or the workpiece!
Do not initiate a control reset during machining.

4. Initiate a control reset.

...for spindles

1. Adapt MP 1040 00041 (spindle override in 32 steps) for each assigned spindle as follows:
   Individual parameter [1]:
   Enter maximum override value to which an input via spindle interface is restricted (in %).

   Individual parameters [2] to [33]:
   Enter override values for the ”Step-related input” (value range: 0 to 150%).

2. Verify that all spindle-related interface signals are correctly generated or evaluated by the PLC and transmitted to the corresponding spindle interface of the control unit.

CAUTION
Danger of damage to the machine or the workpiece!
Do not initiate a control reset during machining.

3. Initiate a control reset.

7.6.2 Activating

Precondition:
- The corresponding override function has been assigned.

...for path feed and machining axes in manual mode

Precondition:
- The required interface signals are evaluated or generated by the PLC and transmitted to the corresponding channel interface of the control unit.

★ Reset NC-I1.2 (1st override 100%)
or – if NC-O20.7 (G63 active) is linked with mit NC-I1.2 –
program G66.

Override

...for auxiliary axes

Precondition:

- The required interface signals are evaluated or generated by the PLC and transmitted to the corresponding axis interface of the control unit.

★ Reset NC-I0.2 (axis override 100%).

...for spindles

Precondition:

- The required interface signals are evaluated or generated by the PLC and transmitted to the corresponding spindle interface of the control unit.

★ Reset NC-I3.3 (spindle override 100%).

7.6.3 Deactivating

...for path feed and machining axes in manual mode

Precondition:

- The required interface signals are evaluated or generated by the PLC and transmitted to the channel interface of the control unit.

★ Set NC-I1.2 (1st override 100%)
or
program G63.

...for auxiliary axes

Precondition:

- The required interface signals are evaluated or generated by the PLC and transmitted to the corresponding axis interface of the control unit.

★ Set NC-I0.2 (axis override 100%).

...for spindles

Precondition:

- The required interface signals are evaluated or generated by the PLC and transmitted to the corresponding spindle interface of the control unit.

★ Set NC-I3.3 (spindle override 100%).

7.7 Automatic program reselection

Function:

When a program is selected for the first time after control unit start-up, one program will always remain selected on this channel in ”Automatic” mode.

Please note:

- A program aborted in ”Automatic” mode is immediately selected again unless aborting the program was initiated by selecting another program (e.g. following control reset on the channel).
- If a program is aborted by changing the operating mode on the channel, the control unit will select the aborted program again as soon as you switch to ”Automatic” mode on this channel.

The selected program can then be started via the channel-specific interface signal ”Cycle start” (NC-I1.0) if the signal ”NC ready” (NC-O16.0) has been set at the channel interface.

Relevant MACODA parameters (MP):

7060 00410: Automatic program reselection

Relevant interface signals:

at the channel interface

NC-I7.0:	Automatic program reselection off (high) or on (low)
NC-I1.0:	Cycle start
NC-O16.0:	NC ready

7.7.1 Activating/deactivating

1. Assign in MP 7060 00410 for the respective channel
   the value ”1” to activate the automatic program reselection
   – or –
   the value ”0” to deactivate the automatic program reselection.

CAUTION
Danger of damage to the machine or the workpiece!
Do not initiate a control RESET during machining.

---

2. Initiate a control RESET.

☞ **When this function is activated, an error detected by the control unit during program selection cannot be eliminated by control reset (the error will occur again and again due to the automatic reselection).**
Eliminate states of this kind by
– selecting a fault-free program,
– correcting the defective program, or
– correcting the possibly defective program selection.

7.8 Canceling the distance to go

Function:

”Cancel distance to go” rejects all blocks already prepared by the block preparation at the time of function initiation and processes them again including the block where the interruption occurred. The channel then changes to state READY.

If, for instance, you want to make sure that the change of an input signal is effective during the execution of the program for blocks which have already been prepared, the PLC can interrupt program execution at an appropriate point using ”Feed hold”, execute ”Cancel distance to go” and continue machining using ”Cycle start”.

☞ **The number of blocks held in block preparation can be set using MP 7060 00110.**

Restrictions:

- ”Feed hold” and thus standstill of the axes is a precondition for ”Cancel distance to go”.
- ”Cancel distance to go” is only effective in the modes ”Automatic”, ”Single block”, ”Single step” and ”Program block”.

Relevant interface signals:

at the channel interface

NC-I3.3: ”Canceling the distance to go”
L/H signal edge initiates the function.

NC-O16.3: ”NC ready”
Is set as soon as ”Cancel distance to go” has been executed and the channel is in READY state again.
”NC ready” is the precondition for the start using ”Cycle start”.

NC-I1.0: ”Cycle start”
The selected program is continued at the end of the current block.
Precondition: Signal ”NC ready” (NC-O16.0) is set at the channel interface.

7.8.1 Activating

1. Activate ”Feed hold”.
2. Use L/H signal edge at NC-I3.3 of the respective channel interface to initiate ”Cancel distance to go”.
3. Continue program via ”Cycle start”.

7.9 Remove and return to the contour

Function:

The function allows for removing and returning to the contour following the interruption of a part program using ”Feed hold”. This may be necessary e.g. because of a broken tool, for manual workpiece/tool measurement or for a visual check.

For returning to the contour, you may choose:

- the return mode
  (**how** the NC is supposed to return to the contour: manually, automatically, by single block)
- the return point
  (**to which point** is the NC supposed to return to the contour: point of interruption, start of block, end of block)

☞ **The desired return mode and return point can be preset in MACODA via MP 7060 00310 and MP 7060 00320.**
You can also set the desired return strategy via operator interface (“Manual” main menu, softkey sequence REMOVE FROM CONTOUR ▶ FIX STRATEGY).
The options set here supersede MP 7060 00310 or MP 7060 00320 and are effective until the next control unit start-up.
For detailed information on the effects of the individual options, please refer to Section 7.9.1, page 7–32 ff.

When a part program is interrupted by ”Feed hold”, it is possible to use the function ”Remove and return to the contour”. The mode ”Remove from contour” has to be activated first (see page 7–36).

From this time on, the control unit will ”remember” all traversing movements that you initiate using the axis-related input signals ”Manual+” or ”Manual–”. In case of axis standstill, the control unit outputs the channel interface signal NC-O16.4 ”Ready for re-entry”.

☞ **Via MP 7060 00330 you can configure the maximum number of traversing movements to be remembered.**

If you don’t want any further traversing movements to be recorded, activate the ”Return” mode (see page 7–36). The NC will then set the output signal ”Remove finished” (channel interface; NC-O16.6). The axes can still be jogged in this mode.
Depending of the chosen return strategy (see Section 7.9.1, page 7–32 ff.), return to the contour can be started via ”Cycle start”.

When returning to the contour, the ”contact point” plays an important role because the distance between contact point and return point (point of re-entry to the contour) are used to calculate the required compensation values. The contact point is

- the current position in return mode ”Manual”

Remove and return to the contour

- the position at the end of the first recorded traversing movement in return mode ”Automatic” or ”Single block”.

If the contact point is chosen favorably, the distance contact point – return point can be determined so that the re-entry to the contour will be as ”smooth” as possible.

☞ **In the mode ”Remove from contour”, machining can be continued at any time at the set return point using ”Cycle start”, if the input signal ”Return to contour” (channel interface; NC-I3.1) has not been set simultaneously.**
In this case, the control unit traverses from the current position to the set return point, takes into account any required compensation values and continues the machining process.

Restrictions:

- When leaving the contour, the jog signals (Manual+ / Manual–) have a direct effect on the respective machine axes. A possibly active axis transformation or a rotation of the program coordinate system (PCS) related to the basis coordinate system (BCS) is not taken into consideration!
- Returning to the contour cannot be performed on several channels simultaneously.
- Returning to the contour is only possible at the end of a possibly active CPL waiting state (e.g. WAIT). If a CPL waiting state is active at the time of interruption, the control unit will output a warning.
- G32 (tapping) must not be active.
- After the interruption, running spindles must be stopped either manually (via switching function) or by the PLC and restarted in time before machining continues.

Relevant MACODA parameters (MP):

7060 00310: Return mode
Presetting of the return mode (manual, automatic or single block).
Can be suppressed by settings in the dialog ”Fix strategy” (“Manual” main menu, softkey sequence REMOVE FROM CONTOUR ▶ FIX STRATEGY) until the next control unit start-up.

7060 00320: Return point.
Presetting of the return point to the contour (start of block, interruption point or end of block).
Can be suppressed by settings in the dialog ”Fix strategy” until the next control unit start-up.

7060 00330: Maximum recording for return to contour in blocks.
Determines the maximum number of traversing movements which can be recorded after activating the mode ”Remove from contour”.

Remove and return to the contour

Relevant interface signals:

at the axis interface

NC-I0.0: ”Manual+”

NC-I0.1: ”Manual–”

at the channel interface

NC-I1.0: ”Cycle start”

NC-I2.0 to 2.3: ”Mode selection bit x”
Required operating mode: ”Return to path”.

NC-I3.0: ”Mode selection by PLC”

NC-I3.1: ”Return to the contour”

NC-I10.0 to 11.7: ”Feed hold”

NC-O15.0 to 15.3: ”Active mode bit x”
Feedback for NC-I2.0 to NC-I2.3.

NC-O16.4: ”Ready for re-entry”

NC-O16.5: ”Re-entry active”

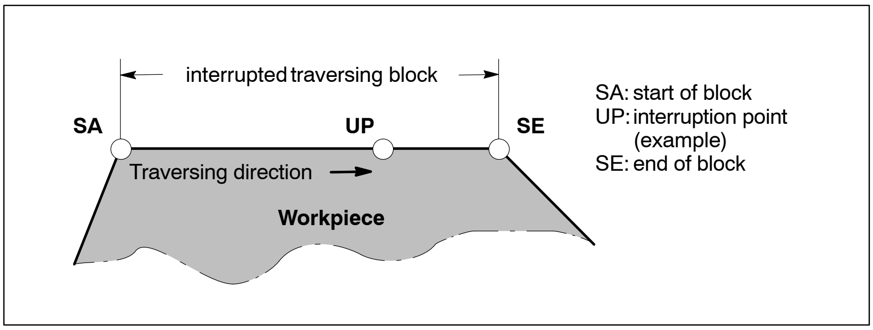
NC-O16.6: ”Remove finished”

7.9.1 Options for returning to contour

Selecting the return point

Available options:

- start of block
- interruption point
- end of block



CAUTION
Danger of damage to the machine or the workpiece!
When you use the option ”end of block”, keep in mind that the contour has not yet been machined from the interruption point on!

Remove and return to the contour

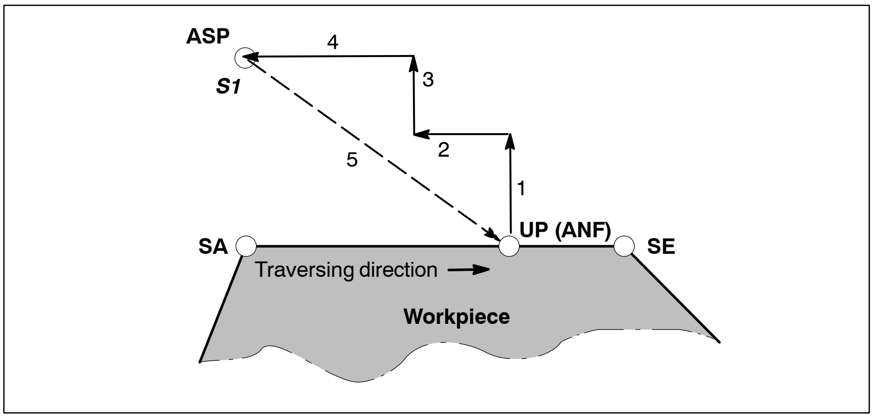
Selecting the return mode

Available options:

- manual
- automatic
- single block

☞ **For the figures in the example below, please note: selection of return point is set to ”interruption point”.**

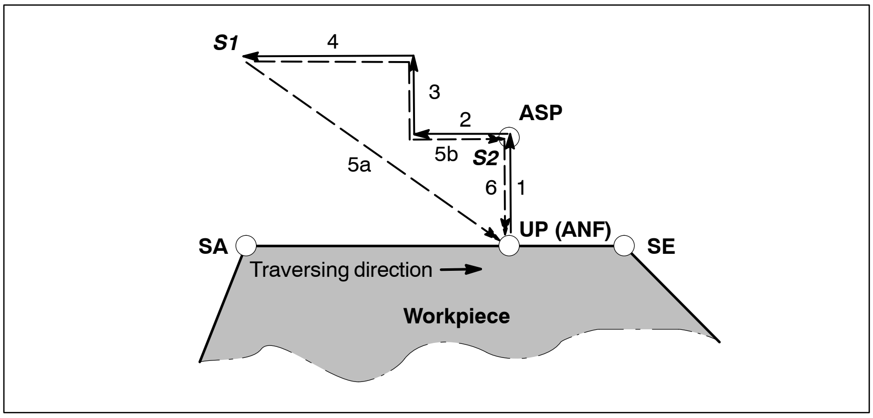
Manual:



1 ... 4:	Leaving the contour through jogging movements. The position at the end of the jogging movement is always the (new) contact point.
S1:	Initiating cycle start.
5:	Return movements to the return point (here: interruption point) and immediate continuation of machining.
ASP:	contact point (example)
SA:	start of block
UP:	interruption point (example)
ANF:	return point
SE:	end of block

Remove and return to the contour

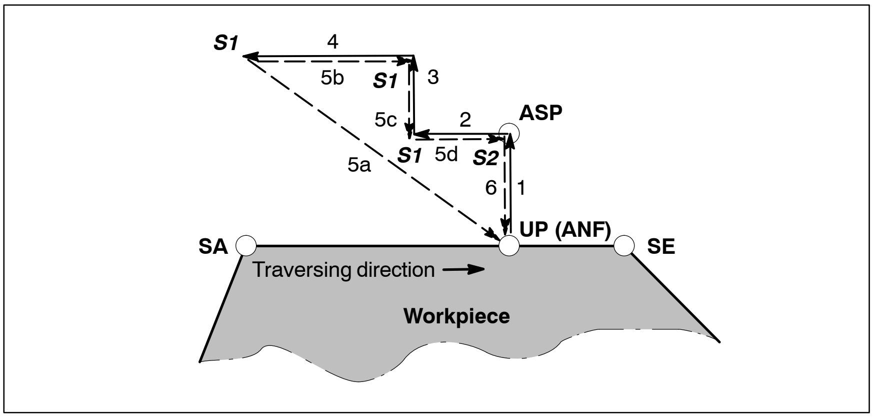
Automatic:



1 ... 4:	Leaving the contour through jogging movements.
S1, S2:	Initiating cycle start.
5a:	Return movement with immediate continuation of machining at the return point (here: interruption point) when the input signal ”Return to contour” has not been set when cycle start is initiated (*S1*).
5b:	Return movement to the contact point when the input signal ”Return to contour” has been set when cycle start is initiated (*S1*). The contact point can be changed subsequently via jogging movements. The NC then traverses to the return point (here: interruption point) via cycle start (*S2*) and continues the machining process.
ASP:	contact point (example)
SA:	start of block
UP:	interruption point (example)
ANF:	return point
SE:	end of block

Remove and return to the contour

Single block:



1 ... 4: Leaving the contour through jogging movements.

S1, S2: Initiating cycle start.

5a: Return movement with immediate continuation of machining at the return point (here: interruption point) when the input signal ”Return to contour” has not been set when cycle start is initiated (*S1*).

5b ... 5d: Separate return movements to the contact point when the input signal ”Return to contour” has been set when cycle start is initiated (*S1*).

☞ **If the input signal ”Return to contour” has not been set when cycle start is initiated (*S1*), the NC traverses directly to the return point and continues the machining process.**

The contact point (ASP) can still be changed via jogging movements. The NC then traverses to the return point (here: interruption point) via cycle start (*S2*) and continues the machining process.

ASP: contact point (example)

SA: start of block

UP: interruption point (example)

ANF: return point

SE: end of block

7.9.2 Assigning

1. Adapt MP 7060 00310, 7060 00320 and 7060 00330 in accordance with the desired standard return strategy.

CAUTION
Danger of damage to the machine or the workpiece!
Do not initiate a system control reset during machining.

---

2. Initiate a system control reset (DIAGNOSTICS ▶ RESET FUNCTIONS ▶ SYSTEM CONTROL RESET).

7.9.3 Activating

☞ **Channel control reset cancels an activated mode! Removing motions which have already been saved are deleted in the process.**

Activating the mode ”Remove from contour” via operator interface (MMI):

Preconditions:

- the active part program has been interrupted by ”Feed hold”
- the output signal ”Remove finished” (channel interface; NC-O16.6) has not been set
- the input signal ”Return to contour” has not been set (channel interface; NC-I3.1).
- the input signal ”Mode selection by PLC” has not been set (channel interface; NC-I3.0).

★ Actuate softkey REMOVE FROM CONTOUR in the “Manual” main menu.

From this time on, the control unit will ”remember” all traversing movements that you initiate using the axis-related input signals ”Manual+” or ”Manual–”. In case of axis standstill, the control unit outputs the channel interface signal NC-O16.4 ”Ready for re-entry”.

Activating the mode ”Remove from contour” via PLC:

Preconditions:

- the active part program has been interrupted by ”Feed hold”
- the output signal ”Remove finished” (channel interface; NC-O16.6) has not been set
- the input signal ”Return to contour” has not been set (channel interface; NC-I3.1).

- the input signal ”Mode selection by PLC” has been set (channel interface; NC-I3.0).

★ Activate the channel mode ”Return” via the NC input signals NC-I2.0 to 2.3.

From this time on, the control unit will ”remember” all traversing movements that you initiate using the axis-related input signals ”Manual+” or ”Manual–”. In case of axis standstill, the control unit outputs the channel interface signal NC-O16.4 ”Ready for re-entry”.

Activating the mode ”Return to path” via operator interface:

Preconditions:

- The mode ”Remove from contour via operator interface” (see above) has been activated.

★ Press softkey RETURN TO PATH.
The softkey is available both in the “Manual” main menu (REMOVE FROM CONTOUR ▶ RETURN TO PATH) and in the “Automatic” main menu.
The NC will then set the output signal ”Remove finished” (channel interface; NC-O16.6).
Additional traversing movements initiated by jogging are possible but are not recorded anymore.

Depending of the chosen return strategy (see Section 7.9.1, page 7–32 ff.), you can start return to path via ”Cycle start”.

Activating the mode ”Return to path” via PLC:

Preconditions:

- The mode ”Remove from contour via PLC” (see above) has been activated.

★ Set positive edge at the input ”Return to contour” (channel interface; NC-I3.1).
The NC will then set the output signal ”Remove finished” (channel interface; NC-O16.6).
Additional traversing movements initiated by jogging are possible but are not recorded anymore.

Depending of the chosen return strategy (see Section 7.9.1, page 7–32 ff.), you can start return to contour via ”Cycle start”.

Remove and return to the contour

Notes:

8 Tool compensations

The dimensions and the position of a tool can be described using:

- tool reference points
- tool coordinate system
- geometric dimensions (compensations) and tool type

8.1 Basics

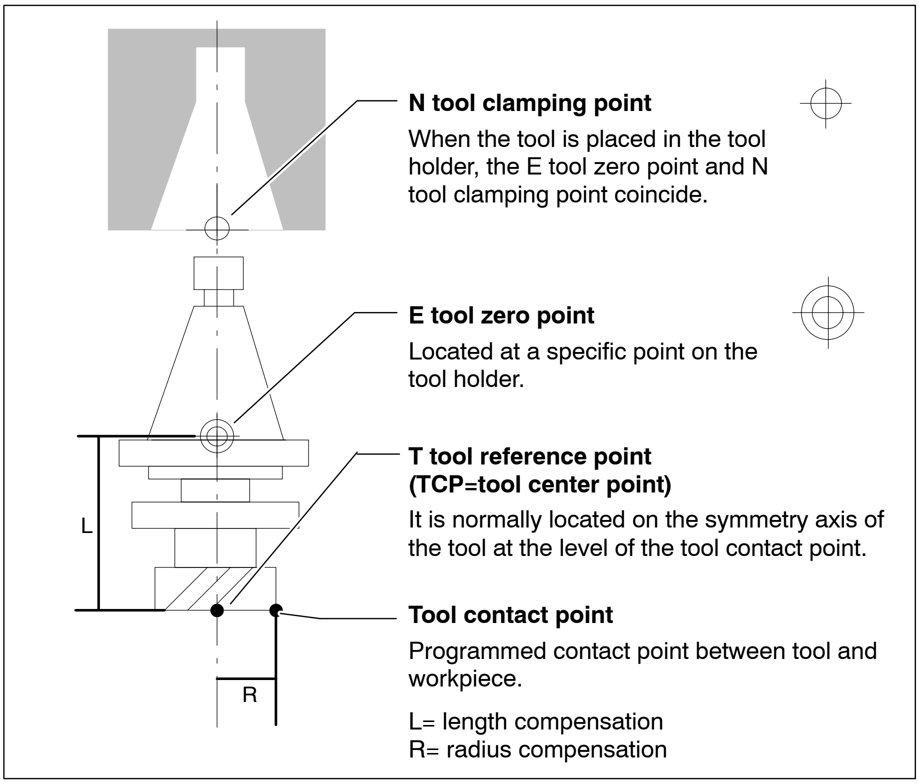
8.1.1 Tool reference points

A tool has different reference points:

The T (TCP tool center point) tool reference point is the coordinate origin of the TCS tool coordinate system. It is normally located on the symmetry axis of the tool at the level of the tool contact point.

To determine the position of the TCP tool reference point, the tool dimensions “R tool radius” and “L tool length” have to be known. These provide the position of the TCP in relation to the E tool zero point.

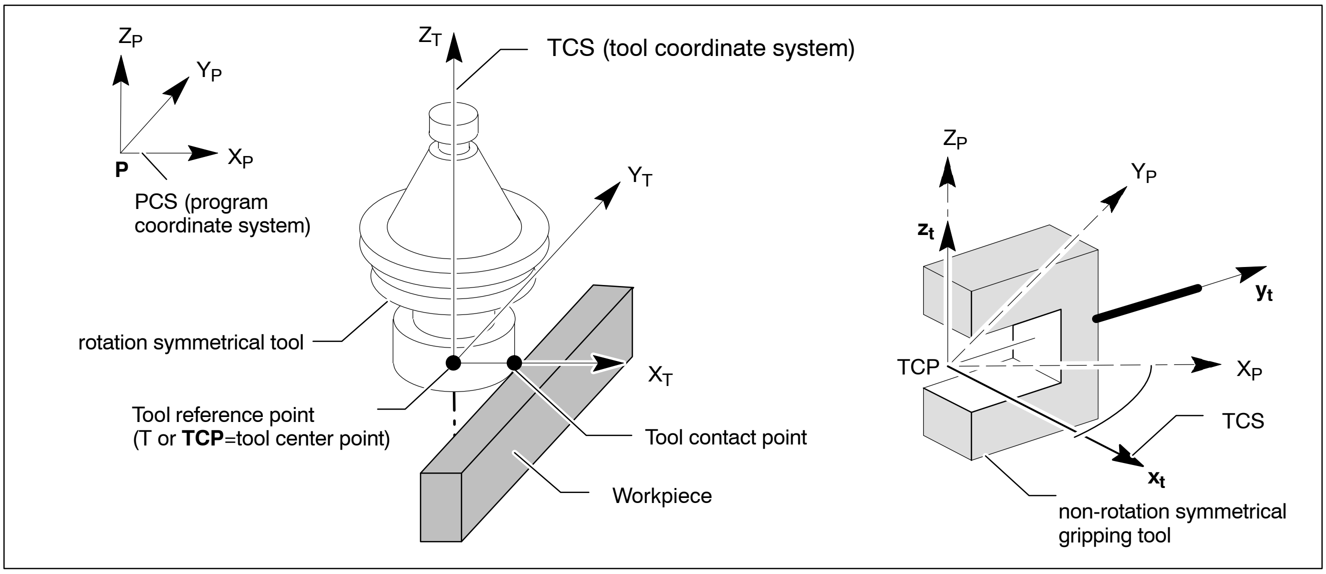
If the tool is located in the tool holder, the E tool zero point and N tool clamping point coincide.



Example: The dimensions of the tools in the case of a 3-axis machine, are related to a fixed tool setting point (e.g. tool clamping point, tool zero point).

8.1.2 Tool coordinate system

The TCS tool coordinate system differs from the range of coordinate systems because it is not defined when it takes the reference pose but is determined by the programmed movement (exception:. position definition in case of active axis transformation, see Section 8.1.3).
When the movement is programmed in the part program, the path of the tool in relation to the workpiece is specified in the PCS program coordinate system.
These path coordinates are the coordinates of the tool contact point. Taking into account the length and radius compensations of the active tool, the control unit calculates the actual path of the TCP which represents the zero point of the TCS.



If the TCS is parallel with the coordinates of the PCS and a rotation symmetrical tool is used, the position of the TCS in relation to a workpiece coordinate system does not play any role. In this case, working range coordinate programming is **not** active.

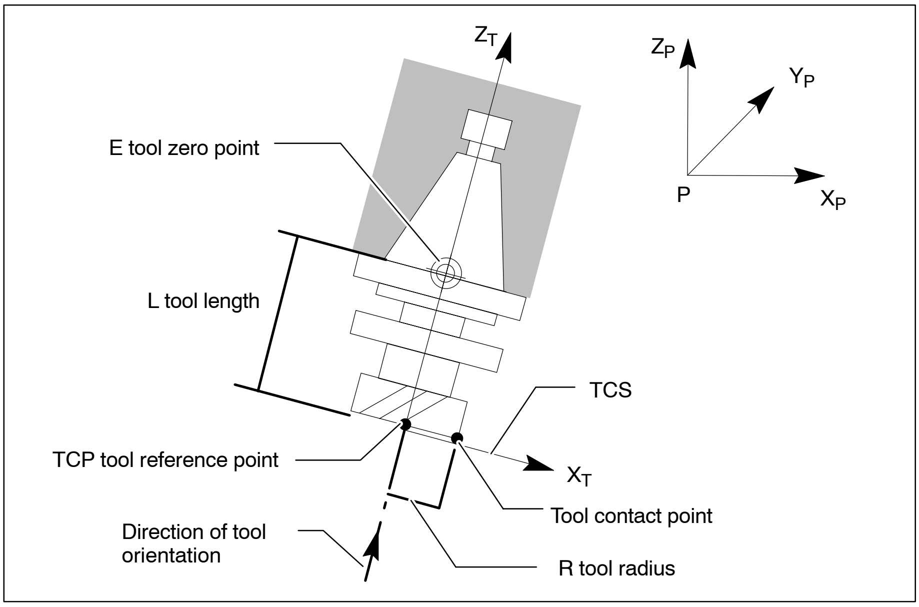
8.1.3 Tool coordinate system with active axis transformation

In case of **working range coordinate programming** with active **axis transformation** (TCP programming), the relative position of the tool contact point with regard to the PCS program coordinate system is programmed.

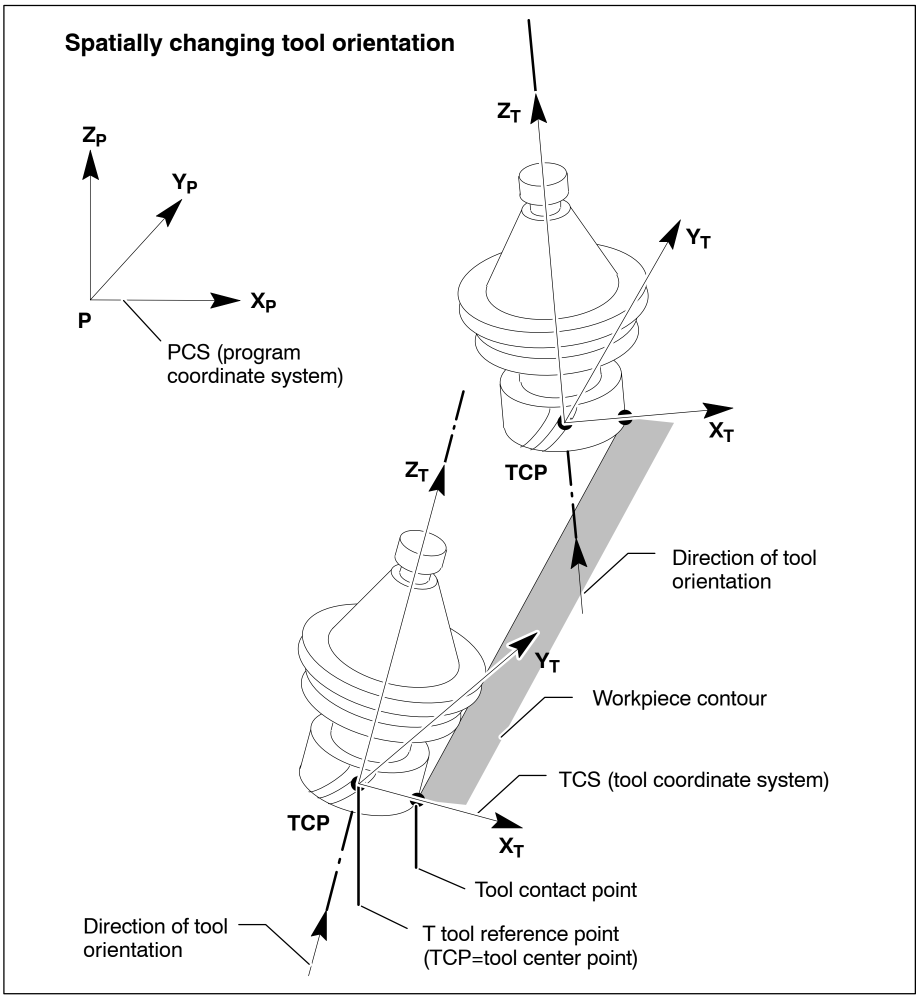
The position of the TCP is determined using the L and R tool compensations relative to the E tool zero point, the current tool orientation and the programmed coordinates.

In case of non-rotation symmetrical tools, the position of the TCS can be determined indirectly via an NC function (ROTAX(..), O(..)).

Tool compensations



In many instances, machining with spatially changing **tool orientation** is performed with **active axis transformation**. The tool compensation is then performed within the axis transformation along the axes of the TCS tool coordinate system.

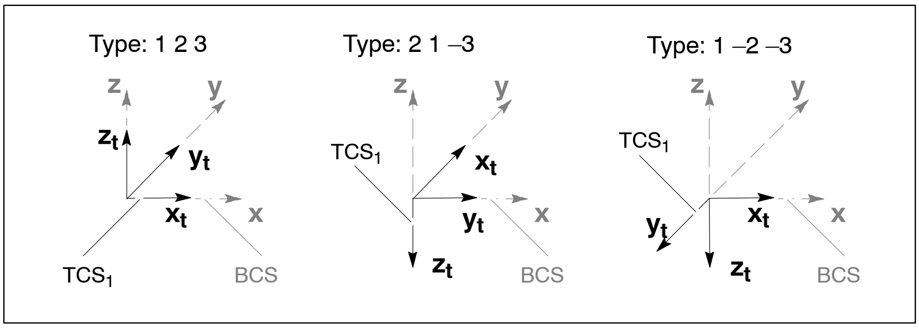


Tool compensations

Reference pose

In case of 5 axis and 6 axis transformation, the TCS tool coordinate system first assumes a **reference pose** which is aligned paraxial relative to the BCS and allows for 24 different orientations (see example below or refer to page 5–95 ff. for details)
The tool coordinate system in its reference pose is referred to as **TCS_1** (for all orientations). The TCS_1 can be defined with the axis transformation parameter MP 1030 00150 for the tool.

Example: Three out of 24 possible orientations of the TCS relative to the BCS in the reference pose of the axis transformation.



Relevant MACODA parameter (MP):
1030 00150: Reference orientation of the tool coordinate system

Orientation of the tool coordinate system

The orientation of the TCP and the TCS connected to it can be influenced by the axis transformation. The TCS_1 is shifted and rotated, if required, by special compensations and definitions.

The orientation of the TCS_1 can be changed as follows:

- by **shifting** the TCS_1 in a tool coordinate system TCSc with an explicit tool compensation (e.g. CPL command “TC”)
- by **shifting and optionally rotating** the TCS_1 in a tool coordinate system TCSc with an explicit tool compensation (e.g. CPL command “TC”)
- by **specifying working range coordinates** for the tool coordinate system **TCS_p** with the function ”TCS definition in program coordinates” (TCSDEF command) .

Tool compensations

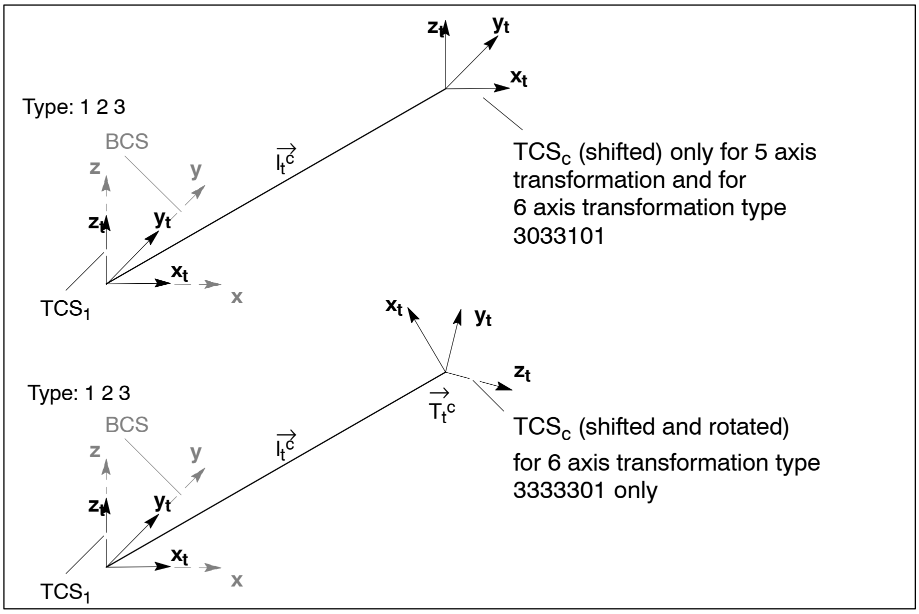
Explicit tool compensation

Precondition:

- Is only applicable in connection with an active 5 axis or 6 axis transformation.

Function:

A TCSc is created via an explicit tool compensation (e.g. CPL command "TC"), which can be shifted and rotated in relation to TCS_0 or TCS_1.



Restrictions:

- Shifting **and** rotating the tool coordinate system only works for the 6 axis transformation type 3333301.
- Shifting the tool coordinate system only works for the 5 axis transformations and for 6 axis transformation type 3033101.

Relevant CPL functions:

TC(..) write (absolute/additive) and read access to compensation values stored within the CNC (L, R, Eulerian angles ϑ, φ, ψ) e.g.
- in ASCII geometry tables with freely definable names
- external geometry compensation values
- in data base tables K4 and K5
(for details, please refer to the CPL Programming Manual).

Tool compensations

TCS definition in program coordinates

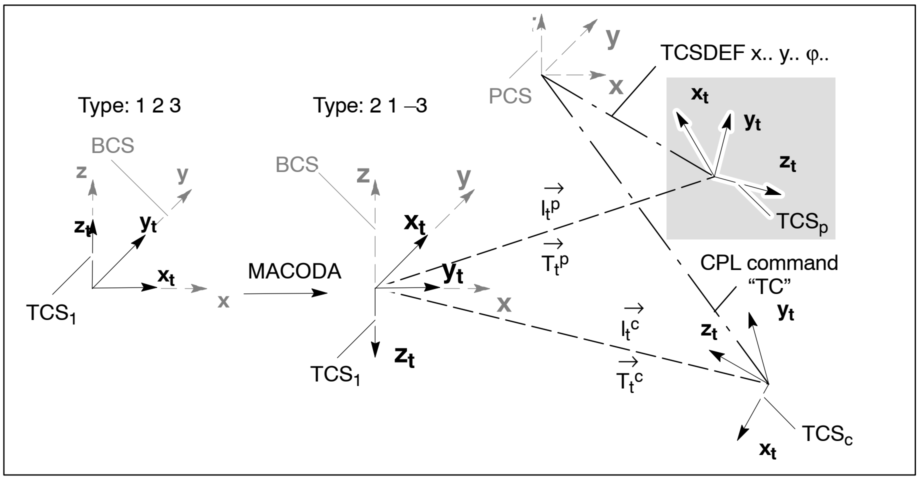
Precondition:

- It is only applicable in connection with an active 6 axis transformation.

Function:

The function ”TCS definition in program coordinates” creates a tool coordinate system TCS_p, which supersedes the current TCS_c or TCS_1 . The coordinate values specified in the TCSDEF command for the TCS_p, are converted to the variables $\vec{l}_t^p$ and $\vec{T}_t^p$ within the NC and stored in the tool compensation memory. Switching off with TCSUNDEF reactivates the TCS_c, or, if no explicit tool compensation is active, the TCS_1.

The figure below shows the active TCS_p and the remaining inactive tool coordinate systems:



Restrictions:

- The command TCSDEF can only be used if an axis transformation with orientation identification 3 (tensor orientation) is active.

Relevant G functions and addresses:

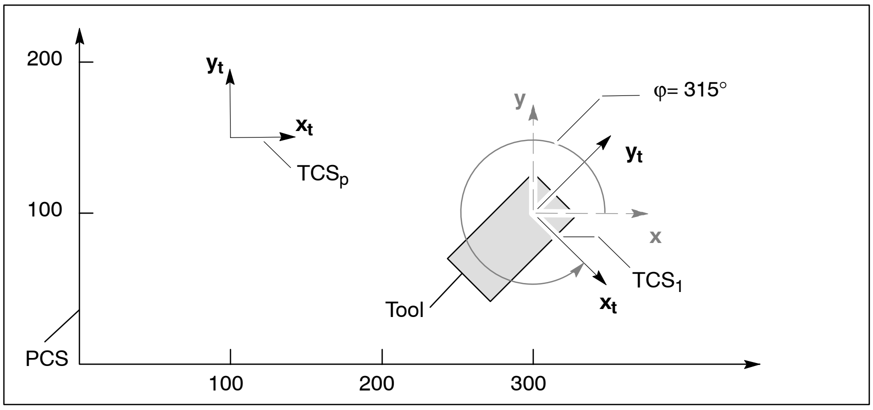
TCSDEF[<*Linear coordinates*>] [<*Orientation coordinates*>]
Defining orientation of the tool coordinate system **TCS_p**

Linear coordinates:	coordinates related to the current PCS
Orientation coordinates:	coordinates related to the current PCS or all alternative syntaxes of the tensor orientation (see page 5–71 ff.)
TCSUNDEF:	reset to previously active tool coordinate system (e.g.: TCS_0 or TCS_1)
	An automatic reset is performed for every coordinate switching ”N.. COORD(n)”.

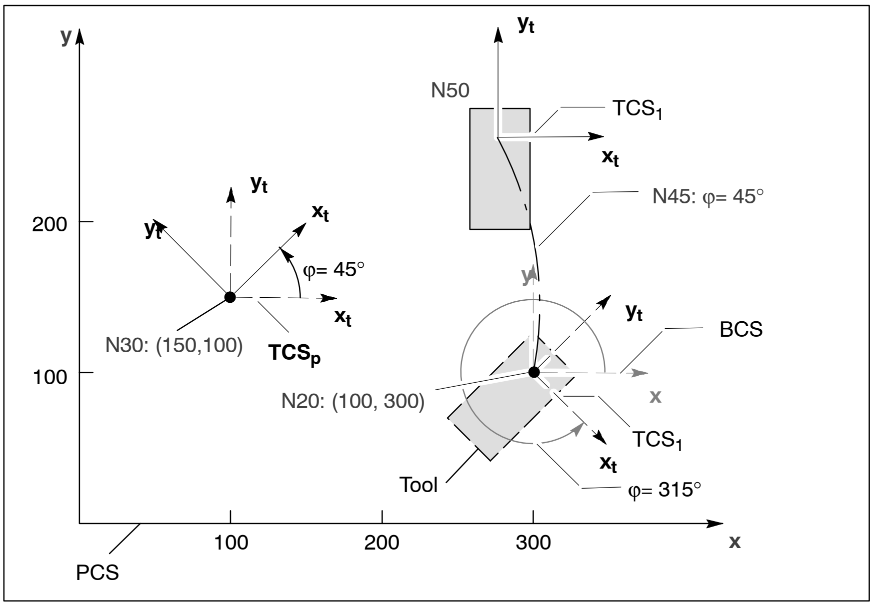
Example:

`N10 COORD(1)`	Axis transformation on
`N20 x300 y100 phi315`	Movement of the TCS_1 to the initial position
`N30 TCSDEF x100 y150 phi0`	Movement of the TCS_1 to the initial positionool compensation to
`N40 phi45`	rotation of the TCS_p
`N45 ...`	Position and orientation data related to
`...`	the TCS_p
`N50 TCSUNDEF`	Programming related to the TCS_1 again

In the following figure, the TCS_1 permanently connected to the tool is located at position x=300, y=100 and has the orientation φ=315°. By programming **TCSDEF x100 y150 phi0** the TCS_p is defined:



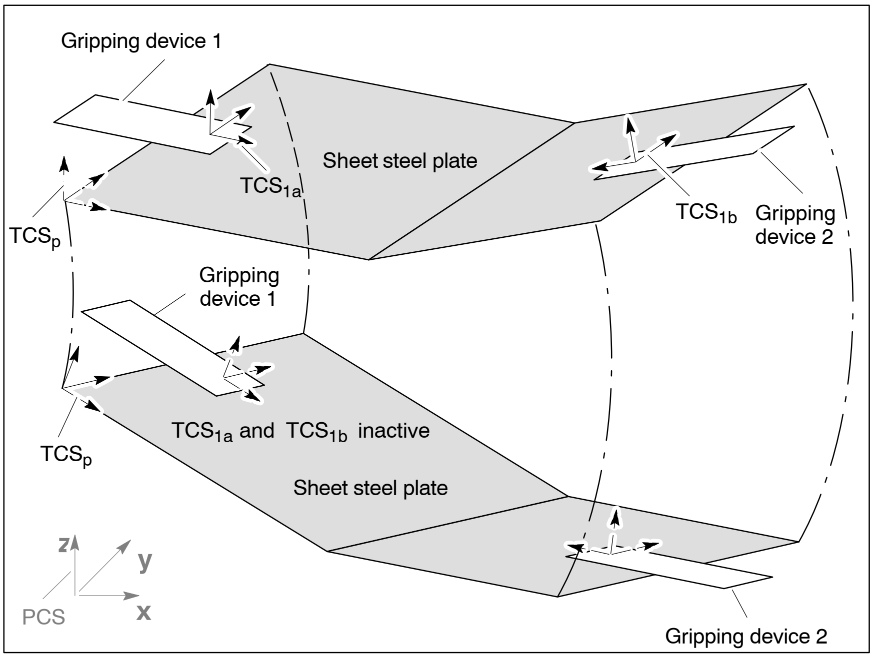
The position and orientation data following TCSDEF in the NC program are related to the movement of the TCS_p (NC program section: N45 phi45):



Application example: sheet steel machining system

Two coupled robots (see also channel-independent coordinate coupling) hold a sheet steel plate at two positions and move it in space.

The tool coordinate systems TCS_{1a} and TCS_{1b} of the two gripping devices are compensated to a joint TCS_p of the sheet steel plate. If the sheet steel plate is to be moved in space, you only have to program the position and orientation of the joint TCS_p.



8.1.4 Assigning

Explicit tool compensation

1. Enter all settings for your application as described in Section 5.7.4.
2. Please note the selection of the transformation type: The TCS can only be shifted and rotated for certain types (see restrictions on page 8–5).

TCS definition in program coordinates

1. Enter all settings for your application as described in Section 5.7.4.
2. Please note the selection of the transformation type: The TCS definition can only be used for types with tensor orientation (see restrictions on page 8–6).

8.1.5 Activating

Explicit tool compensation

1. Enter all settings for your application as described in Section 5.7.5.
2. The function is activated e.g. with the CPL-TC function.

TCS definition in program coordinates

1. Enter all settings for your application as described in Section 5.7.5.
2. The function is activated with TCSDEF .....

8.1.6 Deactivating

Explicit tool compensation

1. Enter all settings for your application as described in Section 5.7.6.
2. The function is reset for every coordinate switching COORD(n).

TCS definition in program coordinates

1. Enter all settings for your application as described in Section 5.7.6.
2. The function is deactivated with:
   - TCSUNDEF or
   - coordinate switching COORD(n).

8.2 Geometry compensations

8.2.1 Overview

Taking into account the current tool data (e.g. length, radius, cutting edge orientation), the PNC can automatically calculate the actually required

- paths and
- path depths (e.g. for drilling)

for a programmed workpiece contour in dependence on the tool used. These tool data are referred to in the PNC as “geometry compensations of the tool”.

In this manner, it is possible to use construction drawing dimensions directly as a basis for part program preparation. Your part program becomes ”independent on the tool” with regard to the dimensions.

In order to use geometry compensations, the control unit has to know the relevant tool data.

Depending on the geometry compensation type used, the tool data are provided by a geometry compensation table (GEO table) or by the PLC.

CAUTION
Input of wrong tool dimensions results in incorrect contour!
As the tools are subject to wear in the course of machining, their length or radius will change, depending on the tool type. This effect also occurs in case of sharpening.
Therefore verify that all tool data always correspond to the actual dimensions of the tools employed when you are working with geometry compensations!
Take new measurements of the tools used at regular intervals (depending on e.g. tool life, tool contact time) and update the tool data provided.

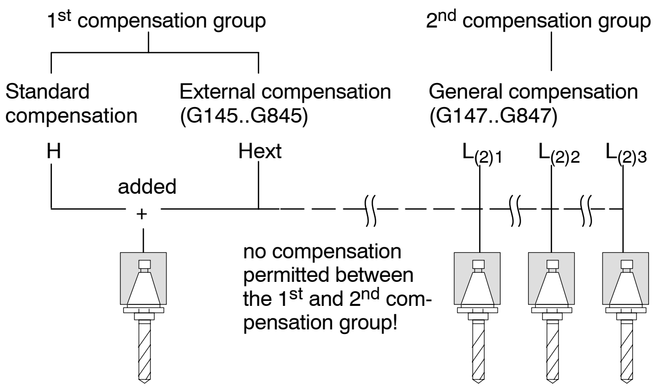
The relationship between geometry compensation type, the used (tool) parameters and how they are made available is shown in the table below:

Geometry compensation type	Compensation group	used tool parameters (set)	Made available
Standard compensation	1	D=radius H=length	via GEO table
External compensation		R_{ext}=radius L_{ext}=length	via PLC
General compensation	2	R=radius $L_{(2)1}$= length (log. axis 1) $L_{(2)2}$= length (log. axis 2) $L_{(2)3}$= length (log. axis 3) TO=cutting edge orientation	via PLC

Geometry compensations

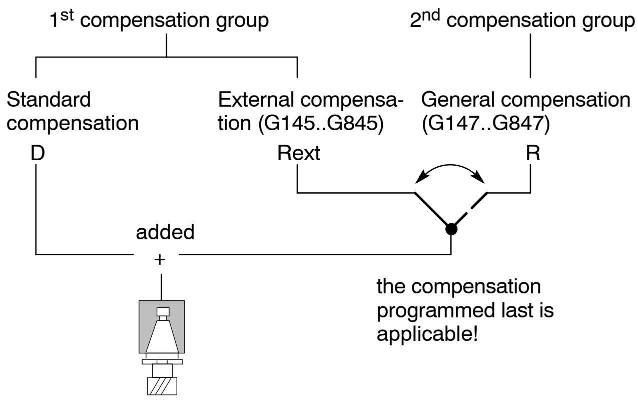
Compensation values of simultaneously activated geometry compensation types may act additively or overwrite each other, depending on what compensation group they pertain to:

Dependencies of the length compensations



Max. 4 length compensations active simultaneously: the length compensation of the 1st compensation group as well as the three length compensations of the 2nd compensation group have an effect on the respective different axes (determination via MACODA or G78).

Dependencies of the radius compensations



The radius compensations of the 1st and 2nd compensation group are added and/or replace each other partially.

For details on the individual geometry compensation types, see the following pages:

- Standard compensation (1st compensation group): page 8–13 ff.
- External compensation (1st compensation group): page 8–14 ff.
- General compensation (1st compensation group): page 8–17 ff.

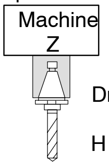
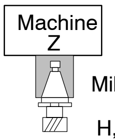
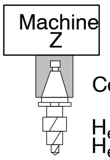
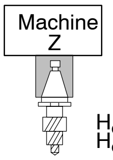
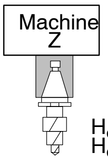
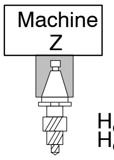
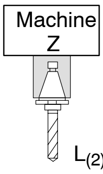
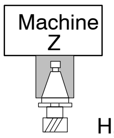
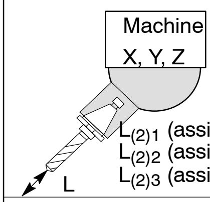
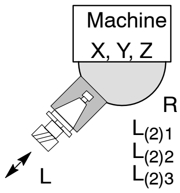
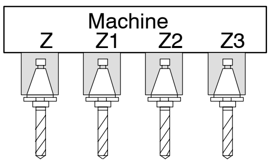
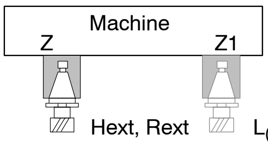
For details on the structure of geometry compensation tables, see page 8–30 ff.

If tool data are used in connection with activated geometry compensation types, the following functions may be used:

- **G41/G42 path compensation**.
  Determines whether positive radius compensation values are to act to the left (G41) or to the right (G42) of the contour (in relation to the traversing direction).
  In case of activated path compensation, the control unit causes the tool to move along an equidistant path at the distance of the active radius compensation value to the programmed path.
  See description on page 8–37 ff.
- **G78 drill axis switching.**
  Using the “Drill axis switching” function (see page 8–32 ff.), it is possible to switch the length compensation to any linear machining axis.

Geometry compensations

The following examples show how the geometry compensations from the 1st and 2nd compensation group can be applied to different tools.

Drilling (length compensation)	**Milling (radius and length compensation)**
1st compensation group: H  Machine Z Drilling tool H	1st compensation group: D,H  Machine Z Milling tool H, D
1st compensation group: G145..G845 (H_{ext})  Machine Z Combined tool H_{ext1} H_{ext2} ...	1st compensation group: G145..G845 (H_{ext} , R_{ext})  Machine Z H_{ext1} R_{ext1} H_{ext2} R_{ext2} ...
1st Compensation group: H + G145..G845 (H_{ext})  Machine Z H_{ext1}+ H H_{ext2}+ H ...	1st compensation group: H, D + G145..G845 (H_{ext} , R_{ext})  Machine Z H_{ext1}+ H, R_{ext1}+ D H_{ext2}+ H, R_{ext2}+ D ...
2nd Compensation group: G147..G847 ($L_{(2)3}$)  Machine Z $L_{(2)3}$	2nd compensation group: G147..G847 ($L_{(2)3}$, R)  Machine Z H, D
2nd Compensation group: G147..G847 ($L_{(2)1}$, $L_{(2)2}$, $L_{(2)3}$)  Machine X, Y, Z Anglehead tool $L_{(2)1}$ (assigned to the X axis) $L_{(2)2}$ (assigned to the Y axis) $L_{(2)3}$ (assigned to the Z axis) L=resulting length compensation L	2nd Compensation group: G147..G847 ($L_{(2)1}$, $L_{(2)2}$, $L_{(2)3}$, R)  Machine X, Y, Z R $L_{(2)1}$ (assigned to the X axis) $L_{(2)2}$ (assigned to the Y axis) $L_{(2)3}$ (assigned to the Z axis) L=resulting length compensation L
1st+2nd Compens. group: H, G147..G847 ($L_{(2)1}$, $L_{(2)2}$, $L_{(2)3}$)  Machine Z Z1 Z2 Z3 H $L_{(2)1}$ $L_{(2)2}$ $L_{(2)3}$ H must **not** be assigned to the Z1, Z2 and Z3 axis	1st+2nd compensation group: G145..G845 (Hext, Rext) **or** G147..G847 ($L_{(2)3}$, R)  Machine Z Z1 Hext, Rext $L_{(2)3}$, R

Geometry compensations

8.2.2 Standard compensation

Precondition:

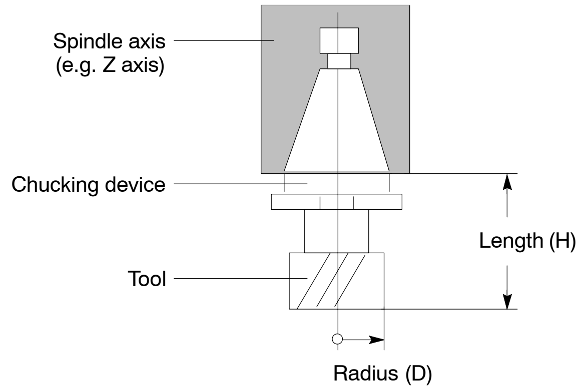
- Compensation group 1 has to be released in MP 7050 00410.

Function:

The standard compensation takes into account the tool dimensions ”length” and ”radius”. For this purpose, a maximum of 48 compensation value pairs can be stored in each geometry compensation table.

☞ **For details on naming conventions, activation and structure of geometry compensation tables, see page 8–30 ff.**

Following a tool change, the relevant compensation data can be activated from the geometry compensation table using address D (radius) and/or H (length). To do this, the correct geometry compensation table must first have been selected using the G22 block.



☞ **Depending on the measuring method, the length compensation may refer to the “spindle nose” or a “zero tool”!**

Restrictions:

- The following have an **additive** effect on the standard compensation:
  - the length compensation value of a possibly active “external compensation” (G145.. G845)
  - the radius compensation value of a possibly active ”external compensation” **or** the radius compensation value of a possibly active “general compensation” (G147.. G847).
- If with regard to the path depth or drilling direction the tool is obstructed by a machining axis which is not arranged paraxial to the working spindle, a drill axis switch has to be performed before the machining phase so that the length compensation value is taken into account correctly (G78, see page 8–32).

Relevant G functions and addresses:

G22 K <*Path/table name*>:
Activating a geometry compensation table via NC block.

H1... H48: Activating the length compensation value (1–48).

H0: Switching off the length compensation value 1 through 48.

D1... D48: Activating the radius compensation value (1–48).

D0: Switching off the radius compensation value 1 through 48.

Relevant MACODA parameters (MP):

7050 00410: Channel-specific release of
compensation group 1 (for H compensation value within the standard compensation and for external compensation)
and/or
compensation group 2 (general compensation).

7050 00420: Channel-specific determination which logical axes are assigned to the individual length compensations/compensation groups.

9020 00010: Specifies the unit of measure used to save the geometry table values within the CNC.

Relevant CPL functions:

TC(..) write (absolute/additive) and read access to compensation values stored within the CNC.

MCODS(...) see ”Compensations” under MCDOS in the CPL Programming Manual

8.2.3 External compensation (1st external compensation)

Precondition:

- Compensation group 1 has to be released in MP 7050 00410.

Function:

The ”external compensation” (also referred to as ”1st external compensation”) takes into account the tool dimensions ”length” and ”radius”, as the standard compensation does.
In contrast to the standard compensation, however, the compensation values are not provided via the geometry compensation table but transferred by the PLC automatically to the CNC (required program module –B22WZKOR) in dependence on the tool number (T word) after a tool is exchanged (via M function).

Geometry compensations

A maximum of 8 compensation value pairs (H_{ext}: length compensation; R_{ext}: radius compensation) are possible for each tool.

The following compensation values are used in dependence on the tool type (MP 7050 00430):

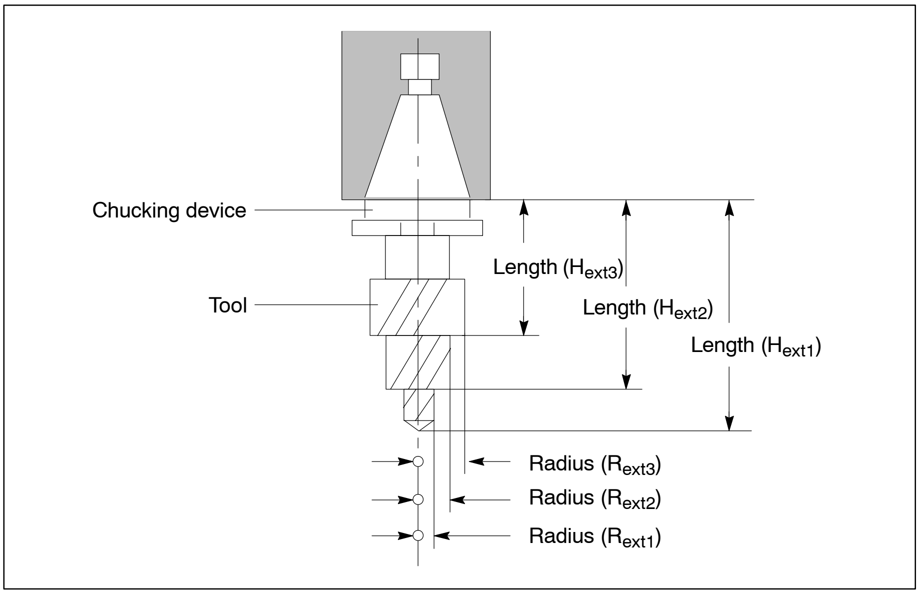
- drilling tool: $L_{(1)3}$ (H_{ext})
- milling tool: $L_{(1)3}$ (H_{ext}), R (R_{ext})

The transfer having been performed, it is possible to activate the relevant compensation pair using G145 through G845.

☞ **The CNC internally saves all compensation values. Read and write access to these values is thus possible using CPL (TC command).**

Owing to the 8 possible compensation value pairs, the "external compensation" is well-suited for use with combination tools or tools with exchangeable head because several length and radius compensation values are required for each tool used.

Example: tool with 3 compensation value pairs (combination tool)



Restrictions:

- The following have an **additive** effect on the external compensation:
  - the length compensation value of a possibly active "standard compensation" (H)
  - the radius compensation value of a possibly active "standard compensation" (D).
- The radius compensation value of the "external compensation" is overwritten by the radius compensation value of a "general compensation" activated later (G147.. G847).

- If with regard to the path depth or drilling direction the tool is obstructed by a machining axis which is not arranged paraxial to the working spindle, a drill axis switch has to be performed before the machining phase so that the length compensation value is taken into account correctly (G78, see page 8–32).
- If the PLC changes the currently active external compensation values in the course of the execution of a part program, this change will only become active in the block **being under preparation as the next block**.
  Under certain circumstances this can mean that even more blocks are due to execution without this change.
  In order to avoid this effect you must program the ”WAIT” CPL command directly after the block, causing the PLC to hand over the new compensation values. By doing so you hold the block preparation of the PNC until all program blocks ahead of ”WAIT” have been executed.
- Following the control unit’s start-up, the compensation values saved internally are set to zero. Internal values may only be changed by PLC or CPL.
- In case of control reset, the compensation values saved within the NC are maintained.

Relevant MACODA parameters (MP):

7050 00410:	Channel-specific release of compensation group 1 (for H compensation value within the standard compensation and for external compensation) and/or compensation group 2 (general compensation).
7050 00420:	Channel-specific determination which logical axes are assigned to the individual length compensations/compensation groups.
7050 00430:	Channel-specific determination for which type of tool the compensation groups 1 and 2 are to be used: Drilling tool, milling tool, lathe tool, anglehead tool. In this manner, you determine how many compensation values the NC can expect from the PLC in case of ”external” or ”general compensation”.
9020 00010:	Specifies how the control unit interprets the compensation values received from the PLC.

Relevant G functions:

G145..G845:	Activating the corresponding compensation value pair of the “external compensation”.
G146:	Deactivating the compensation value pair of the “external compensation”.

Relevant interface signals:

NC-O19.0 through NC-O19.3:
"External tool compensation 2^0 ... 2^3" (channel interface).
The signals show which compensation pair (G145–G845) of the "external compensation" is active on the channel.

Relevant CPL functions:

TC(..)	write (absolute/additive) and read access to compensation values stored within the CNC.
MCODS(...)	see "Compensations" under MCDOS in the CPL Programming Manual
WAIT	Stops block processing.

8.2.4 General compensation (2nd external compensation)

Precondition:

- Compensation group 2 has to be released in MP 7050 00410.

Function:

The "general compensation" (also referred to as "2nd external compensation") corresponds to the "external compensation" with regard to the functions, except for some extensions.

In this context too, compensation values are transferred by the PLC automatically to the CNC (required program module –B22WZKOR) in dependence on the tool number (T word) after a tool is exchanged (via M function).

A maximum of 8 **compensations sets** are possible for each tool. In contrast to the "external compensation", a compensation set consists of

- $L_{(2)1}$, $L_{(2)2}$, $L_{(2)3}$: length compensation or shift values
- R: Radius compensation
- TO: cutting edge orientation
  (Description of cutting edge orientation, see 8–21 ff.)

The following compensation values are used in dependence on the tool type:

- Drilling tool ($L_{(1)3}$)
- Milling tool ($L_{(1)3}$,R)
- Lathing tool ($L_{(2)1}$,$L_{(2)3}$,R,TO)
- Anglehead tool ($L_{(2)1}$, $L_{(2)2}$, $L_{(2)3}$, R, TO)

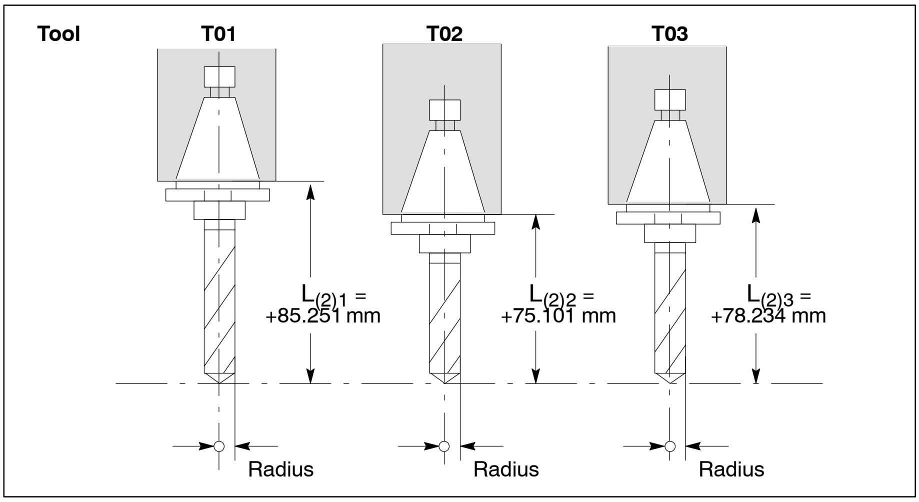
The transfer having been performed, it is possible to activate the relevant compensation set using G147 through G847.

☞ **The CNC internally saves the compensation sets. Read and write access to these values is thus possible using CPL (TC command).**

Geometry compensations

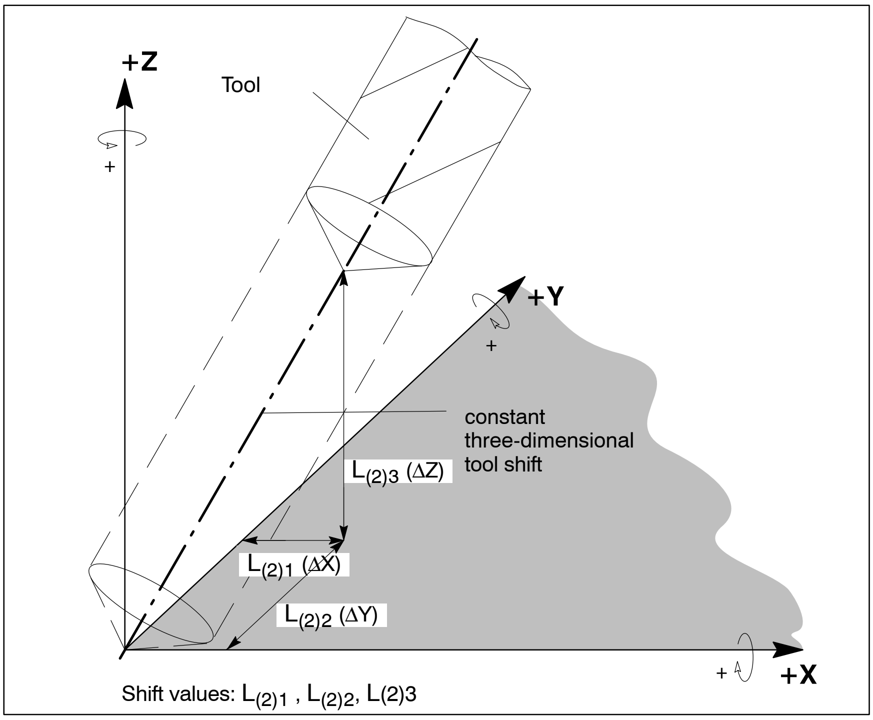
Owing to the 8 compensation sets and the extended compensation value number (in contrast to the ”external compensation” extended by 2 additional length compensation values and cutting edge orientation per set), the ”general compensation” is well-suited for use, e.g. for the simultaneous machining of several similar tools (e.g. several spindles located on different Z axes), in connection with lathing applications or anglehead tools.

Example 1:
Length compensations for the simultaneous machining of a workpiece using a maximum of 3 similar tools:



Example 2 :
Constant three-dimensional tool shift:

The resulting “three-dimensional tool shift” in the direction of the tool axis is calculated on the basis of the shift values of the axes involved.



Restrictions:

- The radius compensation value of an active “standard compensation” (D) acts additively with the radius compensation value of an active “general compensation” (G147..G847).
- The radius compensation value of the ”general compensation” is overwritten by the radius compensation value of an “external compensation” activated later (G145.. G845).
- The default assignment of the length compensation values $L_{(2)1}$ to $L_{(2)3}$ to the individual logical axes of a channel is preset channel-specific in MP 7050 00420.
  In this context, the individual compensations have to be assigned to different axes. In addition, there may be no overlapping with the drill axis of the 1st compensation group during axis assignment. As a consequence, the compensations of the 1st and 2nd compensation group cannot act additively.
- If with regard to the path depth or drilling direction the tool is obstructed by a machining axis which is not arranged paraxial to the working spindle, a drill axis switch has to be performed before the machining phase so that the length compensation value is taken into account correctly (G78, see page 8–32).
- If the PLC changes the currently active compensation values in the course of the execution of a part program, this change will only become active in the block **being under preparation as the next block**.
  Under certain circumstances this can mean that even more blocks are due to execution without this change.
  In order to avoid this effect you must program the ”WAIT” CPL command directly after the block, causing the PLC to hand over the new compensation values. By doing so you hold the block preparation of the PNC until all program blocks ahead of ”WAIT” have been executed.
- Following the control unit’s start-up, the compensation values saved internally are set to zero. The internal values may only be changed by the PLC or CPL.
- In case of control reset, the compensation values saved within the NC are maintained.
- The “general compensation” can be used in connection with the PLC program module –B22WZKOR (external tool compensation) only from V4.4.1 on.

Relevant MACODA parameters (MP):

7050 00410: Channel-specific release of
compensation group 1 (for H compensation value within the standard compensation and for external compensation)
and/or
compensation group 2 (general compensation).

7050 00420: Channel-specific determination which logical axes are assigned to the individual length compensations/compensation groups.

7050 00430: Channel-specific determination for which type of tool the compensation groups 1 and 2 are to be used:
Drilling tool, milling tool, lathe tool, anglehead tool.
In this manner, you determine how many compensation values the NC can expect from the PLC in case of ”external” or ”general compensation”.

9020 00010: Specifies how the control unit interprets the compensation values received from the PLC.

Relevant G functions:

G147..G847: Activating the corresponding compensation set of the “general compensation”.

G148: Deactivating the compensation set of the “general compensation”.

Relevant interface signals:

NC-O23.0 through NC-O23.3:
“General tool compensation 2^0 ... 2^3“ (channel interface).
The signals show which compensation set (G147–G847) of the ”general compensation” is active on the channel.

Relevant CPL functions:

TC(..) write (absolute/additive) and read access to compensation values of the ”general compensation” stored within the CNC.

MCODS(...) see ”Compensations” under MCDOS in the CPL Programming Manual

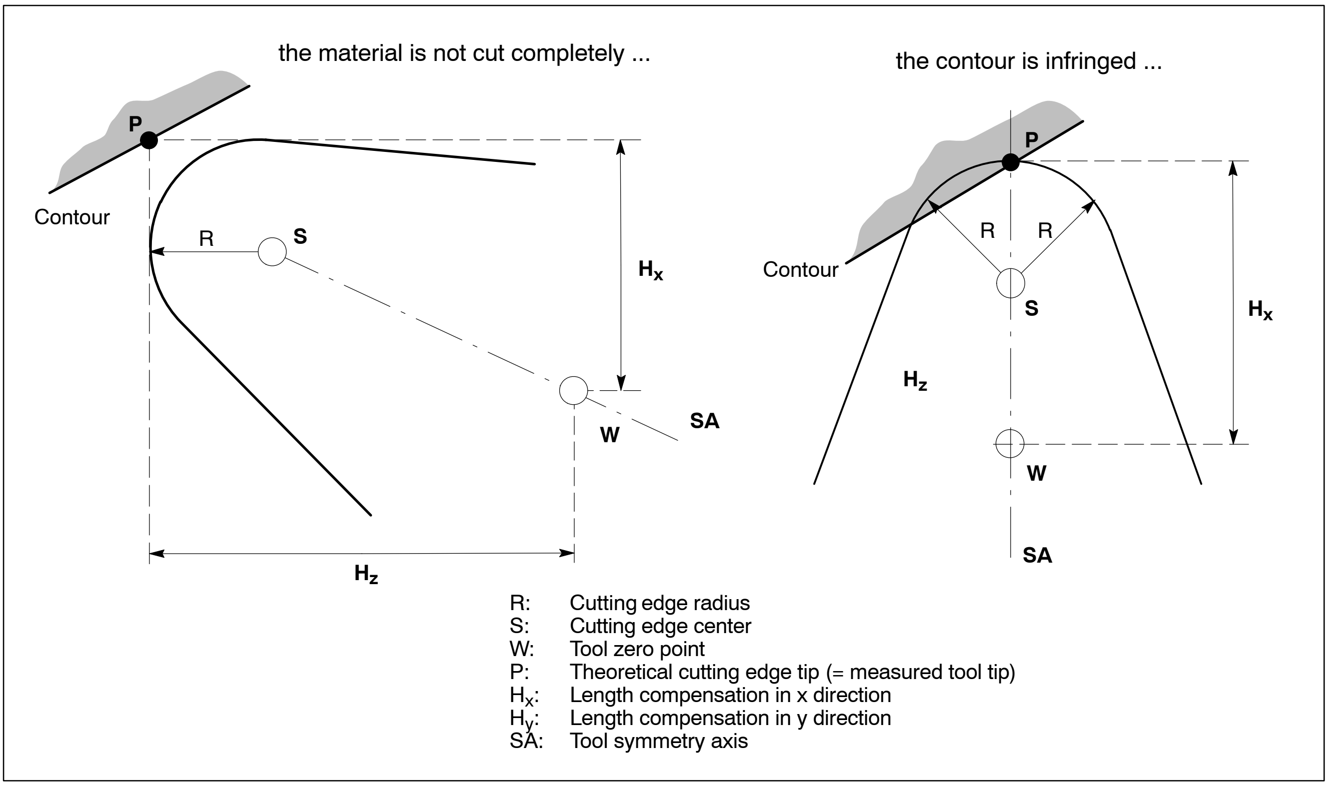
WAIT Stops block processing.

Geometry compensations

Description of the cutting edge (tool) orientation (TO):

The objective is to move the tool contact point along the programmed contour. In this context, please note that the tool contact point may shift along the cutting edge radius in dependence on the programmed contour.

If length compensations are used for the main and second axis exclusively, this will result in a fixed theoretical cutting edge tip P.
Since the control unit leads P along the programmed contour, the machining of freely defined contour paths will also lead to situation where the material is either cut insufficiently or too much:



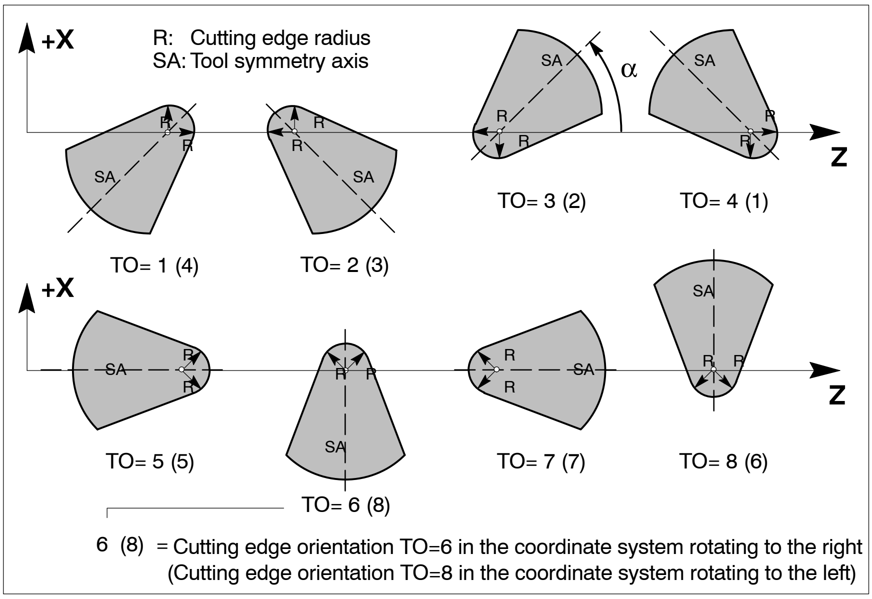
Two types of compensation acting in different ways are used together to compensate the problem shown in the figure above.

- Cutting edge orientation compensation and
- Tool radius compensation (G41, G42).

The cutting edge orientation compensation has the effect that the S tool center point is now located on the programmed contour instead of P. The tool radius compensation then guides the S cutting edge center point to the programmed contour on an equidistant at distance R.

The cutting edge orientation (TO) and the cutting edge radius (R) are needed for the **cutting edge compensation**.
The cutting edge orientation TO=1 to TO=8 describes the basic orientation of the tool, related to the main and second axis (see figure below).
The cutting edge orientation compensation is deactivated with TO=9.

Geometry compensations

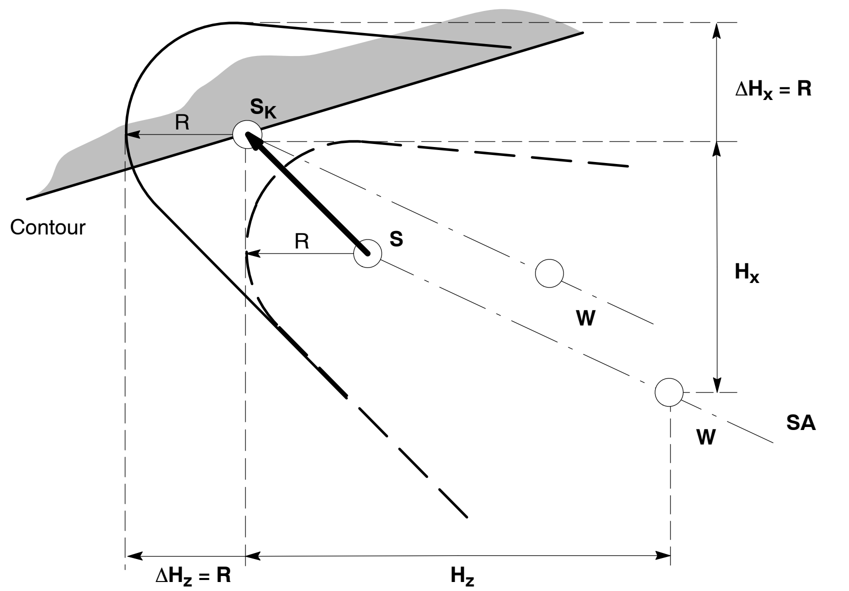


Using the cutting edge radius, the NC calculates an additional shifting of the shape, taking into account the specified cutting edge orientation, which has an additive or subtractive effect on the normal length compensation.

- $(\pm)\Delta H_X = R$ and $(\pm)\Delta H_Z = R$ (for cutting edge orientations 1 to 4)
- $(\pm)\Delta H_X = R$ (for cutting edge orientations 6 or 8)
- $(\pm)\Delta H_Z = R$ (for cutting edge orientations 5 or 7)

Geometry compensations

The principle is shown in the figure below:



R: Cutting edge radius
S: Cutting edge center without shift
SK: compensated position of the cutting edge center point
W: Tool zero point
H_x: Length compensation in x direction
H_y: Length compensation in y direction
ΔH_x: additional length compensation in x direction
ΔH_y: additional length compensation in y direction
SA: Tool symmetry axis

The **tool radius compensation** then shifts the path by the value of the cutting edge radius so that the cutting edge is positioned precisely on the programmed contour.

Geometry compensations

8.2.5 Overview: Methods of tool compensation

Compensation method	**Save under**	**Activate via**	**Programming** (example)	**Notes**	**Effect**
Standard length compensation Radius compensation	Length: H1 – H48 Radius: D1 – D48	G22 K{<Path>} <File name> and H.. D.. (G41, G42)	G22 K Geo1 .. .. G1 Z.. H.. G41 X.. Y..D..	Length and radius compensation values are activated using H or D addresses in the K compensation table <file name>, which is **activated with G22**. For deactivation, use H0, D0.	The effect of a radius compensation always comes to bear on the working plane selected in the program coordinate system.
External tool compensation (1st external tool compensation)	Length: $H_{ext.}$ Radius: $R_{ext.}$	G145 – G845	G145	Up to 8 external length and radius paired values can be activated by PLC. For deactivation use G146.	Additive on • standard length compensation and radius compensation
General tool compensation (2nd external tool compensation)	Length: L(2)1, L(2)2 and/or L(2)3 Radius: R Cutting edge (tool) orientation: TO	G147 – G847	G147	Up to 8 external length and radius paired values can be activated by PLC. For deactivation use G148.	Independent of: • standard length compensation • external length compensation Additive on: • standard radius compensation. External radius compensation is overwritten!
Drill axis compensation of – 1st compensation group – 2nd compensation group	Length: H or $H_{ext.}$ L(2)1, L(2)2 and/or L(2)3	G78	G78 X22	The drill axis (the axis on which the length compensation takes effect) is switched using G78.	Length compensation values refer to the axes to which the compensation group relates.

Geometry compensations

Compensation method	Save under	Activate via	Programming (example)	Notes	Effect
Tool orientation	with Eulerian angles: φ (phi), δ (theta), ψ (psi) or in Cartesian coordinates: O(ρ_x, ρ_y, ρ_z) or specifying the axis of rotation u and the angle of rotation β: ROTAX (ux, uy, uz) O (β)	Coord(<n>)	Coord(2) G1 x10 y20 z30 O(90,90) or G1 x10 y20 z30 O(0,1,0) or O(1,0,0) ROTAX (1,0,1) O(90)	Selection of axis transformation "2": The tool's orientation is along the Y axis.	In case of active axis transformation, the tool compensation is performed along the TCS tool coordinate system.

8.2.6 Assigning

1. Make sure that the required compensation group (1 and/or 2) is released for the respective channel in MP 7050 00410.

Individual parameter	Effect	Default setting
1	Release/inhibit compensation group 1 (for H compensation value within the standard compensation and for external compensation)	1 (=released)
2	Release/inhibit compensation group 2 (for general compensation)	0 (=inhibited)

2. Use MP 7050 00420 to determine channel-specifically which logical axes are assigned to which length compensation/compensation group.
   The setting made here acts as default setting in the respective channel and can be modified via G78.
   G79 sets the active setting for compensation group 1 and/or 2 back to the default setting.

Individual parameter	Length compensation value which is to take effect on the assigned axis	Compensation group	Default setting (log. axis no.)
1	reserved	–	0
2	reserved	–	0
3	H or $L_{(1)3}$	1	3
4	$L_{(2)1}$	2	0
5	$L_{(2)2}$	2	0
6	$L_{(2)3}$	2	0

Example: assigning of 4 drill axes

Individual parameter	Length compensation value which is to take effect on the assigned axis	Com- pensation group	Setting exam- ple (log. axis no.)
1	reserved	–	0
2	reserved	–	0
3	H or $L_{(1)3}$	1	3
4	$L_{(2)1}$	2	4
5	$L_{(2)2}$	2	5
6	$L_{(2)3}$	2	6

Example: constant three-dimensional tool shift (3 axes)

Individual parameter	Length compensation value which is to take effect on the assigned axis	Com- pensation group	Setting exam- ple (log. axis no.)
1	reserved	–	0
2	reserved	–	0
3	H or $L_{(1)3}$	1	0
4	$L_{(2)1}$	2	1
5	$L_{(2)2}$	2	2
6	$L_{(2)3}$	2	3

3. If compensation values are specified by the PLC (in case of ”external” or ”general compensation”, use MP 7050 00430 to determine channel-specifically for which type of tool the compensation group is to be used. The following are defined as types of tools:
   1 = drilling tool, 2 = milling tool, 3 = lathe tool, 4 = anglehead tool

Individual parameter	Effect	Default setting
1	Determines the type of tool used in compensation group 1 (for external compensation)	2 (=milling tool)
2	Determines the type of tool used in compensation group 2 (for general compensation)	2 (=milling tool)

Geometry compensations

Example: a drilling tool is used under compensation group 2 (general compensation)

Individual parameter	Effect	Setting example
1	Determines the type of tool used in compensation group 1 (for external compensation)	2 (=milling tool)
2	Determines the type of tool used in compensation group 2 (for general compensation)	1 (=drilling tool)

Example: an anglehead tool is used under compensation group 2 (general compensation):

Individual parameter	Effect	Setting example
1	Determines the type of tool used in compensation group 1 (for external compensation)	2 (=milling tool)
2	Determines the type of tool used in compensation group 2 (for general compensation)	4 (=anglehead tool)

4. Use MP 9020 00010 to specify in which unit of measure the values of geometry compensation tables are stored within the CNC and how the control unit is supposed to interpret the shift values transmitted by the PLC (in case of external or general compensations).

☞ **Changes in MP 9020 00010 also influence zero shift tables, tables for the "inclined plane" and the interpretation of zero shift values transmitted by the PLC!**

CAUTION
Danger of damage to the machine or the workpiece!
Do not initiate a control reset during machining.

5. Initiate a control reset.

Applies additionally when standard compensation is used:

1. Create a new geometry compensation table:
   MANAGE ▶ EDIT NEW FILE ▶ GEO TABLE

☞ **The file name extension of geometry compensation tables is defined via MP 3080 00200 (individual parameter: 2).**
Presetting: *.geo.

2. Enter name.
3. Define directory for the table:
   via SWITCH DIRECTORY softkey
4. Press OK softkey.
5. Fill table with length and radius compensation values for machining.

Applies additionally when ”external” or ”general compensation” is used (compensation values provided via PLC):

1. Adapt PLC program module –B22WZKOR for the provision of the length and radius compensation and integrate it into the PLC program.

8.2.7 Activating

Standard compensation

Precondition:

- Geometry compensation table for the machining program exists and contains the correct compensation values of the required tools.
- Tools for machining are available.

1. Activate geometry compensation table (see page 8–30).
2. Change machining tool.
3. Prior to the machining phase, activate
   the required radius compensation value via D address
   – or/and –
   the required length compensation value via H address.

External and general compensation

Precondition:

- Any geometry compensations still active must be taken into account when programming other geometry compensations because these may have an additive effect or overwrite each other mutually.
- Tools for machining are available.

1. By programming the T address, the next tool available for machining is identified and its relevant compensation values are transferred to the control unit when it is exchanged (via M function) using the PLC program module –B22WZKOR.
   You should then program “WAIT” in order to halt the block preparation. This is the only way to make sure that the current values will be used in the subsequent activation of the compensation data.

2. Activate a compensation value pair of the “external compensation” using G145–G845,
   or
   a compensation set of the “general compensation” using G147–G847.

8.2.8 Deactivating

Standard compensation

Precondition:

- The tool must no longer be involved in the machining process.

1. Deactivate active compensation values by programming D0 and H0 after the machining phase.

External and general compensation

Precondition:

- The tool must no longer be involved in the machining process.

1. Deactivate the active compensation pair of the “external compensation” using G146,
   or
   the active compensation set of the “general compensation” using G148.

8.2.9 Structure of geometry compensation tables

If you use the ”standard compensation” as geometry compensation (see page 8–13 ff.) the compensation values will be provided in a geometry compensation table. A maximum of 48 compensation value pairs (radius: D, length: H) can be stored in each table.
The control unit saves these tables as ASCII files with an accurately defined structure.

The table editor is available (see PNC operating instructions) for comfortable editing of these tables at the control unit.
The table editor starts automatically when a geometry compensation table has been selected in GOM MANAGE and you press the EDIT softkey (for prerequisites, refer to ”Relevant MACODA parameters”, MP 3080 00200).
If the control unit loads the selected table into the NC editor (ASCII editor), the required preconditions have not been fulfilled. In these cases, we recommend that you close the editor without changing the table and edit the table manually via MANAGE ▶ TABLES and subsequent path/ file selection via the table editor.

☞ **Program block -B24FLTAB (access to geometry compensation or axis zero shift tables) is no longer supported from PNC software version 5.1.18 on!**

Table name

The table name is composed of a freely selected “file name” and a ”file name extension” (e.g. test.geo).
The file name extension can be changed, for details, see MP 3080 00200 on page 8–31.

Structure

A geometry compensation table is structured as follows:

- 1 headline containing an internal system coding of the table
- 48 lines for radius compensation values (2nd to 49th line)

  `D 1X =` <Value>
  `:`
  `D48X =` <Value>

- 48 lines for length compensation values (50th to 97th line)

  `H 1Z =` <Value>
  `:`
  `H48Z =` <Value>

Both positive and negative compensation values are permitted as <value>.

Relevant G functions:

G22 K <*Path/table name*>:
Activating a geometry compensation table via NC block.

Relevant MACODA parameters (MP):

3080 00200: Determines in individual parameter 2 which file name extension the control unit uses as identification criterion for geometry compensation tables (standard: “geo”). Selected files with this file name extension are automatically opened with the table editor (and not with the ASCII editor) for editing. Please note that the entry is case sensitive: ”TEST. GEO” and ”test.geo” designate different files for the control unit!

3080 00001: Search path.

9020 00010: Specifies in which unit of measure the table values are stored internally and how the control unit is supposed to interpret the compensation values transmitted by the PLC (in case of external or general compensations).

☞ **Changes in MP 9020 00010 also influence zero shift tables, tables for the ”inclined plane” and the interpretation of zero shift values transmitted by the PLC!**

8.2.10 Activating

To activate a geometry compensation table via NC block, use G22 with address K:

`G22 K`<*Path*><*Table name*>

☞ **If <*Path*> has not been specified, the PNC uses the search path (MP 3080 00001) to search for the <*Table name*>.**

Example:
Activating the geometry compensation table ”test.geo” in the root directory:
`G22 K/test.geo`

To activate the compensation values, proceed as follows:

1. Activate the geometry compensation table which contains the required compensation values.
2. Activate the desired radius compensation value (1–48) via D address (example: 12th radius compensation value: D12)
   – or/and –
   activate the desired length compensation value (1–48) via H address (example: 32nd length compensation value: H32).

To deactivate active compensation values, program the D or H address, respectively, with the following value ”0” (e.g. D0 or H0, respectively).

8.3 Compensation switchover

Function:

Machining on machine tools may be performed using tools of different sizes, with the tool clamped in different directions, depending on the machine, or oriented in any direction in space in case of the corresponding machine kinematics.

The G78 compensation switchover function assigns the different **length compensations** of the individual functions to the geometry compensation of either

- the individual directions of the current workpiece coordinate system (WCS) or
- the directions of the tool coordinate system (TCS).

An assignment with respect to the directions of the workpiece coordinate system (WCS) is possible if the tool is aligned perpendicularly to the current working plane and its orientation remains constant during machining.
Examples: drilling and turning, milling of plane surfaces.

An assignment of compensations referring to the directions of the tool coordinate system (TCS) is necessary if the orientation of the tool changes during machining, e.g. the milling of free-form surfaces. This allows for a tool length compensation at variable tool orientation. An active axis transformation (e.g. 5 axis or 6 axis transformation) is necessary for this compensation. The compensation values are taken into account within the axis transformation.

The switchover can be activated manually or in the part program.

The tool length compensations are divided into 2 compensation groups within the PNC:

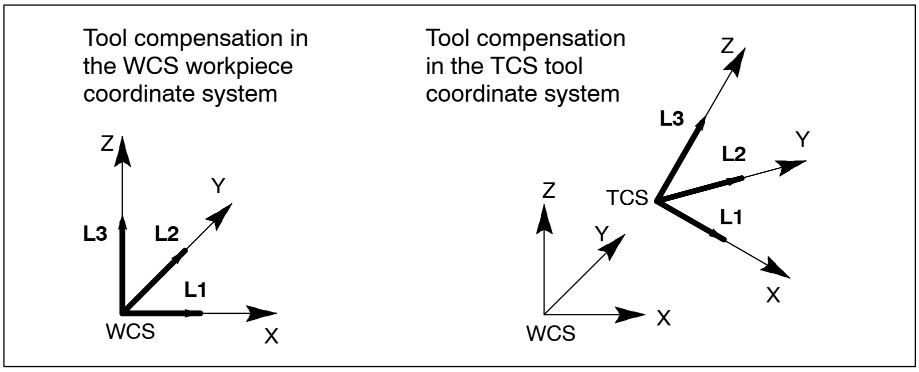
- **1st compensation group** refers to axes to which the following compensations apply:

  H and $H_{ext}/L_{(1)3}$ of the “external tool compensation” (G145–G845)

- **2nd compensation group** refers to axes to which the following compensations apply:

  $L_{(2)1}$, $L_{(2)2}$, $L_{(2)3}$ of the general compensation (G147–G847).

Compensation switchover



Precondition:

- Geometry compensations (see page 8–10 ff.) have been properly established.

Restrictions:

- Drill axes must be linear axes and be suitable for the machine kinematics. The tool has the corresponding drive (e.g. spindle).
- The axis addresses programmed under G78 refer to the workpiece or tool coordinates (XT, YT, ZT).
- The required compensation group (1 and/or 2) has to be released in MP 7050 00410.
- A certain length compensation within a compensation group may not be assigned to several axes at one point in time (see MP 7050 00420).

Relevant MACODA parameters (MP):

7050 00410: Channel-specific release of
compensation group 1 (for H compensation value within the standard compensation and for external compensation)
and/or
compensation group 2 (general compensation)
(individual parameter=1: compensation group is released)

7050 00420: Channel-specific determination which logical axes are assigned to the individual length compensations/compensation groups.

7050 00430: Channel-specific determination for which type of tool the compensation groups 1 and 2 are to be used:
Drilling tool, milling tool, lathe tool, anglehead tool.
In this manner, you tell the control unit how many compensation values are transferred by the PLC in case of ”external” or ”general compensation”.

Compensation switchover

Relevant G functions:

G78..: Switch over of the drill axis.
Program-controlled assignment of an existing length compensation to a coordinate (in the WCS) or to a working range coordinate (in the TCS) on the channel.
For parameters, see DIN Programming Instructions.

G79.. .: Deactivating drill axis switching.
Compensation groups 1 and/or 2 can be reset to the settings in MP 7050 00420.
For parameters, see DIN Programming Instructions.

G145..845: External tool compensation ON

G147..847: General tool compensation ON

G81..86, G186: Boring cycles

Relevant CPL functions:

NCF(”G78”) Provides information on whether G78 or G79 is active on the respective channel.

SD(10,1) Provides the number of the last programmed drill axis (L3 or H compensation) for compensation group 1.

SD(10,1,1) the same as SD(10,1)

SD(10,2) Provides the number of the active drill axis for compensation group 1.

SD(10.2,1) the same as SD(10.2)

SD(10,1,2) Provides the number of the last programmed drill axis (L3 compensation)

SD(10,2,2) Provides the number of the active drill axis for compensation group 2.

MCODS(...) see ”Compensations” under MCODS in the CPL Programming Manual

8.3.1 Assigning

1. Make sure that the required compensation group (1 and/or 2) is released for the respective channel in MP 7050 00410.
2. Use MP 7050 00420 to determine channel-specifically which logical axes are assigned to which length compensation.
   The setting made here acts as default setting in the respective channel and can be modified via G78.
   G79 sets the active setting for compensation group 1 and/or 2 back to the default setting.

   Example: the Z axis (log. axis number 3) is assigned the length compensation H or L3 from compensation group 1. Because the compensation values are taken into account automatically, it becomes usable as drill axis.

Compensation switchover

Individual parameter	Length compensation value which is to take effect on the assigned axis	Compensation group	Default setting (log. axis no.)
1	reserved	–	0
2	reserved	–	0
3	H or L3	1	3
4	L1	2	0
5	L2	2	0
6	L3	2	0

3. If compensation values are specified by the PLC (in case of "external" or "general compensation", use MP 7050 00430 to determine channel-specifically for which type of tool the compensation group is to be used. The following are defined as types of tools:
   1 = drilling tool, 2 = milling tool, 3 = lathe tool, 4 = anglehead tool

Individual parameter	Effect	Default setting
1	Determines the type of tool used in compensation group 1 (for external compensation)	2 (=milling tool)
2	Determines the type of tool used in compensation group 2 (for general compensation)	2 (=milling tool)

CAUTION
Danger of damage to the machine or the workpiece!
Do not initiate a control RESET during machining.

4. Initiate a control RESET.

8.3.2 Activating

Precondition:

- Drill axis switching is activated (see Section 8.3.1).

1. Program G78.
   For possible parameters, see DIN Programming Instructions.
2. Activate the corresponding geometry compensations (H, G145 – G845 or G147 – G847).

8.3.3 Deactivating

Precondition:

- The drill axis is no longer in contact with the workpiece.

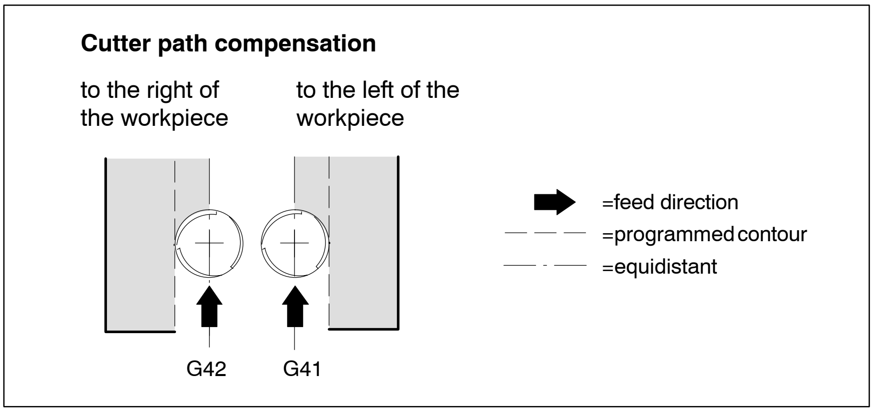
1. Deactivate the active geometry compensations using the respective functions (H0, G146 or G148).
2. Program G79.
   It is optionally possible to reset either all compensation groups or a specific compensation group to the settings according to MP 7050 00420.

8.4 Path compensation (cutter radius compensation)

Function:

The path compensation leads to an offset of the tool with right-angle constant distance **to the left (G41)** or **to the right (G42)** of the programmed contour.
This “offset” path is referred to as “equidistant“. The distance between the equidistant and the programmed path depends on the active radius compensation value of the current tool.

Principle:



The radius compensation value of the tool can be made available by the following geometry compensations:

- Standard tool compensation (D)
- External tool compensation (R_{ext})
- General tool compensation (R)

☞ **The radius compensations of the standard, external and general compensation act additively in part or exclude each other.**

Preconditions:

- Tools and their dimensions (compensations) are measured for the machining process.
- The machining program exchanges the tools in accordance with the machining phase.
- The PLC or tool compensation table provides the correct compensations for the tools used.

Restrictions:

- If the control unit is in a circular mode (e.g. G2, G3, G5), **no** traversing movement must be programmed in the G40, G41 and G42 blocks.
- In case of active path compensation, the following functions are not permitted:
  - G17 to G20 (plane change-over)
  - G32 (tapping without compensation chuck)
  - G70, G71 (inch/metric change-over)
  - G74 (approach reference point)
  - G75 (probe input)
  - G92 (set actual value)
- G76 (traverse to machine-oriented absolute axis position) deactivates an active path compensation.

Relevant G functions:

G41: Switching path compensation on **to the left of the workpiece** (seen in the direction of machining), referred to positive compensation values.
If no traversing movement is programmed in the G41 block, the control unit will activate the compensation immediately and vertically to the next traversing block.
If a traversing movement is programmed in the G41 block, the control unit will activate the compensation linearly on the way to the end point of the traversing movement.

G42: Switching path compensation on **to the right of the workpiece**. Apart from that, identical to G41.

G40: Path compensation OFF.
If no traversing movement is programmed in the G40 block, the control unit will activate the compensation immediately and vertically to the previous traversing block.
If a traversing movement is programmed in the G40 block, the control unit will deactivate the compensation linearly on the way to the end point of the traversing movement.

G145..G845: External tool compensation ON

G147..G847: General tool compensation ON

Relevant interface signals:

NC-O20.5: Set: G41 is active on the respective channel.

NC-O20.6: Set: G42 is active on the respective channel.

If **both** signals are **not** active, the path compensation is deactivated (G40).

Relevant CPL functions:

NCF("G40") Provides information on whether G40, G41 or G42 is active on the respective channel.

8.4.1 Assigning

Using a geometry compensation table:

1. Create a new geometry compensation table:
   MANAGE ▶ EDIT NEW FILE ▶ GEO TABLE

☞ **The file name extension of geometry compensation tables is defined via MP 3080 00200 (individual parameter: 2).**
Presetting: *.geo.

2. Enter name.
3. Define directory for the table:
   via SWITCH DIRECTORY softkey
4. Press OK softkey.
5. Fill table with length and radius compensation values for machining.

When ”external” or ”general geometry compensation” (compensation values provided via PLC) is used:

1. Adapt PLC program module –B22WZKOR for the provision of the length and radius compensation.

8.4.2 Activating

Using a geometry compensation table:

1. Activate the desired table via G22 K<*path/table name*>.
2. Program D.. and H.. to select the respective radius and length compensation value from the activated table.
3. Select a starting point which allows a nearly tangential approach to the contour. Make sure that activation of the compensation does not lead to a damage of the contour.
4. Program G41 or G42.

When ”external” or ”general geometry compensation” (compensation values provided via PLC) is used:

1. Select tool using T.. tool address and change the tool using the M function: the relevant tool compensations will be selected automatically.
2. Program G145...G845 (external tool compensation) and/or G147...G847 (general tool compensation).
3. Select a starting point which allows a nearly tangential approach to the contour. Make sure that activation of the compensation does not lead to a damage of the contour.
4. Program G41 or G42.

8.4.3 Deactivating

Precondition:

- The tool is no longer in contact with the workpiece and has moved away from the contour.

Using a geometry compensation table:

1. Make sure that deactivation of the compensation does not lead to a damage of the contour. Approach an adequate end point for this purpose.
2. Program G40.
3. Program D0, H0 if required.

When ”external” or ”general geometry compensation” (compensation values provided via PLC) is used:

1. Make sure that deactivation of the compensation does not lead to a damage of the contour. Approach an adequate end point for this purpose.
2. Program G40.
3. Program G146 (external tool compensation) and/or G148 (general tool compensation), if required.

9 Safety / Monitoring

9.1 Electronic limit switches (software limit switches)

Function:
Prevents an axis from deviating from a defined physical traversing range due to positional data.

Features:
- Two physical traversing ranges (travel limit ranges) can be defined for each axis. The ranges may overlap.
- The desired traversing range can be selected via two interface signals at the respective axis interface.
- Monitoring of the limit switches can be deactivated (suppressed) dynamically by means of an interface signal.

Restrictions:
- As long as the control unit does not know the absolute axis position (see interface signal NC-O15.7), the electronic limit switches will not take effect for this axis (reference point has to be known).
- The limits of the area must not exceed the position limits (S-0-0049, S-0-0050) set in the drive.
- Exceeding the selected active limit switch range results in a runtime error.

Relevant MACODA parameters (MP):

1020 00001:	Software limit switch positive.
1020 00002:	Software limit switch negative.
1020 00003:	2. Software limit switch positive.
1020 00004:	2. Software limit switch negative.

Relevant interface signals:

NC-I1.4:	”Activate travel limit range bit 0” (axis interface).
NC-I1.5:	”Activate travel limit range bit 1” (axis interface). Selection of the desired travel limit range: NC-I1.4 and NC-I1.5 = 0: Range 1 NC-I1.4 = 0 and NC-I1.5 = 1: Range 2
NC-I3.2:	”Suppress software limit switch” (axis interface). High signal: selected range is suppressed
NC-O15.7	”Reference point is known” (axis interface). Low signal: limit switches for the axis are not effective!

9.1.1 Activating

1. Adapt MP 1020 00001 through 1020 00004.

CAUTION
Danger of damage to the machine or the workpiece!
Do not initiate a system control reset during machining.

2. Initiate a system control reset (DIAGNOSTICS ▶ RESET FUNCTIONS ▶ SYSTEM CONTROL RESET).

☞ **Unless the relevant interface signals are activated by the PLC, range (area) 1 is always selected and activated if the value ”99999.999” has not been parametrized for the corresponding axis in MP 1020 00001.**

Switching off (permanent)

1. Deactivate travel limit range 1 and 2:
   MP 1020 00001= 99999.999.

Suppressing (switching off temporarily)

1. Set NC-I3.2 ”Suppress software limit switch” at the relevant axis interface.

Moving from one travel limit range to the other range

Precondition:

- Active traversing range has been selected at the relevant axis interface via NC-I1.4 and NC-I1.5.

1. Suppress active travel limit range (set NC-I3.2 at the relevant axis interface).
2. Move axis to the other travel limit range manually.
3. Select other traversing range at the relevant axis interface via NC-I1.4 and NC-I1.5.
4. Reset the NC-I3.2 signal set previously at the relevant axis interface.

9.2 Torque reduction

Function:

Torque reduction allows a temporary influencing of the set torque limit value of an axis (S-0-0092: bipolar torque limit value).

Features:

- The function can be activated/deactivated via the NC-I3.7 interface signal ”Torque reduction”.
- Feedback of the function status is received via the axis-related NC-O17.6 interface signal ”Torque reduction active”.
- The desired temporary torque limit value of the axis can be entered via MP 1003 00010 and changed program-controlled using G177. A torque limit value change via G177 becomes active with the next positive edge of the axis-related NC-I3.7 interface signal.

Restrictions:

- Switching the torque reduction on/off is only permitted with the axis standing still.
- S-0-0092 has to be interpreted as % of the nominal torque in the drive (in S-0-0086, ”percent weighting” has to be set as type of weighting).

Relevant MACODA parameters (MP):

1003 00010: Presetting for reduced maximum axis torque.

Relevant G functions:

G177: Torque reduction.

Relevant interface signals:

NC-I3.7: ”Torque reduction” (axis interface).
L/H edge: switch on torque reduction.
H/L edge: switch off torque reduction.
When switched off, the torque limit value set in S-0-0092 after the last SERCOS phase start-up will be set in the drive again.

NC-O17.6: ”Torque reduction active” (axis interface).

9.2.1 Assigning

1. Adapt MP 1003 00010 correspondingly for all relevant axes.

CAUTION
Danger of damage to the machine or the workpiece!
Do not initiate a system control reset during machining.

---

2. Initiate a system control reset (DIAGNOSTICS ▶ RESET FUNCTIONS ▶ SYSTEM CONTROL RESET).

9.2.2 Activating/deactivating

Precondition:

- The necessary axis-related interface signals are evaluated or generated, respectively.
- The axis is standing still.

CAUTION
Any random or unintended change of the torque limit value can cause damage to the workpiece/tool or lead to hazardous states on the machine.
Therefore consider the possible consequences before you influence the torque limit value.

---

1. If required, program the desired torque limit value using G177 if you do not want to use the value from MP 1003 00010.
2. Specify the L/H edge at NC-I3.7 of the relevant axis interface to activate the function
   or
   specify the H/L edge at NC-I3.7 of the relevant axis interface to deactivate the function.

9.3 Limit rapid travel

Function:

When part programs are activated, it is in many instances not desired for rapid travel blocks to be traversed at maximum speed. A reaction by the operator using ”Override” (see page 7–23) may be too late in case of incorrect programming to avoid potential damage.
That is why the control unit reduces the rapid travel movements of synchronous axes (G0, G10, G200, ...) on the respective channel in dependence on the channel-related NC-I1.7 interface signal ”Limit rapid travel” to

- a parametrizable value (in case of MP 7030 00110 > 0), or
- to the feedrate speed programmed last on the channel (in case of MP 7030 00110 = 0).

☞ **You can also use the function to limit rapid travel as long as no reference points have been approached yet.**

If the ”Override” function has been activated on the relevant channel, it will also take effect on limit rapid travel. In this case, an override of 100% corresponds to the limit rapid travel.

Restrictions:

- Is only effective in the modes ”Manual input NC block”, ”Automatic”, ”Program block” and ”Single step”.
- Does not take effect on spindle speeds or on the active path feed.
- A change of the NC-I1.7 interface signal does not act on program blocks which were already processed by block preparation at the time the change was performed.
  If this behavior is not desired, you have to stop the machining process using ”Feed hold”, cancel blocks already prepared using ”Cancel distance to go” (see page 7–29) and continue machining by means of cycle start.

Relevant MACODA parameters (MP):

7030 00110: Value of limited rapid traverse velocity.

Relevant interface signals:

NC-I1.7: ”Limit rapid travel” (channel interface).

9.3.1 Assigning

1. Adapt MP 7030 00110.

CAUTION
Danger of damage to the machine or the workpiece!
Do not initiate a system control reset during machining.

---

2. Initiate a system control reset (DIAGNOSTICS ▶ RESET FUNCTIONS ▶ SYSTEM CONTROL RESET).

9.3.2 Activating/deactivating

1. Set NC-I1.7 at the channel interface in order to activate the limit rapid travel.
   Reset NC-I1.7 in order to deactivate the limit rapid travel.

9.4 Axis position monitoring

Function:

Reports via interface signal in which area the axis is located with reference to a parametrizable position (⇒ MP 2010 00110). The position always refers to the axis coordinate system (ACS).
Any active LSECs (see page 10–1 ff.) are taken into account by the control unit automatically for the current axis position.
Please note for signal generation:

current axis position ≥ parametrized position: signal set
current axis position < parametrized position: signal not set

The function is suited for the activation of other functionalities by the PLC.

Features:

- Up to 8 positions per axis can be defined with one corresponding interface signal each.
- In case of rotary axes, the monitoring position may refer to the complete traversing range or to the range between 0 and 360 degrees.
- It is possible to set whether the monitoring is performed at referenced or non-referenced axis only.

Restrictions:

- Monitoring positions refer to the axis coordinate system
- A maximum of 64 monitoring positions throughout the system

Relevant MACODA parameters (MP):

2010 00100: Monitored axis.
2010 00110: Position (of the point).
2010 00120: Interface signal (of the point).
2010 00130: Cyclic monitoring of rotary axes.

Relevant interface signals:

NC-O18.x: ”Axis position x” (axis interface).
High signal: current axis position ≥ parametrized position
Low signal Signal: current axis position < parametrized position

☞ **In order to monitor two-dimensional areas with paraxial limits, the function ”Area control” is available (see page 9–10 ff.).**

9.4.1 Activating

1. Adapt all individual parameters of MP 2010 00100 through MP 2010 00130 which have the same index, for each desired monitoring position.

CAUTION
Danger of damage to the machine or the workpiece!
Do not initiate a control reset during machining.

2. Initiate a control reset.

9.5 Distance to end point

Function:

Output of an axis interface signal as soon as the current distance to the programmed end point of an NC block falls below a defined length. This way it is possible for the PLC to e.g. initialize or prepare own functions before the axis has reached its actual end point. This may, under certain circumstances, reduce the overall time required for a manufacturing process.

Restrictions:

”Distance to end point” has no effect for

- slave axes with active axis coupling
- handwheel mode
- operation mode ”Approaching the reference point”

Relevant MACODA parameters (MP):

1015 00010: Distance to end point.

Relevant interface signals:

NC-O17.2: ”Axis near end point” (axis interface)

9.5.1 Assigning

1. Enter the desired distances in mm or degrees, respectively, into the corresponding individual parameters of MP 1015 00010.

CAUTION
Danger of damage to the machine or the workpiece!
Do not initiate a control reset during machining.

---

2. Initiate a control reset.

9.5.2 Deactivating

1. Set the corresponding individual parameters in MP 1015 00010 to value ”–1.0”.

CAUTION
Danger of damage to the machine or the workpiece!
Do not initiate a control reset during machining.

---

2. Initiate a control RESET.

9.6 Area control

Function:

By means of the area control, it is possible to monitor 10 two-dimensional rectangular areas with paraxial limits throughout the system.

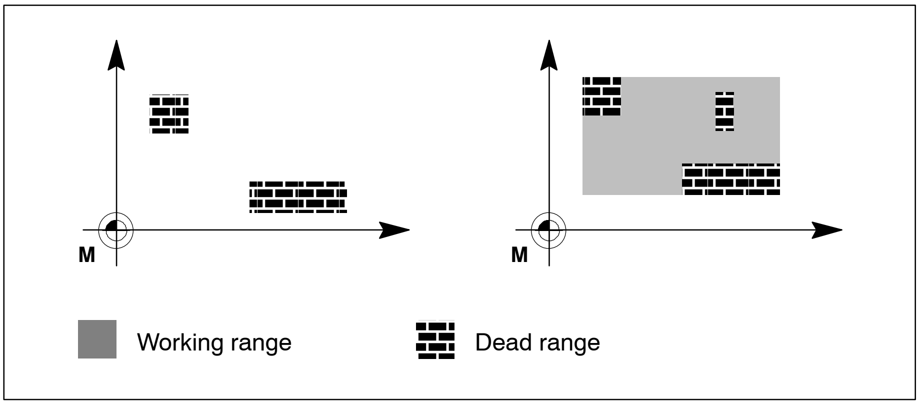
Traversing movements which ”infringe” an activated area, produce a runtime error in automatic mode; in jog mode, it is only possible to move the axis up to the limit of the area. In this case, the PNC will display a warning message.

A control area can either be parametrized as working range or as dead range:

- Working range:
  Traversing movements exceeding an active working range are not possible. The limits are part of the working range.
  When a working range is activated, the current machine position has to be within the working range.
- Dead range:
  Traversing movements into a dead range are not possible. The limits are part of the dead range.
  When a dead range is activated, the current machine position has to be outside the dead range.

Several control areas may be active simultaneously within one channel. Thus it is possible to e.g. monitor one working range and several dead ranges located inside the working range simultaneously.

Examples: Dead ranges and working range



To be able to define an area, the control unit requires the basic information of which system axes define the plane where a control area is to be located. These data can be parametrized exclusively via MP 8002 00001 and MP 8002 00002.
Any further data such as area position, size or type (dead/working range) can be parametrized via configuration parameters as presetting and changed program-controlled via AREADEF NC command. The program-controlled change of a control area can also be deactivated selectively using MP 8002 00032.

Area control

The AREADEF and AREAVALID NC commands are available to activate/deactivate control areas.

Restrictions:

- A max. of 10 control areas may be defined throughout the system.
- To activate a control area, all system axes involved in the area have to be present on the current channel.
- Area control is possible only if the control unit knows the absolute position of all axes involved (axes referenced).
- An axis can only be transferred to another channel (”Axis transfer” function) if the respective axis is not involved in an active control area on the channel from which the axis is transferred.
- If all axes of a control area are transferred to another channel, the control unit replaces all data of the control area both on the channel from which the axes are transferred as well on the new channel with the presetting values in the corresponding configuration parameters. The control areas are deactivated in the process.
- In automatic mode, only those activated areas are monitored which are located in the active plane of the channel. In jog mode, all active areas of the channel are monitored.
- The two axes involved in a dead range may not be jogged simultaneously.
- Traversing movements initiated by handwheel are not monitored.

Relevant NC functions:

AREADEF: Parametrizing control area with regard to position, extension and type. Activation/deactivation is possible additionally.
The programming of an area using AREADEF is only valid for the current channel and will be effective there until the area is changed by means of a new AREADEF command.
Only one single area can be parametrized for each AREADEF block. For information on syntax, see DIN Programming Instructions.

AREAVALID: Activating/deactivating control area.
The control unit has to know the position, extension and type of the area (either by means of parametrization in MACODA or a previously used AREADEF command).

Syntax: AREAVALID(<*Area number*>,<*Action*>)

Please note: <*Area number*>: 1..10; –1=all areas
<*Action*>: 0: deactivate
1: activate

Relevant MACODA parameters (MP):

For definition of all 10 control areas in the PNC, use the configuration parameter group 8002.
Directly after start-up, the control area data are preset with precisely these values.
The control areas are still deactivated.

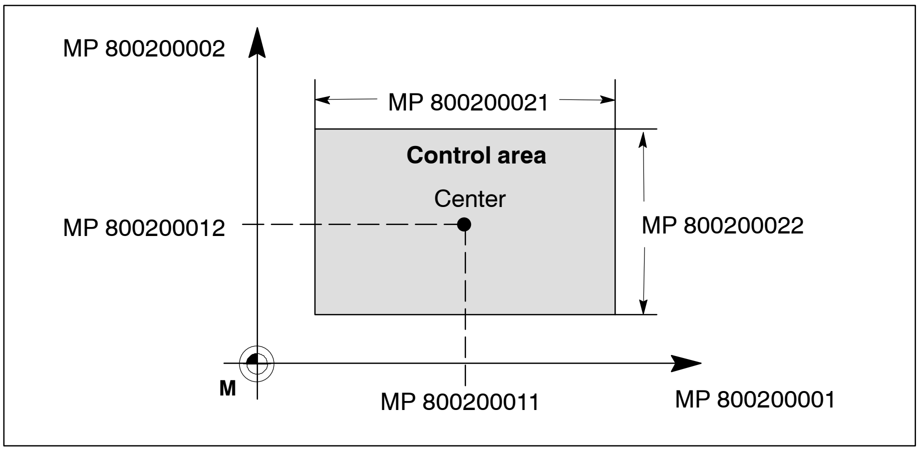
8002 00001: System axis number of 1st dimension of area.
8002 00002: System axis number of 2nd dimension of area.
Both parameters have to include the system axis number of the axes defining the plane of the control area.
The control area limits are always parallel with these axes.

8002 00011: Center of area, 1st dimension.
8002 00012: Center of area, 2nd dimension.
Both parameters determine the center point of the control area within the machine coordinate system.

8002 00021: Enlargement of area, 1st dimension.
8002 00022: Enlargement of area, 2nd dimension.
Both parameters determine the side length of the control area relative to the unit of measurement set in the control unit.

8002 00031: Type of control area.
0: Area is not used
1: Dead range
2: Working range

8002 00032: Area is programmable.
0: Area cannot be modified using AREADEF
1: Area can be modified using AREADEF



9.6.1 Assigning

1. Make sure that the correct system axis numbers have been entered in MP 8002 00001 and MP 8002 00002 for all areas used.
2. Enter the data for the control area used into the remaining parameters of group 8002. The PNC uses these data as presetting.
   These can be overridden later using the AREADEF NC command, if the value ”1” has been assigned using MP 8002 00032.

CAUTION
Danger of damage to the machine or the workpiece!
Do not initiate a control RESET during machining.

---

3. If you have changed the configuration parameters, initiate a control reset.

9.6.2 Activating

Precondition:

- The control areas used have been assigned correctly in accordance with Section 9.6.1.
- The ”Inclined plane” function is not effective in connection with a system axis involved in the area control.
- All axes involved are present on the channel.
- All axes involved are referenced (axis-related interface signal NC-O15.7 ”Reference point is known” has been set).
- Handwheel mode is not activated.

1. Make sure that the machine position is located in the permitted range.
   When a working range is activated, the current machine position has to be within the working range.
   When a dead range is activated, the current machine position has to be outside the dead range.
   If required, move the axes to permitted machine coordinates.
2. Program AREAVALID(<*Area number*>,1) with the desired area number.
   If, in addition to activating, you wish to enter the data of a control area, use the AREADEF command with the corresponding parameters instead of the AREAVALID command (only possible if AREADEF is not blocked via MP 8002 00032).

9.6.3 Deactivating

CAUTION
Danger of damage to the machine or the workpiece!
After deactivation of control areas, the machine may traverse into dead ranges or leave the permitted working range again.

---

1. Program AREAVALID(<*Area number*>,0) with the desired area number (”–1” for all areas).

Area control

Notes:

10 Axis compensations

Axis compensations include
- leadscrew error compensation (LSEC)
- cross compensation (CCOMP)

10.1 Leadscrew error compensation (LSEC)

Function:

Compensates linearity errors of an axis occurring in the area of
- the axis measuring system and
- mechanical transmission of movement (motor axis). This also includes the backlash of an axis.

☞ **Servodyn drives are capable of compensating the backlash (reversing play) automatically via SERCOS parameter S-0-0058.**

Restrictions:
- Maximum 12000 compensation values for all axes to be compensated.
- The compensation values of an active cross compensation (CCOMP; see page 10–7 ff.) reduce the maximum number of LSEC values possible.

Commissioning of LSEC:
1. Determine compensation values for all relevant axes
2. Enter compensation values into the control unit
3. Activate the compensation values entered

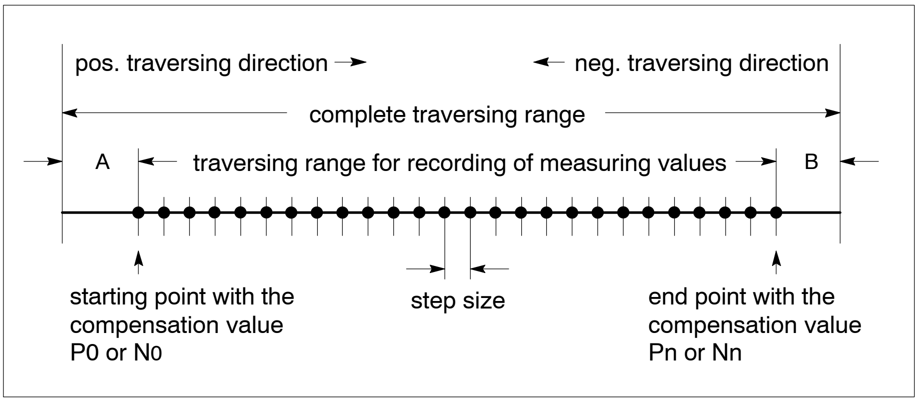
10.1.1 Determining the compensation values of an axis

1. Divide the **traversing range to be compensated** into a number of sections of equal size (recommended value: approx. 10–20 mm). By this grid you determine the positions within the traversing range where compensation values have to be determined.

☞ **The maximum number of compensation values, including LSEC and CCOMP (cross compensation), is 12000.**

☞ **For rotary or endless axes (cf. MP 1003 00004), it is only necessary to determine the compensation values for axis positions in the interval [0 .. 360]. In case of axis positions outside of this interval, the control unit first transforms the actual position into the interval [0 .. 360] before it takes into account the respective compensation value in this interval.**

☞ **Compensation values of Hirth axes (⇒ page 4–6) have to be on the Hirth grid (necessary for compensation value determination).**



Sep size: Distance between two consecutive measuring positions.
Value range (linear axis):
0.1 through 214748364.7 µm
Value range (rotary/endless axes):
0.1 through 359999.9*10^{-3} degrees

For axis positions between two consecutive measuring positions, the control unit automatically calculates a compensation from the two adjacent compensation values by means of linear interpolation.

Starting point: Lowest coordinate value for which a compensation value is determined.
Range of values (linear axes):
±214748364.7 µm
Value range (rotary/endless axes):
0.0 through 359999.9*10^{-3} degrees

If the subsequent measuring process is not started at this position, all measured compensation values will have to be sorted accordingly before they can be entered in the LSEC file.
If the starting point is not identical with the beginning of the complete traversing range, the PNC applies the compensation values of the starting point (P0 and N0) to range "A" (cf. figure).

End point: Highest coordinate value for which a compensation value is determined.
Range of values the same as "starting point", however, "end point" must be greater than the "starting point".
If the end point is not identical with the end of the complete traversing range, the PNC applies the compensation values of the end point (Pn and Nn) to range "B" (cf. figure).

2. Make sure that neither LSEC nor CCOMP (cross compensation) have been activated for the axis to be measured.
   To deactivate LSEC, see page 10–6.
   To deactivate CCOMP, see page 10–14.

3. Make sure that the axis has been referenced.
4. Move the axis to the start of the traversing range.
5. Determine the **compensation value P_i** (i=absolute number of the measuring position: 0 to n; cf. figure) for each measuring position in positive traversing direction with the help of an external measuring device (e.g. laser).
   Please note:

   Compensation value in µm = position setpoint (NC) – actual position (machine)

   All compensation values must be determined in the unit of measure ”micrometer” with a maximum resolution of 0.1 µm at present and may be in the range of ±3276.7 µm.

6. Having determined the last compensation value in positive traversing direction, move the axis to the end of the traversing range.
7. Now determine the **compensation value N_i** (i=absolute number of measuring position: n to 0; cf. figure) for each measuring position in negative traversing direction.
   Calculation of the compensation values as under item 5.

10.1.2 Entering compensation values

All compensation values for an axis must be entered in an ASCII file named ”`lsecc`<*Axis number*>`.tab`”.
<*Axis number*>: system axis index (cf. MP 1003 00004).

Examples:

`lsecc01.tab`	(Compensation values for axis 1)
:	
`lsecc08.tab`	(Compensation values for axis 8)
:	
`lsecc64.tab`	(Compensation values for axis 64)

For reasons of compatibility, it is also possible to use the name ”`lsec`<*Axis number*>`.tab`” for **axes 1 through 8**: <*Axis number*>: 0 through 7 !!

Examples:

`lsec0.tab`	(Compensation values for axis 1)
`lsec1.tab`	(Compensation values for axis 2)
:	
`lsec7.tab`	(Compensation values for axis 8)

☞ **If both file name types exist for an axis, the PNC automatically evaluates the contents of ”`lsecc`<*Axis number*>`.tab`”!**

Required structure of LSEC files:

```
<Starting point> <Step size> <Comment>
<P0> <N0> <Comment>
<P1> <N1> <Comment>
<P2> <N2> <Comment>
<P3> <N3> <Comment>
<P4> <N4> <Comment>
  :
<Pn> <Nn> <Comment>
```

- At least one blank must be programmed between the individual items in one line.
- Finish every line by hitting<enter>.

`<Starting point>` For more information, please refer to Section 10.1.1.

`<Step size>` For more information, please refer to Section 10.1.1.

`<Comment>` Optional character string. Is not evaluated by the control unit when the file is read. For example, you may enter a line number (starting with "0") and/or the axis position.
The string must not exceed the current line because 2 compensation values are always expected at the beginning of a new line.

<P_i> Compensation value for **positive** traversing direction effective at the axis position `<Starting point>` + i * `<Step size>`. The following applies:
Compensation value = position setpoint (NC) – actual position (machine)
Unit: µm
Resolution: 0.1 µm
Max. range of values: ±3276.7 µm

<N_i> Compensation value for **negative** traversing direction.
Apart from that, identical to <P_i>.

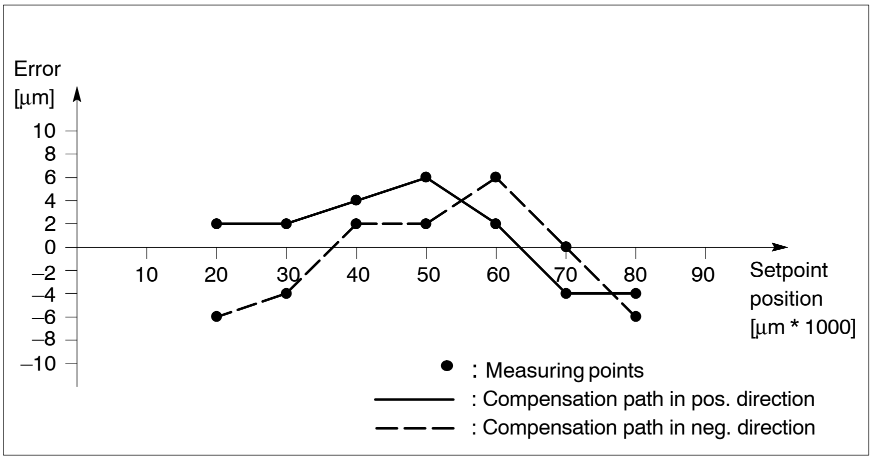
Programming example:

The traversing range of the X axis (0 to 90 mm) was subdivided into a grid with a step size of 10000 (10 mm).
The absolute axis position of 20000 µm (20 mm) was selected as starting point for LSEC.

Seven axis positions were measured for each the positive and negative traversing direction. The compensation values of the starting point apply to axis positions below 20 mm. The compensation values of the end point apply to axis positions greater than 80 mm.

The X axis is defined as 1st axis in the system. Therefore, the LSEC data for this axis must be entered in file "`lsecc01.tab`".

Leadscrew error compensation (LSEC)



Contents of file "`lsecc01.tab`":

```
20000.0  10000.0   (starting point 20mm; step size 10mm;
                    7 measurements)
2.0   -6.0  (position 20mm)
2.0   -4.0  (position 30mm)
4.0   2.0   (position 40mm)
6.0   2.0   (position 50mm)
2.0   6.0   (position 60mm)
-4.0  0.0   (position 70mm)
-4.0  -6.0  (position 80mm)
```

10.1.3 Activating

1. Copy or load all necessary LSEC files into the ”root directory” (”/”) or the ”User FEPROM” (”/usrfep”) subdirectory (for directory structure, refer to the ”Operating Instructions” manual).

CAUTION
Danger of damage to the machine or the workpiece!
Do not initiate a control reset during machining.

---

2. Initiate a control reset.

Compensation values of axes will not be activated if

- there is no LSEC file for an axis or
- an existing LSEC file contains values that are not allowed.

Possible error messages if an LSEC file contains values that are not allowed:

LSEC: Error when reading the line <*n*>, file <*m*> !
or
LSEC: One of the numbers in line <*n*>, file <*m*> is too high !
or
LSEC: Maximum number of LSEC values in line <*n*>, file <*m*> is exceeded !

Meaning:

<*n*> Line number (1,2,3...)
<*m*> Name of incorrect LSEC file:

lsec0.tab ... lsec7.tab:	”old” LSEC file name syntax
lsecc01.tab ... lsecc64.tab:	”new” LSEC file name syntax

10.1.4 Deactivating

1. Rename the relevant LSEC files existing in the ”root directory” (”/”) or the ”User FEPROM” (”/usrfep”) subdirectory.
   We recommend that you only change the file name extension ”tab” to ”off”, for example (e.g. lsecc03.tab ⇒ lsecc03.off).

CAUTION
Danger of damage to the machine or the workpiece!
Do not initiate a control reset during machining.

---

2. Initiate a control reset.

The respective LSEC compensation values will no longer be active after start-up.

10.2 Cross compensation (CCOMP)

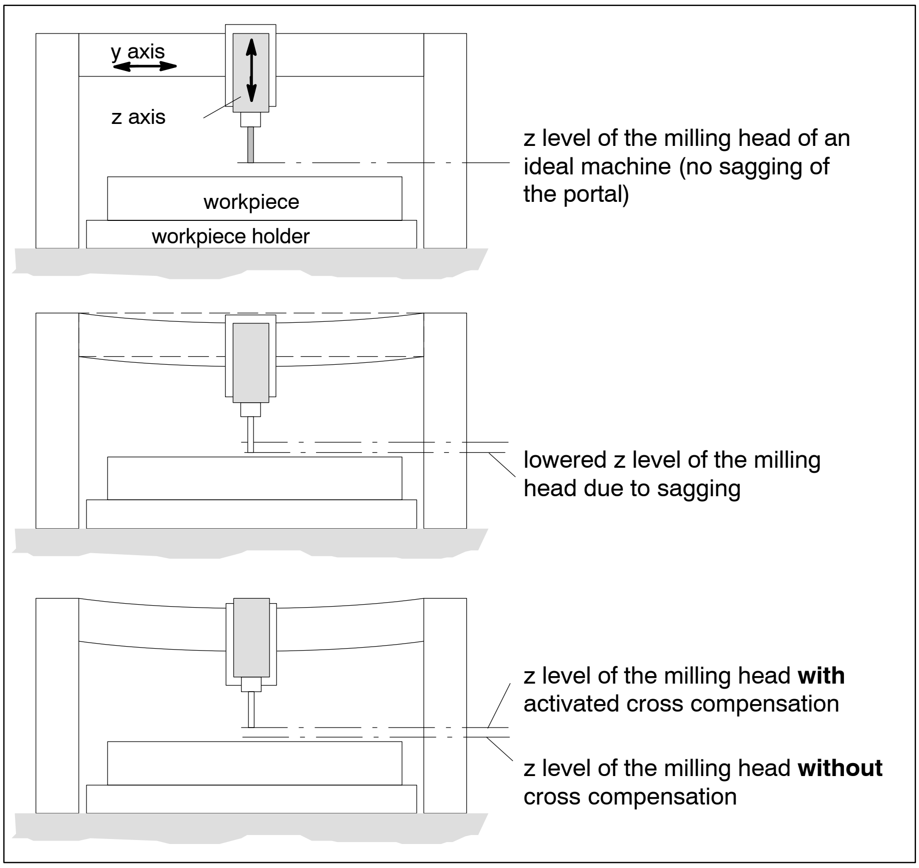
Function:

Compensates the geometric deviation between theoretical and actual position of an axis, if this deviation is dependent on the actual positions of max. two reference axes.

The axis to be compensated will also be referred to as ”target axis” in the following.

Example:

Portal milling machine. The sagging of the Z axis is to be compensated in dependence on the Y axis position.
⇒ target axis (axis to be compensated): Z
⇒ reference axis: Y



Restrictions:

- Maximum 12000 compensation values for all axes to be compensated.
- The number of compensation values of active LSEC tables (see page 10–1 ff.) reduces the maximum number of cross compensation values possible.
- A max. of two reference axes can be taken into account for a target axis.

10.2.1 Assigning

1. If you wish to use the CCOMP function, commission LSEC first. This is to make sure that the corresponding linearity error of the reference axis is minimized when determining the CCOMP values.
2. Determine compensation values for all relevant axes
3. Enter compensation values into the control unit
4. Activate the compensation values entered

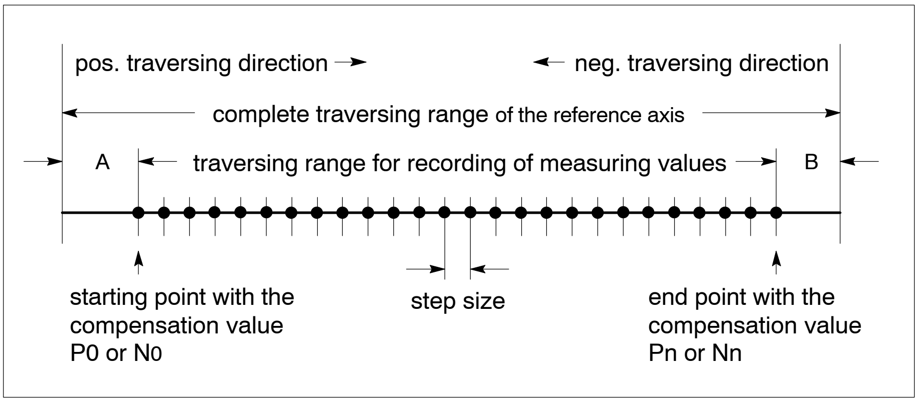
Determining the compensation values of an axis

1. Divide the **traversing range of the reference axis (axes)** into a number of sections of equal size.
   By this grid you determine the positions within the traversing range where compensation values for the target axis have to be determined.
   If you have to use two reference axes (positions of the target axis are influenced by two axes), these may have different grids.

☞ **The maximum number of compensation values, including CCOMP and LSEC, is 12000. CCOMP and LSEC compensation values may have different grids.**

☞ **If you use rotary or endless axes (cf. MP 1003 00004) as reference axis, it is only necessary to determine the compensation values for axis positions in the interval [0 .. 360]. In connection with CCOMP, axis positions outside of this interval are always transformed into the interval [0 .. 360[.**

☞ **If you use a Hirth axis (⇒ page 4–6) as reference axis, the positions have to be on the Hirth grid (necessary for compensation value determination).**



Division of the traversing range of the reference axis

Cross compensation (CCOMP)

Sep size: Distance between two consecutive measuring positions.
Value range (linear axis):
0.1 to 214748364.7 µm
Value range (rotary/endless axes):
0.1 through $359999.9*10^{-3}$ degrees

For axis positions between two consecutive measuring positions, the control unit automatically calculates a compensation from the two adjacent compensation values by means of linear interpolation.

Starting point: Lowest coordinate value for which a compensation value is determined.
Range of values (linear axes):
±214748364.7 µm
Value range (rotary/endless axes):
0.0 through $359999.9*10^{-3}$ degrees

If the subsequent measuring process is not started at this position, all measured compensation values will have to be sorted accordingly before they can be entered in the CCOMP file.
If the starting point is not identical with the beginning of the complete traversing range, the PNC applies the compensation values of the starting point (P0 and N0) to range ”A” (cf. figure 1).

End point: Highest coordinate value for which a compensation value is determined.
Range of values the same as ”starting point”, however, ”end point” must be greater than the ”starting point”.
If the end point is not identical with the end of the complete traversing range, the PNC applies the compensation values of the end point (Pn and Nn) to range ”B” (cf. figure 1).

2. Make sure that CCOMP has been deactivated for the axis pair(s) target axis/reference axis.
To deactivate CCOMP, see page 10–14.
3. Make sure that the target and reference axis (axes) has (have) been referenced.
4. Move the reference axis (axes) to the start of the traversing range.
5. Move the target axis to the position setpoint to be measured. It is not necessary to change this position in the measurement process for the current target axis.
6. If you use two reference axes, proceed through items 7. to 8. first for one and then for the other reference axis.
Which reference axis you begin with, is up to you.
7. Determine the **compensation value P_i** (i=absolute number of the measuring position: 0 to n; cf. next figure) for each measuring position in positive traversing direction with the help of an external measuring device.
In this context, you exclusively move the reference axis and measure the resulting actual position of the target axis.

Please note for the target axis:

Compensation value in µm = position setpoint (NC) – actual position (machine)

All compensation values must be determined in the unit of measure "micrometer" with a maximum resolution of 0.1 µm at present and may be in the range of ±3276.7 µm.

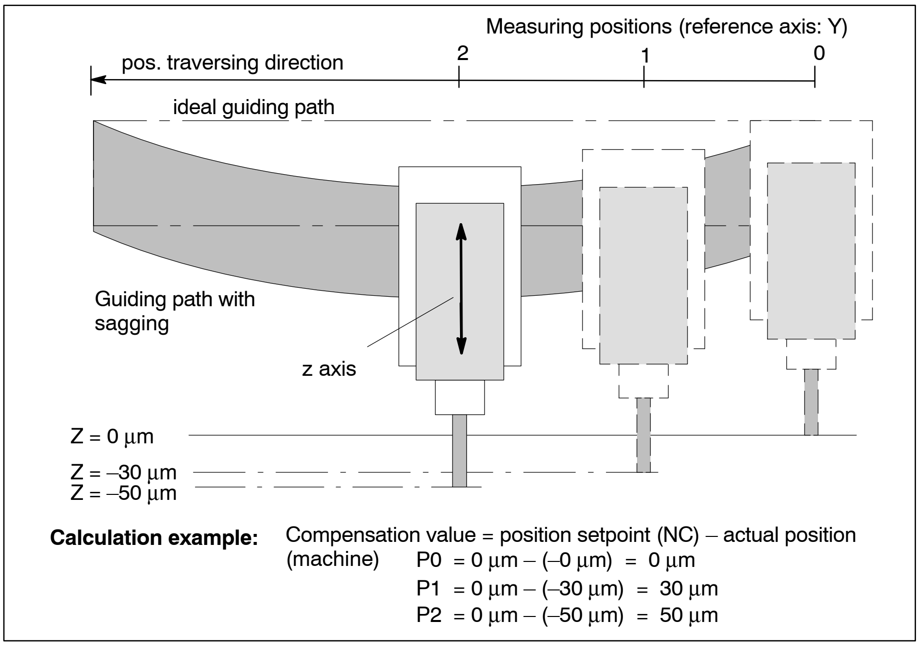


Figure 2: Determination of the compensation values for pos. traversing direction of the reference axis

After having determined the last compensation value in positive traversing direction, move the reference axis to the end of the axis traversing range.
The position of the target axis remains unchanged.

8. Now determine the **compensation value N_i** (i=absolute number of measuring position: n to 0) for each measuring position in negative traversing direction.
   Calculation of the compensation values as under item 7.
   After having determined the last compensation value in negative traversing direction, move the reference axis to the start of the axis traversing range.
   The position of the target axis remains unchanged.

Entering compensation values

The compensation values determined for individual axis pairs (target axis/reference axis) have to be entered in separate ASCII files.
The control unit recognizes via the file name to which pair of axes the file content is to be assigned.

Cross compensation (CCOMP)

Syntax for CCOMP file names: `cnc##r$$.tab`

Meaning:

System axis index of the target axis (01 to 64)

$$ System axis index of the reference axis (01 to 64)

- *$$* and *##* must be different
- System axis index : cf. MP 1003 00004

Examples for CCOMP file names:

`cnc03r01.tab`	contains CCOMP compensation values for target axis 3. Axis 1 is the reference axis.
`cnc03r02.tab`	contains CCOMP compensation values for target axis 3. Axis 2 is the reference axis.

☞ **Max. two CCOMP files can be created for one target axis because the CCOMP function is limited to max. two reference axes per target axis.**

Required structure of CCOMP files:

```
<Starting point> <Step size> <Comment>
<P0> <N0> <Comment>
<P1> <N1> <Comment>
<P2> <N2> <Comment>
<P3> <N3> <Comment>
<P4> <N4> <Comment>
   :
<Pn> <Nn> <Comment>
```

- At least one blank must be programmed between the individual items in one line.
- Finish every line by hitting<enter>.

`<Starting point>`	For more information, please refer to page 10–9.
`<Step size>`	For more information, please refer to page 10–9.
`<Comment>`	Optional character string. Is not evaluated by the control unit when the file is read. For example, you may enter a line number (starting with ”0”) and/or the reference axis position. The string must not exceed the current line because 2 compensation values are always expected at the beginning of a new line.

Cross compensation (CCOMP)

`<Pi>` Compensation value for **positive** traversing direction effective at the axis position `<Starting point>` + i * `<Step size>`. The following applies:
Compensation value = position setpoint (NC) – actual position (machine)
Unit: µm
Resolution: 0.1 µm
Max. range of values: ±3276.7 µm

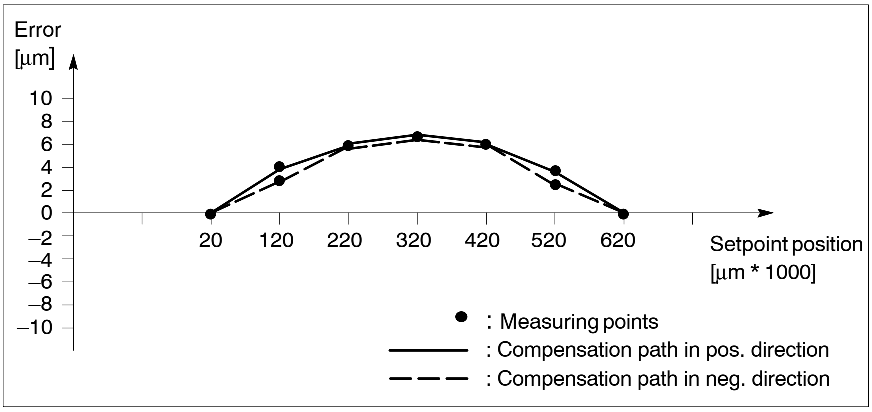
`<Ni>` Compensation value for **negative** traversing direction.
Apart from that, identical to `<Pi>`.

Programming example:

In case of a portal machine, Y axis sagging is to be compensated because it unduly influences the Z axis position.
⇒ target axis (axis to be compensated): Z
⇒ reference axis: Y

The traversing range of the Y axis (0 to 700 mm) was subdivided into a grid with a step size of 100000 (100 mm).
The absolute axis position of 20000 µm (20 mm) was selected as starting point for CCOMP.

Seven axis positions were measured for each the positive and negative traversing direction. The compensation values of the starting point apply to axis positions below 20 mm. The compensation values of the end point apply to axis positions greater than 620 mm.



The target axis Z has system axis index 3, the reference axis Y has system axis index 2.
⇒ Therefore CCOMP data must be entered in file ”`cnc03r02.tab`”.

Required file contents of ”cnc03r02.tab”:

```
20000.0  100000.0  (starting point. 20mm; step size.
                    100mm; 7 measurements)
0.0  0.0  (position 20mm)
4.0  3.0  (position 120mm)
6.0  6.0  (position 220mm)
6.5  6.5  (position 320mm)
6.0  6.0  (position 420mm)
4.0  3.0  (position 520mm)
0.0  0.0  (position 620mm)
```

10.2.2 Activating

1. Copy or load all necessary CCOMP files into the ”root directory” (”/”) or the ”User FEPROM” (”/usrfep”) subdirectory (for directory structure, refer to the ”Operating Instructions” manual).

CAUTION
Danger of damage to the machine or the workpiece!
Do not initiate a control reset during machining.

2. Initiate a control reset.

Compensation values of a target axis will not be activated if

- there is no CCOMP file for an axis or
- an existing CCOMP file contains values that are not allowed.

Possible error messages if a CCOMP file contains values that are not allowed:

`CCOMP: Error when reading the line` <*n*>`, file` <*m*> `!`

or

`CCOMP: One of the numbers in line` <*n*>`, file` <*m*> `is too high !`

or

`CCOMP: Maximum number of CCOMP values in line` <*n*>`, file` <*m*> `is exceeded !`

Meaning:

<*n*> Line number (1,2,3...)
<*m*> Name of incorrect CCOMP file

☞ **If more than two CCOMP files exist for one and the same target axis, the control unit will only take into account the two files it encounters first during start-up.**

☞ **If the CCOMP and LSEC functions are active simultaneously for a target axis, the respective compensation values will take effect additively.**

10.2.3 Deactivating

1. Rename the relevant CCOMP files existing in the ”root directory” (”`/`”) or the ”User FEPROM” (”`/usrfep`”) subdirectory.
   We recommend that you only change the file name extension ”tab” to ”off”, for example (e.g. `cnc03r01.tab` ⇒ `cnc03r01.off`).

CAUTION
Danger of damage to the machine or the workpiece!
Do not initiate a control reset during machining.

---

2. Initiate a control reset.

The respective CCOMP compensation values will no longer be active after start-up.

11 Communication with the PLC

11.1 Auxiliary functions

The PNC provides different auxiliary functions (auxiliary function types) which can be used to create own auxiliary functions.

These auxiliary functions can then be used both in the part program as well as in the ”manual” mode to exchange information with the PLC by means of markers (e.g. to switch functions on the machine on, off or over or to transfer or receive numerical data values).
In case of auxiliary functions with compulsory acknowledgment, the control unit first waits for a positive acknowledgment (e.g. by reset of markers by the PLC), before it executes the next program block or changes over into READY state.

The table below provides an overview of the types of auxiliary functions available and their properties:

Properties of the types of auxiliary functions available							
Auxiliary function type	One-dimensional array with ”channel-dependent bit aux.fct.”	”discrete bit aux.fct.”	”individually defined bit aux.fct.”	”32-bit BCD aux. function”		”64-bit BCD aux. function”	
Use	Exchange of **binary** information with the PLC (e.g. switching functions on, off, over)			Exchange of **numerical** information with the PLC in **BCD format** (transfer of values to the PLC, e.g. tool numbers)			
Coding	bit-coded			BCD-coded			
Memory required	dependent on the array size; per ”channel-dependent bit aux.fct.”: 1 bit	1 bit	1 bit	64 bit (Bit 0: acknowledge/change Bit 1: sign Bit 2–31: reserved Bit 32–63: 32-bit BCD)		96 bit (Bit 0: acknowledge/change Bit 1: sign Bit 2–31: reserved Bit 32–63: 32-bit BCD Bit 64–95: 32-bit BCD)	
Number	defined via 3010 00010 [2]. A total maximum of 1536 bit-coded auxiliary functions per channel (192 bytes).			defined via 3010 00010 [5]	defined via 3010 00010 [3]	defined via 3010 00010 [6]	defined via 3010 00010 [4]
Type	channel-dependent	channel-independent	channel-dependent	channel-dependent	channel-independent	channel-dependent	channel-independent
Existence	1 x per channel	1 x in the system (on precisely **one** channel)	1 x per channel	1 x per channel	1 x in the system	1 x per channel	1 x in the system
Access to auxiliary functions	from the respective channel. With restrictions, also possible from other channels.	from all channels	from the respective channel. With restrictions, also possible from other channels.	from the respective channel. With restrictions, also possible from other channels.	from all channels	from the respective channel. With restrictions, also possible from other channels.	from all channels

For definition of all auxiliary functions in the PNC, use the configuration parameter group 3010. A max. of 30 separate auxiliary functions can be parametrized, each auxiliary functions having to be assigned to a certain auxiliary function type.

Restrictions:

- Marker ranges of the individual types of auxiliary functions may not overlap.
- The M0 through M319 M functions are reserved for internal purposes.

Relevant MACODA parameters (MP):

3010 00010: Number of auxiliary functions.
3010 00020: Acknowledgment requirement for bit-coded auxiliary functions.
3010 00030: Names of auxiliary functions.
3010 00040: Classes of auxiliary functions.
3010 00050: Dependencies of bit-coded auxiliary functions.
3010 00060: BCD format of integer positions, starting bit addresses or bit addresses for individually defined auxiliary functions.
3010 00070: BCD format of decimal positions, end bit addresses or channel numbers.
3010 00080: BCD index, starting code of bit-coded auxiliary functions.
3010 00090: BCD acknowledgment requirement or end code of bit-coded auxiliary functions.
2060 00002: Operand type.
(Individual parameters relevant for auxiliary functions: [5], [6], [7])
2060 00003: Operand address (general).
(Individual parameters relevant for channel-independent 32-bit and 64-bit BCD aux. functions: [6], [7])
2060 00009: Addresses of channel-specific bit-coded auxiliary functions.
2060 00010: Addresses of channel-specific 32-bit BCD-coded auxiliary functions.
2060 00011: Addresses of channel-specific 64-bit BCD-coded auxiliary functions.

11.1.1 Assigning

☞ **For detailed information, please refer to the manual ”Configuration parameters and MACODA, parameter description”. An example for parametrization is included.**

Use the table below to see which configuration parameters are used to parametrize an auxiliary function type.

Auxiliary functions

Overview for parametrization of the auxiliary functions available							
Auxiliary function type	One-dimensional array with ”channel-dependent bit aux.fct.”	”discrete bit aux.fct.”	”individually defined bit aux.fct.”	”32-bit BCD aux. function”		”64-bit BCD aux. function”	
Type	channel-dependent	channel-independent	channel-dependent	channel-dependent	channel-independent	channel-dependent	channel-independent
Existence of the aux. function type	1 x per channel	1 x in the system (on precisely **one** channel)	1 x per channel	1 x per channel	1 x in the system	1 x per channel	1 x in the system
Coding	bit-coded			BCD-coded			
Auxiliary function name	301000030 (max. 30 names)						
Usable number of all defined aux. fct. names	301000010 [1]						
Auxiliary function class	301000040 = 1	301000040 = 1	301000040 = 1	301000040 = 4	301000040 = 2	301000040 = 5	301000040 = 3
Number of all aux. functions of one class	301000010 [2]. Max. total of 1536 bit-coded auxiliary functions per channel			301000010 [5]	301000010 [3]	301000010 [6]	301000010 [4]
Bit auxiliary function type	301000050 = 1	301000050 = 3	301000050 = 4	301000050 = 0			
Digits before decimal point	–			301000060			
Digits after decimal point	–			301000070			
BCD index	–			301000080			
Compulsory acknowledgment	301000020	301000020	301000020	301000090			
Start address of the aux. fct.	301000060	301000060	301000060	–			
End address of the aux. fct.	301000070	–		–			
Channel (for ”discrete aux. functions”)	–	301000070	–	–			
Starting code	301000080	–		–			
End code	301000090	–		–			

Auxiliary functions

Start address of the aux. function range in the PLC	206000009. The address ranges for the individual channels have to be chosen so that they do not overlap. Minimum range size: 301000010 [2] !	206000010. Order of the aux. fct. acc. to BCD index (301000080). Minimum range size: 64-bit x 301000010 [5]	206000003 [6]. Order of the aux. fct. acc. to BCD index (301000080). Minimum range size: 64-bit x 301000010 [3]	206000011. Order of the aux. fct. acc. to BCD index (301000080). Minimum range size: 96-bit x 301000010 [6]	206000003 [7]. Order of the aux. fct. acc. to BCD index (301000080). Minimum range size: 64-bit x 301000010 [4]

☞ **If the number of auxiliary functions is to be dependent on the range of functions of individual options of a machine, define the auxiliary functions – relative to the index of the individual parameters of 3010 00030 – needed for the minimum range of functions before those auxiliary functions intended to expand your machine's range of functions.**
This way you can first parametrize the auxiliary functions according to the maximum options and, if required, simply use 3010 00010 individual parameter [1] to suppress auxiliary functions that have not been released (see page 11–6).

1. Define the NC syntax of the desired auxiliary function by means of 3010 00030.

2. Specify the auxiliary function class in 3010 00040:
   1: bit-coded auxiliary function
   2: channel-independent 32-bit BCD aux. function
   3: channel-independent 64-bit BCD aux. function
   4: channel-dependent 32-bit BCD aux. function
   5: channel-dependent 64-bit BCD aux. function

3. For BCD-coded auxiliary functions only:
   – Define the number of integer positions by means of 3010 00060
     (Value range 32-bit BCD aux. fct.: 0 to 8
     64-bit BCD aux. fct.: 0 to 16)
     and the number of decimal positions by means of 3010 00070
     (Value range 32-bit BCD aux. fct.: depending on integer position
     64-bit BCD aux. fct.: depending on integer position)
   – Specify the index of the BCD auxiliary function using 3010 00080. You thereby influence the order in which several auxiliary functions are stored in the marker field within one function class.
     Value range: 0, 1, 2, ..., n; where n depends on the number of existing BCD auxiliary functions in the respective class. See also item 13.

Example:

Name and index of 32-bit BCD aux. functions used	Order of the aux. functions in the marker field
STEVE (index: 2)	SARAH
SARAH (index: 0)	64 bits free
	STEVE

Please note: STEVE can only be assigned index 2 if at least 3 BCD auxiliary functions of this class have been defined in 3010 00010!

– Define compulsory acknowledgment via 3010 00090:
  0: no compulsory acknowledgment
  1: compulsory acknowledgment
– Make sure that the value “0” has been assigned to 3010 00050.

4. If it is a BCD-coded auxiliary function, the definition is complete now. If it is a bit aux. fct., continue with 5.

5. For BCD-coded auxiliary functions only:
   Enter in 3010 00050 whether the bit aux. fct. is defined channel-dependent, discrete or individually:
   1: channel-dependent
   3: discrete
   4: individually defined

6. For ”discrete” auxiliary functions only:
   – Enter the channel (0 to 12) in 3010 00070 in whose marker range the auxiliary function bit is to be located.

7. For ”discrete” and ”individually defined” auxiliary functions only:
   – Enter the address of the auxiliary function bit in 3010 00060.

8. For ”channel-dependent” bit aux. fct. (bit field) only:
   – Enter the address of the first bit in 3010 00060 and the address of the last bit of the auxiliary function in 3010 00070.
   – Enter the programming code of the first bit in 3010 00080 and the programming code of the last bit of the auxiliary function in 3010 00090.

   Example: channel-dependent bit aux. fct. with the name “RUDY”
   Address of the first bit: 301
   Address of the last bit: 320
   Programming code of the first bit: 1
   Programming code of the last bit: 20
   ⇒ Programming “RUDY1” sets bit 301 in the marker field.
   ⇒ Programming “RUDY20” sets bit 320 in the marker field.

9. Define compulsory acknowledgment via 3010 00020.

10. The definition of the auxiliary function type bit aux. fct. is completed.

11. Repeat steps 1. through 10. if you want to define further auxiliary functions. If all required auxiliary functions have been defined, continue with 12.

12. Adapt individual parameters [1] in MP 3010 00010.
    If you do not wish to suppress auxiliary functions, you have to enter the number of all auxiliary functions whose name is specified in 3010 00030.

13. Adapt the corresponding individual parameter ([2] to [6]) in 3010 00010, depending on the auxiliary function class.
    The sum of all bit-coded auxiliary functions per channel (individual parameter[2]) or the number of the respective BCD-coded auxiliary functions (individual parameters [3] to [6]) has to be entered there.

☞ **Changes in the configuration parameters 3010 00020 and 3010 00040 through 3010 00090 are accepted by system control reset (DIAGNOSTICS ▶ RESET FUNCTIONS ▶ SYSTEM CONTROL RESET).**
Changes in 3010 00010 and 3010 00030 are only accepted after a control unit start-up.

CAUTION
Danger of damage to the machine or the workpiece!
Do not initiate a control reset during machining.

---

14. Initiate a control reset.

Programming the respective auxiliary function now leads to the corresponding marker bit being set.
The PLC program is responsible for evaluating the markers, for locking, if required, or for resetting markers (acknowledgment).

11.1.2 Activating

Parametrized auxiliary functions can be released for use or suppressed via MP 3010 00010 [1]. This way, you can, e.g. completely parametrize all auxiliary functions available within a machine series and then only release those for use which are relevant for a certain machine type / option.

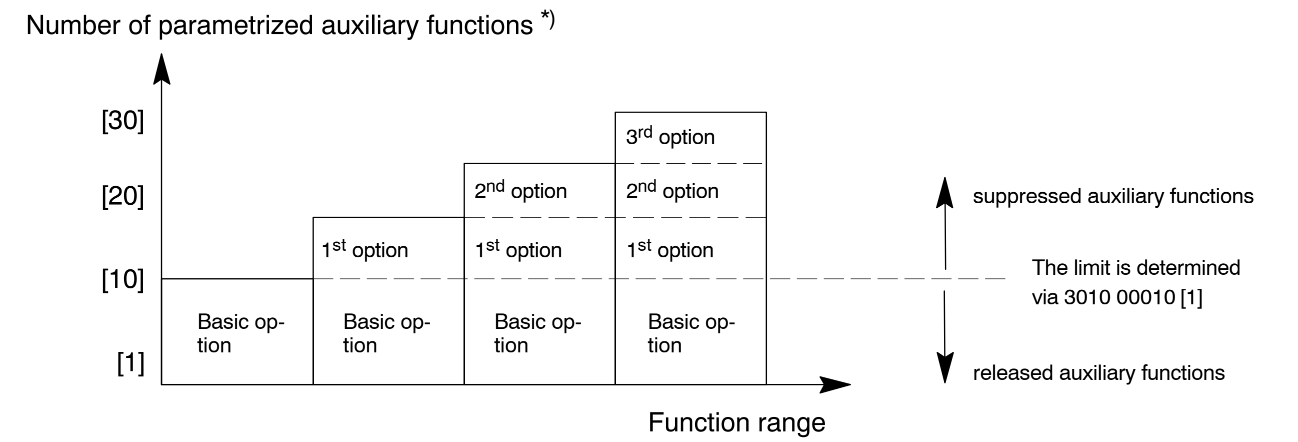
Auxiliary functions

Example 1:

3010 00010	3010 00030		Auxiliary function is valid
[1] 3	[1]	M	yes
	[2]	M500	yes
	[3]	M999	yes
	[4]	O	no
	[5]	S?	no

Example 2:

By increasing the value in 3010 00010 [1], a further auxiliary function is released for use. This way, the function range of the machine regarding auxiliary functions is quickly established:



*) Index of the individual parameters in 3010 00030, which are assigned a value unequal to ”0”.

1. Adapt individual parameters [1] in MP 3010 00010.
   Enter the number of usable parametrized auxiliary functions.

CAUTION
Danger of damage to the machine or the workpiece!
Do not initiate a control RESET during machining.

CAUTION
Danger of damage to the machine or the workpiece!
Do not initiate a control RESET during machining.

2. Initiate a control RESET.

Auxiliary functions

Notes:

12 Technology / machining functions

12.1 Oscillating axis

Function:

Assigns a random synchronous axis as oscillating axis. Oscillating axes perform an independent oscillating movement, whereas interpolation is carried out for the other synchronous axes on the channel.
The function is suitable, e.g. for simple surface grinding applications.

After activation (⇒ G301), the oscillating axis first traverses to the reversing point (⇒ G350) it can reach on the shortest path.
For as long as oscillating is active, the oscillating movement will be steady and can be differentiated across block limits.
If oscillating is deactivated (see Section 12.1.2), the oscillating axis first traverses to the next reversing point it can reach. The subsequent programmed movement will then start.
After control reset, an active oscillating process will only be interrupted upon reaching the next reversing point.

Restrictions:

- Circular interpolation with combined oscillating movement is not possible.
- Feed potentiometer does not act on oscillating axes.
- When oscillating is activated, G4, G14/G15, G32, G75, G114/G115, G374, G590/G591 or G900 must not be programmed.
  Otherwise, the interpolation will be stopped abruptly. This may cause a servo error.
- Oscillating movements are performed according to a cosine function with a sinusoidal speed pattern and cosine-shaped acceleration pattern.
- In the axis display, the last position before the oscillating movement starts is shown as ”end position” and value ”0” as ”program value”.

Relevant G functions:

G350: Initializing the oscillating axis
(axis selection, reversing points, frequency/speed of the oscillating movement).
The data are retained until they are overwritten by G350 or control reset is performed.

G301: Activating the oscillating movement.
Programming the path feed and the target positions for a linear interpolation of the axes involved.
The oscillating movement can also be carried out for a specific period of time, if no traversing movement has been programmed in a G301 block.

12.1.1 Activating

Precondition:

- The axis is already available in the system as a synchronous axis (machining axis) with its speed, dynamics and, if required, its positioning logic correctly parameterized in accordance with your requirements.
- The necessary axis-related interface signals are evaluated or generated, respectively.

1. If active, switch the ”In position” function off for the oscillating axis.
2. Program the required initialization via G350.
3. Activate oscillating by programming G301 accordingly.

12.1.2 Deactivating

1. Deselect G301 by another G function of group 2 (e.g. G1).

☞ **For information on groups, see ”Overview of G instructions” in the annex of the DIN Programming Instructions.**

12.2 Punching and nibbling

Function:

Expands the PNC by various functions which are required for its application in punching or nibbling machines with hydraulic punchers.

Features:

- Program-controlled switching on and off of punching/nibbling by means of G functions G 662 (”Nibbling ON”), G661 (”Punching ON”) and G660 (”Punching/nibbling OFF”). All three functions form a modal group, i.e. at any point in time, one of the functions can be active only.
  For information on groups, see ”Overview of G instructions” in the annex of the DIN Programming Instructions.
- Automatic segmentation of a programmed traversing block into linear path segments with identical length (block splitting). The end points of the path segments are always located on the programmed path.
- Automatic tangential axis setting of the punching/nibbling tool on the programmed path at the block end via function ”Tangential tool orientation” (see page 12–10 ff.).
- Providing a high-speed input and output signal for communication with a stroke-related punching control.
  - PNC-P: plug-in card “PNC Highspeed I/O”
  - PNC-R: module osa dc I/O.

  The signal wave between NC and punching control can be controlled by the NC using a stroke time monitoring system.
- The NC’s internal high-speed logic for exchange of information between punching control, NC and PLC.
- Stroke request possible via the NC as well as via the PLC. The NC provides different signals at the general interface for communication with the PLC. Among others, the PLC may thus also initiate a stroke inhibit.
- Shifting the stroke release times by the NC (early or delayed stroke release). The time referenced point may be when the interpolator reaches the end point of a traversing movement or all axes reach the Inpos window.

Punching and nibbling

Punching and nibbling differ with respect to the following points:

Function	**for punching**	**for nibbling**
Block splitting	possible if required	required
Stroke release at block end	only required at the end of the programmed traversing block	required at the end of each path segment
Stroke release at the start of a programmed traversing block	not required	only required, if the previous block was not a traversing block, or G60 was active in the previous block (”Punching/nibbling OFF”)

In case of active punching or nibbling, a stroke is always released at the block end, as a rule. Exceptions:

- The current block does not have an axis coordinate in the active plane. Machining is continued without stroke with the next block.
- stroke release is suppressed by the PLC. In this case, machining is only continued carrying out the stroke after release by the PLC.

Restrictions:

- Nibbling and punching is not suited for machine tools with a free-running puncher
- Nibbling and punching can only be switched on if MP 8001 00010 (”Activation of punching function” has been assigned the value ”1”
- Nibbling and punching can only be used on one channel at any given point in time.
- Path shape (G408 or G608) must be active
- PNC-P: plug-in card “PNC Highspeed I/O” required
  PNC-R: module osa dc I/O required
- One high-speed digital input and output is necessary to communicate with the punching control

Relevant MACODA parameters (MP):

8001 00010: Activation of punching function.
8001 00020: Axis-specific stroke release times.
8001 00021: Reference of stroke release times.
8001 00030: Stroke check (monitoring) time

Relevant G functions:

G660: (programming alternative: SPOF)
Punching or nibbling OFF.

G661: (programming alternative: PON)
Punching ON.

G662: (programming alternative: SON)
Nibbling ON. Block splitting has to be activated for nibbling (”LEN=” > 0).

G612: Determination of axis-specific stroke release times in relation to the time referenced point ”Inpos window reached”. Early stroke release is not possible here.

G611: Determination of axis-specific stroke release times in relation to the time referenced point ”Interpolator end point reached”. Both early stroke release as well as delayed stroke release possible.

G610: Set stroke release times and time referenced points of all axes to the MACODA values (MP 8001 00020 and MP 8001 00021).

Relevant interface signals:

HSO-0: High-speed digitaloutput
(can be set via MP 4075 00102).
NC demands stroke release.

HSO-0: High-speed digitalinput
(can be set via MP 4075 00101).
Punching control acknowledged.

NC-O5.0: ”Stroke intended” (global interface).
The NC indicates to the PLC that it wants to release a stroke.

NC-O5.1: ”Stroke is not running” (global interface).
Image of HSI-0 for the PLC.

NC-I1.0: ”Stroke inhibit” (global interface).
PLC prevents the setting of HSO-0.

NC-I1.1: ”Stroke reservation” (global interface).
PLC reserves the HSO-0 for own stroke release.

NC-I1.2: ”Stroke ON” (global interface).
The PLC instructs the NC to release a stroke.

12.2.1 Assigning

1. Set MP 8001 00010 (activation of punching and nibbling) to value ”1”.
2. Set time referenced point (MP 8001 00021) and stroke release times (MP 8001 00020) in accordance with your application.
3. If desired, parametrize stroke monitoring (MP 8001 00030).
4. Reserve a high-speed digital input and output via MP 4075 00101 and MP 4075 00102.

5. In order to determine the function status during start-up and after control reset, we recommend the following settings in MP 7060 00010 (start-up) and in MP 7060 00020 (control reset) as default status:

G660	Nibbling/punching OFF
G610	Stroke release times and time referenced point according to MP 8001 00020 and MP 8001 00021
G630	Tangential tool orientation OFF.

CAUTION
Danger of damage to the machine or the workpiece!
Do not initiate a control reset during machining.

6. Initiate a control reset.

12.2.2 Activating

Precondition:

- Punching/nibbling is assigned
- High-speed digital input and output has been reserved via MP 4075 00101 or MP 4075 00102.
- Path shape (G408 or G608) is active
- The necessary interface signals are evaluated or generated, respectively.

1. Optional for punching, required for nibbling:
   Specify block segment length via ”LEN=”.
2. To activate punching, program G661
   – or –
   program G662 to activate nibbling.

12.2.3 Deactivating

1. Program G660.

12.3 Tangential tool guidance

Function:

Moves a tool axis (rotary or endless axis) along tangentially to the programmed path in the active plane of a channel.

Features:

- Parametrizable offset angle (angle between symmetry axis of the tool and the path tangent).
- Taking into account the tool symmetry. This allows the control unit to optimize the required offset rotation of the tool in case of knees in the contour automatically (”shortest path” logic).
- Automatic insertion of intermediate blocks for rotating the tool in case the knee angle between two contour elements exceeds a parametrizable value.
- Communication with the PLC, if automatic insertion of an intermediate block is necessary due to a knee in the contour. This way, the PLC can execute any necessary steps (e.g. lift the tool) prior to executing the intermediate block, before the NC rotates the tool.
- Complete parametrization of the ”Tangential tool guidance” via G131 in the part program alternatively, or by means of the MACODA parameters.
- Read access to active parameters of G131 possible via SD function.

Restrictions:

- The ”Tangential tool orientation” function (for nibbling/punching; see page 12–10 ff.) must not be active at the same time
- Only rotary and endless axes are permitted as tool axes
- G131 must never be programmed in a block with a traversing movement.
- Automatic insertion of an intermediate block is not possible if a single WAIT block has been programmed between 2 traversing blocks with active ”Tangential tool guidance”.
- Tangent angles at start and end of the programmed contour section must be located within the travel limit range of the tool axis (if the tool axis is a rotary axis).

Relevant MACODA parameters (MP):

The MACODA parameter values act as initialization values if G131 has been programmed without parameters. You can overwrite the initialization values in the G131 block by the parameters indicated in brackets.

7050 00210:	Number of the axis of rotation	(TAX=)
7050 00220:	Tool symmetry	(SYM=)
7050 00230:	Adaptation angle	(IA=)
7050 00240:	Offset angle	(ANG=)
7050 00250:	Init mode of the offset angle	–
7050 00260:	NC-PLC communication in respect of intermediate blocks	(PLC=)

Relevant G functions:

G131: Initializing and activating the ”Tangential tool guidance”. G131 does not lead to a traversing movement. The tool axis is only offset with the next traversing block. Depending on the active adaptation angle, the tools axis rotates either within an automatically generated intermediate block or jumps to the offset position.

G130: Deactivating the ”Tangential tool guidance”.

Relevant interface signals:

NC-O4.0: ”G131 tool rotation” (channel interface).
High signal indicates to the PLC that the knee angle between two contour elements exceeds the currently set adaptation angle.
You determine via MP 7050 00260 or using G131 parameter ”PLC=” whether the NC should wait for the release by the PLC before it executes the intermediate block.

NC-I3.2: ”G131 tool rotation release” (channel interface).
High signal releases the execution of the intermediate block by the NC, if MP 7050 00260 or G131 parameter ”PLC=” has been assigned the value ”1”.
Any blocks following the intermediate block will only be executed by the control unit when the PLC resets this signal.

Relevant SD functions:

SD(131,1) provides the logical axis number of the axis of rotation

SD(131,2) provides the tool symmetry

SD(131,3) provides the offset angle in degrees

SD(131,4) provides the adaptation angle in degrees

12.3.1 Activating

Precondition:

- The axis of rotation is already available in the system as a rotary or an endless axis with its speed, dynamics and, if required, its positioning logic correctly parameterized in accordance with its operational requirements.
- MP 7050 00250 (init mode of the offset angle) is parameterized in accordance with your requirements.
  We recommend that all other MACODA parameters which are relevant for the ”Tangential tool guidance” be set to appropriate values. This way you make sure that the function is initialized correctly even if not all parameters available have been programmed in the G131 block.
  If you want to activate the ”Tangential tool guidance” via G131 block without parameters, all MACODA parameters need to be set to appropriate values!

- The necessary axis- and channel-related interface signals are evaluated or generated.

1. If it is active, deactivate the ”Tangential tool orientation” function for the respective axis (G630).
2. Activate ”Tangential tool guidance” via G131 block. If required, adjust the initialization values via G131 parameters.

12.3.2 Deactivating

1. Program G130.

12.4 Tangential tool orientation

Function:

Offsets a synchronous tool axis, type ”endless” movement (⇒ MP 1003 00004) tangentially to the programmed path in the active plane of a channel for each stroke, in case of active ”punching” (G661) or ”nibbling” (G662).

The tool axis moves synchronously with the linear axes defining the plane arriving at the offset angle at the same time they reach their target position. The term ”target position” refers both to the end position of the programmed block and the end position of block segments which the NC can generate in connection with the automatic block splitting.
If a stroke has to be released at the start of the programmed block already (may be required for nibbling), the NC automatically generates a separate block in which the tool axis is first rotated to the required offset angle.

Features:

- Parametrizable offset angle (angle between symmetry axis of the tool and the path tangent).
- Taking into account the tool symmetry. This allows the control unit to optimize the offset rotation of the tool required for a stroke in case of knees in the contour automatically (”shortest path” logic).
- Offset angle and tool symmetry are parametrizable in the program

Restrictions:

- The ”Tangential tool guidance” function (see page 12–7 ff.) must not be active at the same time
- The ”Tangential tool orientation” can only be programmed while G662 (”Nibbling ON”), G661 (”Punching ON”) or G660 (”Punching/ nibbling OFF”) are active.
- G631 must be programmed in a separate block.
- The tool axis must be defined via MP 7050 00210. A program-controlled change of the tool axis is not possible.
- Only synchronous endless axes are permitted as tool axes

Relevant MACODA parameters (MP):

7050 00210: Number of the axis of rotation

Relevant G functions:

G631: (programming alternative: TTON)
Initializing and activating the ”Tangential tool orientation”. G631 does not lead to a traversing movement. The offset of the tool axis only starts with the next punching or nibbling traversing block, if it is not necessary to release a stroke at the block start (cf. section ”Function”).

G630: (programming alternative: TTOFF)
Deactivating the ”Tangential tool orientation”

12.4.1 Activating

Precondition:

- The axis of rotation is already available in the system as a synchronous endless axis with its speed, dynamics and, if required, its positioning logic correctly parameterized in accordance with your operational requirements.
- MP 7050 00210 (number of axis of rotation) is parametrized in accordance with your requirements.
- The necessary axis- and channel-related interface signals are evaluated or generated.

1. If active, deactivate the "Tangential tool guidance" function for the respective axis (G130).
2. Activate ("Nibbling ON"), G661 ("Punching ON") or G660 ("Punching/nibbling OFF").
3. Activate "Tangential tool orientation" via G631 block or TTON command. Set the initialization values of tool symmetry and offset angle via "SYM=" or "ANG=" parameter, depending on your application, in one and the same block.

☞ **The "Tangential tool orientation" can also be parametrized and activated in case of active G660 ("Punching/nibbling OFF").**
Offset movements of the tool axis, however, are only carried out in case of active G661 or G662.

12.4.2 Deactivating

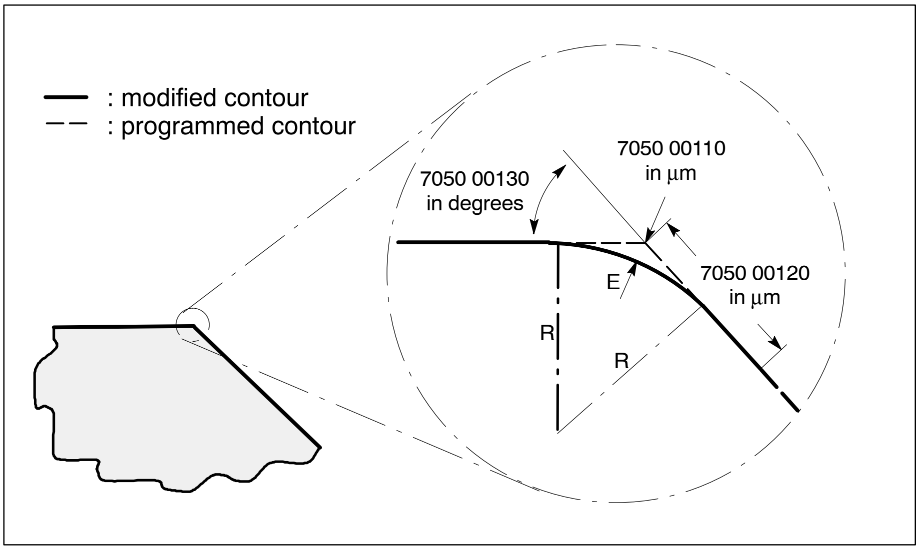
1. Program G630.

12.5 Rounding of corners

Function:

The control unit can use G34/G134 to insert tangential transition arcs between linear or circular/helical blocks in the active plane.
This results in a slight modification of the programmed contour at the block transition, on the one hand, but on the other, it is possible to achieve continuous speed and acceleration patterns (if the radius is sufficiently large) on the path.

Configuration parameters 7050 00120 and 7050 00130 are used to determine under which conditions the control unit inserts transition arcs.



Restrictions:

- G34 can only be used in connection with two linear blocks, whereas G134 acts on circular blocks and helical blocks additionally.
- Both neighboring blocks have to be located on the selected main plane. In the case of helical blocks, only the components of the circular plane for rounding are taken into account.

Relevant MACODA parameters (MP):

7050 00110: Max. admissible deviation from programmed contour. Based on this, the control unit automatically calculates the matching radius of the transition arc.

7050 00120: Minimum block length for corner rounding function. If one of the neighboring traversing blocks is smaller, no transition arc is generated.

7050 00130: Maximum angle for corner rounding function (up to which block transitions are considered to be quasi-continuous).
No transition arcs are generated in case of larger angles.

Rounding of corners

Relevant G functions:

G34: Switches on ”Corner rounding” between 2 linear blocks. Using ”E” address, the maximum admissible deviation between modified and programmed contour (presetting via MP 7050 00110) can be influenced temporarily (see G36).

G35: Switches off ”Corner rounding”.

G36: Deletes a maximum admissible deviation programmed via E address. The value from MP 7050 00110 is effective again.

G134: Switches on ”Corner rounding” between linear or circular blocks. The desired radius of the transition arc has to be programmed via ”R” address when G134 is first called. It is effective until it is overwritten by reprogramming using G134.

☞ **The G34, G35 and G134 G functions form a group and therefore deselect each other mutually.**

12.5.1 Assigning

1. Use MP 7050 00110 to determine the presetting for the maximum admissible deviation at the block transition between the modified and the programmed contour.
   This data is only relevant when G34 is used.
2. Use configuration parameters 7050 00120 and 7050 00130 to specify under which conditions it is permitted to insert transition arcs.

CAUTION
Danger of damage to the machine or the workpiece!
Do not initiate a system control reset during machining.

---

3. Initiate system control reset
   (DIAGNOSTICS ▶ RESET FUNCTIONS ▶ SYSTEM CONTROL RESET).

12.5.2 Activating

1. Program G34 or G134, depending on the function that you want.

12.5.3 Deactivating

1. Program G35.

12.6 Switching NC blocks via high-speed signal

Function:

Using the ”Switching NC blocks via high-speed signal” function (G575), it is possible to initiate an ”early” block change in the PNC via high-speed input.
This way, e.g. external control/monitoring devices can cancel an active linear traversing movement. The control unit subsequently continues machining with the next programmed block.

The function permits

- ”on-the-fly” change of blocks:
  the programmed end points of axes are not deleted.
  If these axes are not programmed with new end points in the next block, the control unit takes over the end points from the cancelled block and interpolates these axes together with the other axes.
  The current speed in the cancelled block is not influenced at the block change (see exceptions under ”Restrictions”). Any necessary change in speed will be initiated by the NC in the subsequent block only, however without checking the maximum velocity step change.
- Block change with ”Cancel distance to go”.
  The programmed end points of axes are deleted.
  If these axes are not selectively programmed again in the subsequent block, they will not be traversed anymore.
  The current path speed of the cancelled block is always reduced to v=0 at the block change. In this context, a downslope with the current deceleration as well as a speed jump to v=0 can be selected via HSSTOP parameter.

You can use ”on-the-fly” block change e.g. if you intend to traverse along a straight path at different feedrates in dependence on an external event.
Block change with ”Cancel distance to go” is suitable e.g. for applications requiring pressure or torque-driven ”Traversing to dead stop”.
For examples, see G 575 in the ”DIN Programming Instructions”.

CAUTION
The function changes the programmed contour!
This may cause damage to the workpiece and/or the tool!
Only use this function if the tool is not in contact with the workpiece or damage to workpiece and/or tool can be excluded!

Restrictions:

- A linear traversing movement has to be programmed both in the cancelled block as well as in the subsequent block (with active G0, G1, G10, G11, G73 or G200).
- PNC-P: plug-in card “PNC Highspeed I/O” required
  PNC-R: module osa dc I/O required
- The desired high-speed input has to be assigned the ”G575” functionality in MP 4075 00101.

Switching NC blocks via high-speed signal

- ”On-the-fly” block change within the automatic operation mode is performed at axis standstill if
  - there is a knee > 90 degrees in the contour between the block to be cancelled and the next block.
  - the G575 block would always end with v=0 due to the currently active type of interpolation (in case of active G0, G10, G73)
  - the ”In position” function is active (G61, G161, G163)
  - the next block begins with v=0 due to additional information programmed (G14, G15, G114, G115)
  - the ”Shape” function is active (see page 6–7).
- In the ”Single block” and ”Single step” operating modes, decelerating to v=0 is performed in spite of programmed ”on-the-fly” block change because the control unit executes only one single program block per cycle start.
- In the ”Program block” and ”Manual input” modes, decelerating to v=0 is performed in spite of programmed ”on-the-fly” block change **and the distance to go is cancelled!** Any next NC block with incremental programming will therefore not lead to the previously programmed end point.

Relevant G functions:

G575: Activating ”Switching NC blocks via high-speed signal” for the duration of the G575 block.
The desired conditions for initiation and the selection of the digital input can be set via the HS parameters:
HSx=0: low level (x: number of the input)
HSx=1: high level.
The HSSTOP parameter influences the effects of the function:

without HSSTOP:	”on-the-fly” change of blocks. No change of speed at the block transition.
HSSTOP=0:	block change with ”Cancel distance to go” and downslope to v=0.
HSSTOP=–1:	block change with ”Cancel distance to go” and velocity jump to v=0.

If the level of the high-speed input for the G575 block does not change during the machining time, the block is executed to the end position in accordance with its programming (with or without HSSTOP parameter). Then the PNC executes the next program block.

Relevant interface signals:

HSO-0: High-speed digitalinput
(can be set via MP 4075 00101).
Initiates change of blocks.
The input is requested in the interpolation cycle.

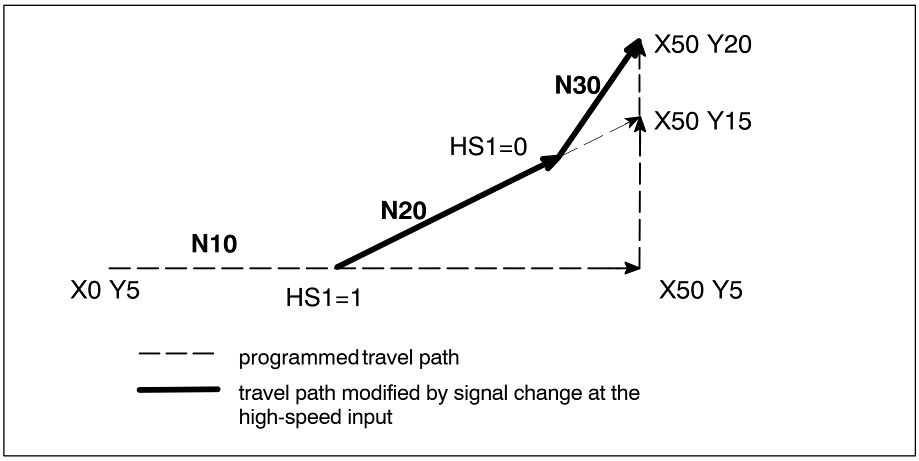
Switching NC blocks via high-speed signal

Relevant MACODA parameters (MP):

4075 00101: Assignment of digital inputs.
Determines which high-speed digital input is assigned to the ”G575” function, among others.

Example: “on-the-fly” change of blocks.
The resulting traversing movement to the end point (X50/Y20) may change in dependence on the signal change at the high-speed input, but the end point itself remains the same.
The programmed speed is maintained during the change of blocks.

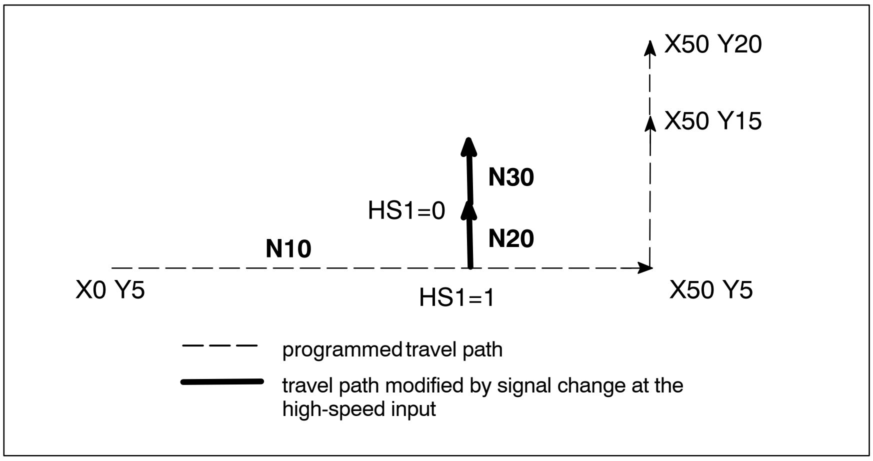
N05 G1 F1000 X0 Y5	End point: X0 Y5
N10 G575 HS1=1 **G90** X50	End point: X50 Y5
N20 G575 HS1=0 **G91** Y10	End point: X50 Y15 (abs. Y5+incr. Y10)
N30 **G91** Y5	End point: X50 Y20 (abs. Y15+incr. Y5)



Example: block change with ”Cancel distance to go”.
The programmed end point (X50/Y20) is only maintained if there is no signal change at the high-speed input. In all other instances, the end point is dependent on the signal change at the high-speed input.
Decelerating to v=0 is performed via downslope at block transitions.

N05 G1 F1000 X0 Y5	End point: X0 Y5
N10 **G575** HS1=1 HSSTOP=0 **G90** X50	End point: external event at $X \leqslant 50$
N20 G575 HS1=0 HSSTOP=0 **G91** Y10	End point: external event at $Y \leqslant 10$
N30 **G91** Y5	End point: distance from the last occurrence of external event is incremental Y5

Switching NC blocks via high-speed signal



12.6.1 Assigning

1. Make sure that all high-speed inputs which you want to use in connection with the ”Switching NC blocks via high-speed signal” function are assigned the value ”0003” in MP 4075 00101.

CAUTION
Danger of damage to the machine or the workpiece!
Do not initiate a control reset during machining.

2. If you have changed the configuration parameter, initiate a control reset.

12.6.2 Activating

Precondition:

- A linear type of interpolation is active (G0, G1, G10, G11, G73 or G200).

1. Program G575 in connection with the HS and, if required, the HSSTOP parameters (see ”Relevant G functions”).
   The function is active up to the block change initiated, but no longer than the duration of the G575 block and is then switched off automatically, unless a new G575 blocks follows right away.

12.7 Tapping

Tapping can be performed in two ways:

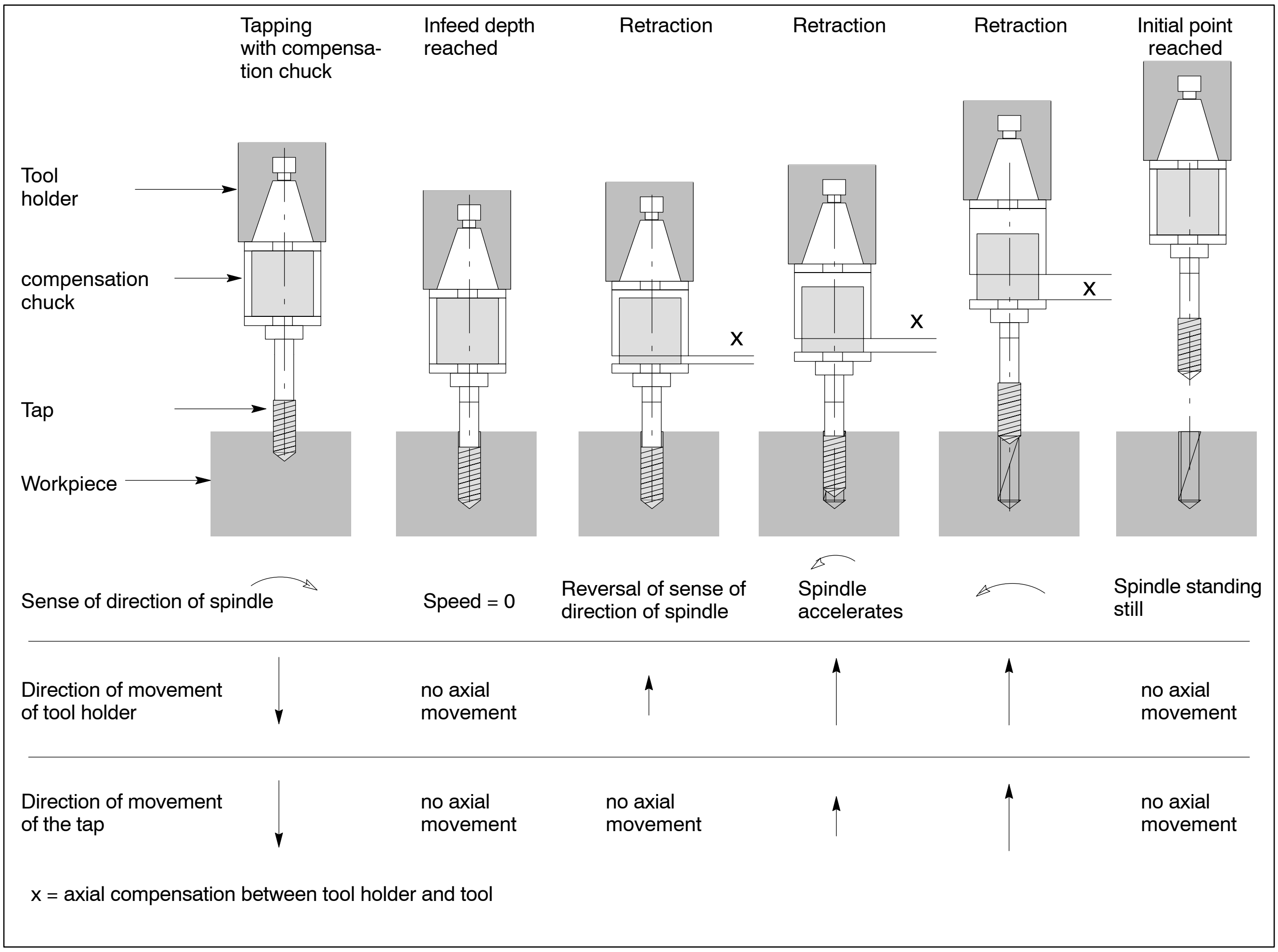
- with compensation chuck
- without compensation chuck.

Tapping with compensation chuck

Function:

As there is no direct (interpolary) synchronization of drill axis feedrate and spindle rotation when “Tapping with compensation chuck”, there is an unintended **offset** between the drill axis position and the spindle position (one axis ”lags” behind the other axis).
This offset (limited in time and size) is compensated by using a compensation chuck.

In contrast to tapping without compensation chuck, the spindle is in “Speed mode” all the time during the tapping procedure.



Tapping

The “Tapping with compensation chuck” function can be

- programmed with linear interpolation at feedrate, specification of the spindle speed and sense of rotation, or
- with tapping cycle G84 via parameter assignment.

Restrictions:

- G84 is only permitted in automatic mode.
- Switchover of the drill axis via G78 has been assigned correctly.
- Control reset deselects an active cycle; selection via M02 and M30 is only performed if “G80” has been entered in MP 3030 00001.
- G9321/G9322 (retraction from tapped hole) cannot be used in case of ”Tapping with compensation chuck”.

Relevant MACODA parameters (MP):

Definition of the NC syntax (G84) and the subprogram name

309000005: Defines the NC syntax of up to 16 functions which can be used to call **15 modal** subprograms.

309000006: Defines up to 16 names of subprograms which are called in dependence on 3090 00005.

Relevant G functions:

G1: Linear interpolation

G84[Z,...]: Tapping cycle “Tapping **with** compensation chuck”

G80: deselects cycle

Relevant M functions:

M3, M4, M5 programmed spindle turn right or left, spindle stop

---

CAUTION
It is possible that the wrong spindle is addressed!
The syntax for available spindle functionalities are determined in MACODA parameters 1040 00101 ff.
Therefore make sure that the syntax used in the cycle corresponds to the entries in MACODA!

Relevant interface signals:

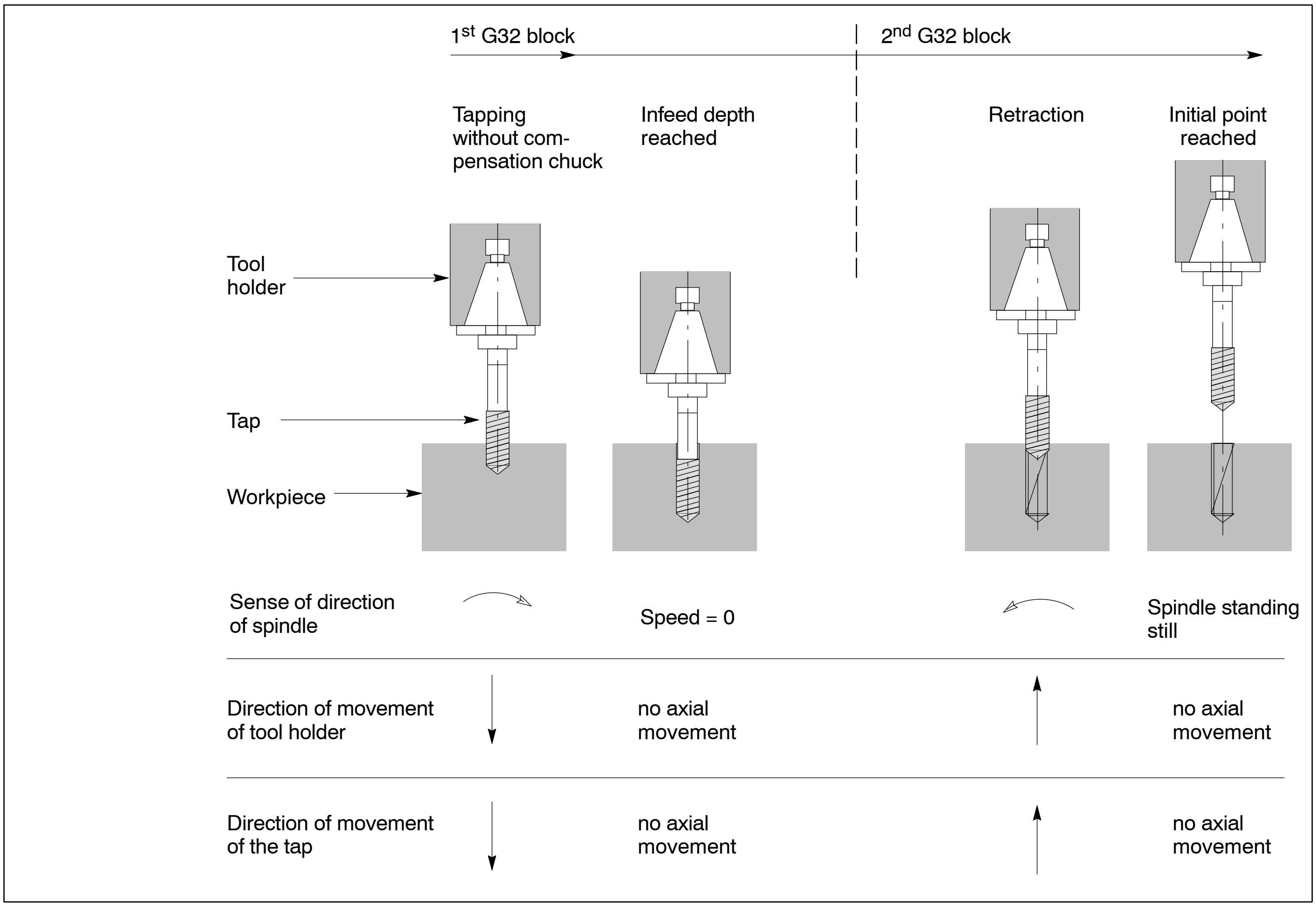
NC-I 7.6 “General drive OFF” has to be blocked for the duration of the cycle, if required.

Tapping without compensation chuck

Function 1: Tapping without compensation chuck

Higher demands regarding thread quality require a precise interaction between drill axis and spindle.

To produce a thread without compensation chuck, the PNC synchronizes the linear infeed movement of the drill axis with the rotation movement of the spindle. **No offset** is produced in the process because the infeed position of the drill axis and the spindle rotation interpolate with each other, i.e. the drill axis and the spindle accelerate/decelerate **at the same time** and ratio.



“Tapping without compensation chuck” always consists of **two** consecutive blocks

- G32 for “drilling” and
- G32 for “retraction” from the tapped hole.

Instead of block-by-block programming, it is also possible to select a tapping cycle (e.g. G184). The program called with the tapping cycle contains the consecutive G32 blocks.

The thread pitch is specified as follows:

- as the ratio between path feed of the drill axis and spindle speed (F/S). Programming: `G32 F.. S..`

Tapping

- **or** by direct input of the thread pitch (H).
  Programming: `G32 H..`
- **or** by input of the thread pitch with G184 (boring (tapping) cycle).
  Programming: `G184[...,GS,...]`

Function 2: Retraction from the tapped hole (without compensation chuck)

It may occur that the tapping process is interrupted. This may be due to various causes:

Cause of the interruption	Tapping data available	Tapping continued	Programming
Feed potentiometer = 0	yes	Yes: set feed potentiometer >0	–
Control reset initiated	yes	No: automatic retraction of the tap	G9321 and G9322 with parameter F
Power failure (PNC startup required)	no	No: manual retraction of the tap	G9321 and G9322 with the parameters S, F, M3/M4, drill axis and incremental retraction path
Breaking of tool	yes	No: automatic retraction of the tap after Control reset	G9321 and G9322 with F

Retraction (automatic and manual) is principally initiated by the G9321 function (switching the spindle to C axis operation). The retraction movement is started with G9322.

The corresponding parameters

- starting position of the 1st tapping block (G32)
- thread pitch (F/S)
- direction of rotation
- spindle involved

are either provided automatically by the control unit (after Control reset) or have to be entered manually (after power failure and subsequent control unit start-up).

G 9321 and G9322 can either be programmed via manual data input or via cycle.

Restrictions for function 1 and 2:

- The spindle has to be appropriate for C axis operation
- Switching the spindle to C axis operation is performed automatically via G32. When G32 is deselected, the spindle is set back to spindle mode.
- Drill axis and spindle (in C axis operation) have to be located in the “in-position range” before tapping is started.

- In case of active G32, controlling the spindle via **machine function**
  - speed S or SSPG,
  - spindle functions M3/M13, M4/M14, M5 or M19,
  - gear ranges M41–M44 or M48

  or via the **interface** is not possible.
  - Spindle manual M3, M4, M5, M19
  - Spindle jog M3, M4
- During tapping, only the feed potentiometer is active.
- Drilling and retraction must always be programmed with identical thread pitch (F/S or H, respectively).
- A block must not be programmed between the NC blocks for “drilling” and “retraction”.
- If manual or automatic retraction (G9321/G9322) is used, axis movement type 2 (rotary axis with modulo calculation) has to be set in MP 1003 00004 for the C axis operation of the spindle.
- The retraction movement can only be carried out **manually** after an interruption with active inclined plane (G352).

Relevant MACODA parameters (MP):

1001 00001:	Drive function type (drive moves a spindle which can be switched to C axis operation temporarily)
1003 00004:	Axis movement type: C axis/spindle is a rotary axis with modulo calculation (endless rotary axis for G9321/G9322). The setting has to be compatible with S-0-0076!
1040 00001:	Selection of spindle type (e.g. SERCOS/analog spindle, ..)
1050 00001:	Specifies the manufacturer for each drive.

Relevant SERCOS parameters:

S-0-0016:	C axis/spindle: configuration list DT. Has to include ”S-0-0051” or ”S-0-0053”.
S-0-0024:	C axis/spindle: configuration list MDT. Has to include ”S-0-0047”.
S-0-0057:	C axis/spindle: positioning window.
S-0-0261:	C axis/spindle: positioning window rough in phase 3 (for Servodyn drives)
S-0-0076:	C axis/spindle: Type of weighting for position data in phase 2 (C axis/spindle as endless rotary axis). The setting has to be compatible with MP 1003 00004!
S-0-0033:	C axis/spindle: secondary mode 1 in phase 2: position mode required.
S-0-0034:	C axis/spindle: secondary mode 2 in phase 2: position mode required.
S-0-0103:	C axis/spindle: modulo value in phase 3

Tapping

Relevant G functions:

G32..	Tapping without compensation chuck (acts block by block) with specification of the drill axis, infeed depth, feedrate, sense of rotation of spindle, speed, thread pitch
G184[Z,...]:	Tapping cycle “Tapping **without** compensation chuck” (active G32 required)
G80:	deselects cycle

Relevant interface signals:

at the channel interface

NC-O 20.2	“G32 active” for the duration of tapping

at the spindle interface

NC-O 14.3	“Spindle command” active for the duration of “C axis switching active”
NC-O17.7:	“C axis switching active
NC-O 17.0	“C axis active” for the duration of tapping

at the axis interface

NC-O 14.3	“Travel command” for the C axis and drill axis for the duration of tapping

12.7.1 Assigning

1. Set telegrams for the C axis. See ”Relevant SERCOS parameters”: S-0-0016, S-0-0024.
2. Set operating modes for the C axis. See ”Relevant SERCOS parameters”: S-0-0033, S-0-0034.
3. Adjust SERCOS drive data of the C axis/spindle and the drive of the drill axis so that the in-position data (S-0-0057, S-0-0261) meet the desired production specifications (thread quality, speed).
4. Set the drive function type of the spindle drive to “3” (spindle can be switched to C axis operation) in MP 1001 00001.
5. Set spindle type to “1” (SERCOS spindle) in MP 1040 00001.
6. The drive type (e.g. 0=Servodyn) has to be entered in MP 1050 00001.

Continue with item 11. if G9321/G9322 is not used.

If G9321/G9322 is used (retraction from tapped hole):

7. The axis movement type for the C axis/spindle has to be set to “2” (rotary axis with modulo calculation) in MP 1003 00004.
8. Set the processing format to ”modulo format” (bit 7 =1) in S-0-0076 of the C axis/spindle (type of weighting of position data).

9. Bit 8 (drive-controlled change of operation mode) must not be set in S-0-0033 of the C axis/spindle (drive-controlled change of operating mode).
   ”Position control” has to be activated via bit 0 and 1 (depending on the encoder used).
   Bit 3 has to be set (feed forward control).
10. Set C axis/spindle modulo value to “360” in S-0-0103.
11. Modify the interface program with the required interface signals, e.g.
    – NC-O 20.2 “G32 active”
    – NC-O 14.3 “Spindle command”
    – NC-O17.7: “C axis switching active”
    – NC-O17.0: “C axis active”
    – NC-O 14.3 “Spindle command”.
12. Start up the control unit with the set MACODA parameters.
13. Initialize SERCOS.

12.7.2 Activating

G32 (tapping):

- The spindle has to be in “spindle mode” before it is activated.
- The tool for the tapping process has been installed in the spindle.

Create G32 tapping program or G184 tapping cycle with the following specifications:

1. Position the tool above the position to be carried out.
2. Set absolute or incremental data via G90/G91.
3. Select program with 2 blocks “G32 drilling” and “G32 retraction”.

G9321/G9322 (automatic retraction after control reset):

- Abort condition “Control reset” has occurred.
- The tapping tool is stuck in the workpiece.

1. Enter G9321 and G9322 via manual data input with a feedrate value and activate.
   – or –
   call cycle (which contains G9321 and G9322) with the respective data for retraction.

G9321/G9322 (manual retraction after power failure and subsequent control start-up):

- Abort condition “power failure” has occurred.
- The tapping tool is stuck in the workpiece.

- Control unit is ready for operation again after start-up!

1. Define retraction position above the tapped hole position.
2. Enter G91 (incremental programming) manually and activate.
3. Enter G9321 and G9322 via manual data input with all tapping parameters and activate.

As an alternative, call a cycle (which contains points 1. to 3.) for retraction.

☞ **For detailed information on cycles, see DIN Programming Instructions (G instructions).**

12.7.3 Deactivating

When the second G32 block is completed, the spindle switches from “C axis operation” back to “spindle mode”.
G32, G9321/G9322 is thus deactivated.

12.8 Probe

Function:

The control unit supports the evaluation of a probe signal for synchronous axes fed in at the respective drive. For Servodyn drives, this hardware input is referred to as ”MT” (probe).

☞ **To be able to use the probe function within the control unit, a corresponding probe functionality has to be available in the SERCOS drive which the NC can access.**

The NC activates the drive’s own probe function and guides the measuring axis to the programmed position at feedrate after programming the respective G function (G75, G275). Determination of the probe switching time and saving the encoder position current at the switching time is performed directly in the drive. This ensures extremely short response times and thus highly accurate switching position values.
At the end of the block, the NC reads the switching position value from the drive.

The communication between NC and the drive can take place in 2 ways:

- via service channel (standard; data are only transmitted if required)
- via cyclical communication.

Since other data can be transmitted on the service channel too, unnecessary waiting times may occur in the part program. If this is not acceptable, use the cyclical communication.

☞ **For the use of cyclical communication, the drive telegram (DT) must be freely configurable.**

Relevant MACODA parameters (MP):

1003 00011: Probe edge.

1003 00012: Probe function can be activated.

Relevant G functions:

G75 Start measuring cycle.

G175 On-the-fly measurement: initialize.

G275 On-the-fly measurement: start measuring cycle.

Relevant CPL functions:

PPOS,PROBE Request of the current axis actual position at the probe switching time. For information on the differences between the two CPL functions, please refer to the CPL Programming Manual.

SD(9) Provides the information whether one of the activated probes (see MP 1003 00012) has performed a switching action.

12.8.1 Assigning

★ Make sure that the connected drive supports the probe functionality. Otherwise, the probe function within the control unit cannot be used.

★ Make sure that the probe switching signal is generated appropriately and connected at the correct hardware input of the drive (Servodyn-D: hardware input ”MT”).

★ Create a suitable CPL program for measurement evaluation.
For examples, please refer to ”G75” in the DIN Programming Instructions or ”PPOS” in the CPL Programming Manual.

Depending on the desired type of communication (service channel, cyclical communication), the probe function has to be applied in different ways in the following.

☞ **Please note that the drive telegram (DT) has to be freely configurable for the use of cyclical communication!**

Communication via service channel

1. Configure MP 1003 00011 and MP 1003 00012 in accordance with your requirements.
2. If the probe function at the relevant drive has never been set to ”cyclical communication”, no further settings are needed.

   However, if communication via service channel is required instead of an already assigned ”cyclical communication”, remove
   - SERCOS parameters S-0-0130 or S-0-0131 in S-0-0016, and
   - SERCOS parameters S-0-0409 or S-0-0410 in S-0-0026.

   If you have made any changes, save all SERCOS parameters in the EEPROM of the drive or enter their values into the relevant SERCOS files of the NC. The SERCOS files are transferred by the control unit to the drive during SERCOS initialization (for information on SERCOS, see page 4–67 ff.).

CAUTION
Danger of damage to the machine or the workpiece!
Do not initiate a control RESET during machining.

---

3. Start up the control unit and the drives.

Cyclical communication

1. Configure MP 1003 00011 and MP 1003 00012 in accordance with your requirements.
2. Depending on the probe edge parametrized previously, enter
   - SERCOS parameters S-0-0130 or S-0-0131 in S-0-0016, and
   - SERCOS parameters S-0-0409 or S-0-0410 in S-0-0026.

   If you have made any changes, save all SERCOS parameters in the EEPROM of the drive or enter their values into the relevant SERCOS files of the NC. The SERCOS files are transferred by the control unit to the drive during SERCOS initialization (for information on SERCOS, see page 4–67 ff.).

CAUTION
Danger of damage to the machine or the workpiece!
Do not initiate a control RESET during machining.

---

3. Start up the control unit and the drives.

12.9 Digitizing (recording measured values)

Function

The “digitizing” function provides the technical prerequisites to record a surface profile and to store recorded axis positions:

- Recording of the workpiece in X/Y plane and Z direction with automatic **level compensation** by means of simultaneous **online correction** of the Z axis:
  In order to measure a specified surface (e.g. free-form surface), it is necessary to record a dot matrix of the surface that is as detailed as possible. Using an appropriate sensor (e.g. laser), the workpiece is scanned in wave form. The Z sensor infeed axis is continuously adapted to the axis distance line of the surface profile “online” (”axis distance control”, see page 12–31).
- Recording **axis positions in real time** via SERCOS:
  By recording the X, Y axis positions and the Z axis position compensated in real time, the axis distance lines are digitized by placing the dots in a row.
  The surface profile can also be measured with active 5 axis transformation through recording the axis positions of 5 axes.
- **Storing the real time data** (axis data and Z compensation values) in a recording file:
  Saving on the hard disk of the PC panel or an external PC via Ethernet network.
- Digitizing **process controllable** via NC program:
  The recording of axis positions can be started, interrupted, continued and cancelled via NC command in the program.
- Influencing the digitizing process with **programmable presettings**.
- **Synchronizing an external device** with the NC-internal recording of the measured values. One of the high-speed digital outputs of the control unit can be used for this purpose. For more information, please refer to Section 12.9.1 (page 12–42 ff.) under item 5.
- The recording file provides additional information (**standardization functions**) for the subsequent processing of the real time data.
- Programming the laser path where the workpiece is “scanned” is **not** part of the “Digitizing“ function. A program-flow related solution has to be found so that the recorded data are suitable for subsequent processing using a CAD system.

Application options:

- **Transfer of the data** into an appropriate CAD/CAM system (changing, visualizing workpieces, implementing design requirements).
- **Creating an NC program** for “direct automatic execution” **from the recording data** (copying the part on the same or different machine).

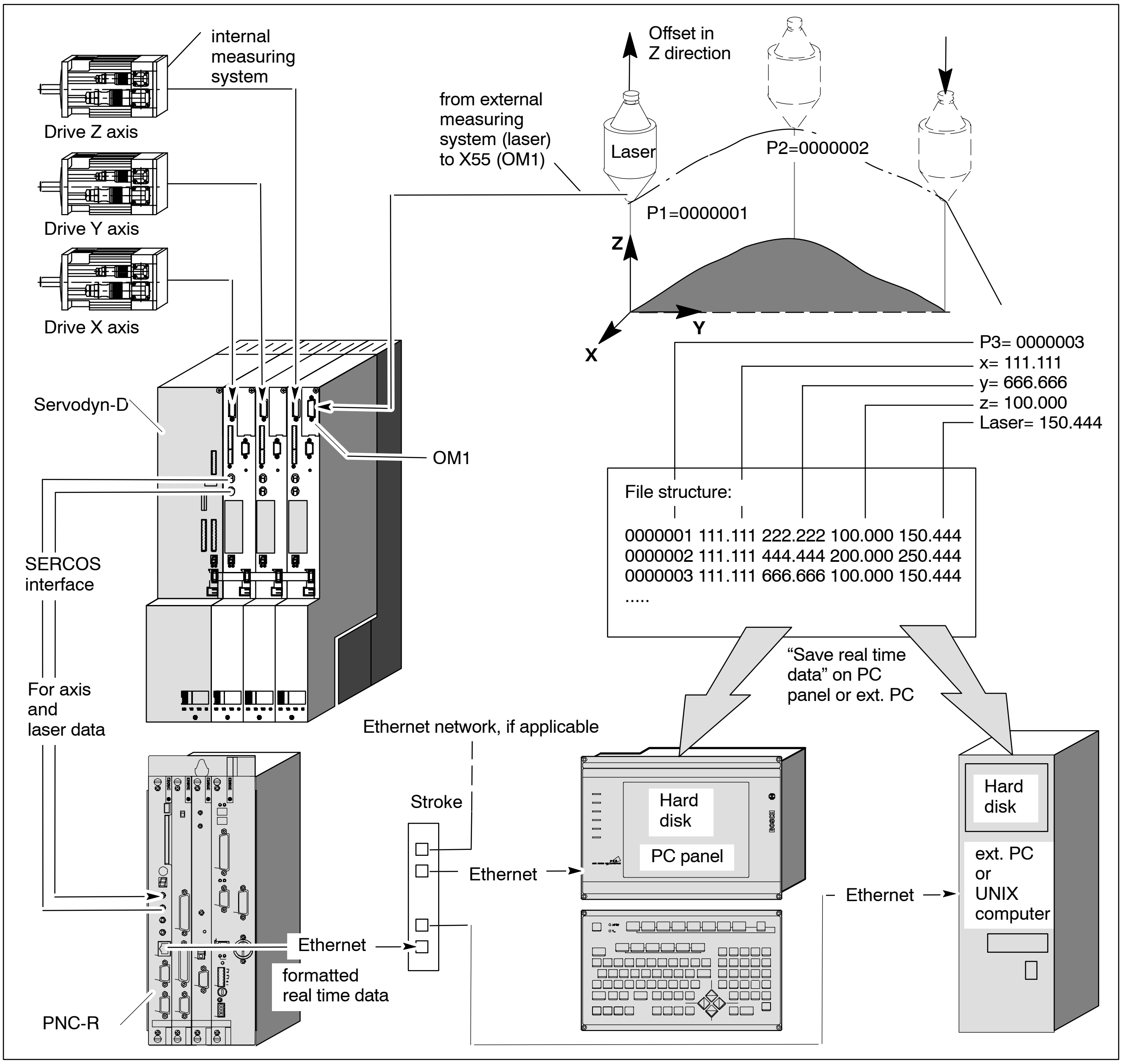
Required hardware components for digitizing:

- SERCOS drive with additional input for external measuring system.
- internal measuring systems (drives) for recording of the real time positions
- external measuring system (e.g. laser) for the online correction

Digitizing

- Ethernet network (TCP/IP) for data transfer to PC
- PNC with largest possible free RAM
- PC panel for operation
- free harddisk memory on PC panel for saving real time data if there is no external PC or UNIX computer (via NFS) (see also ”Network settings” in “PNC Software Installation”).
- ext. PC or UNIX computer networked with PNC to store the real time data on the harddisk.

Example: structure and connection of the PNC and Servodyn-D components for digitizing on a 3 axis machine.



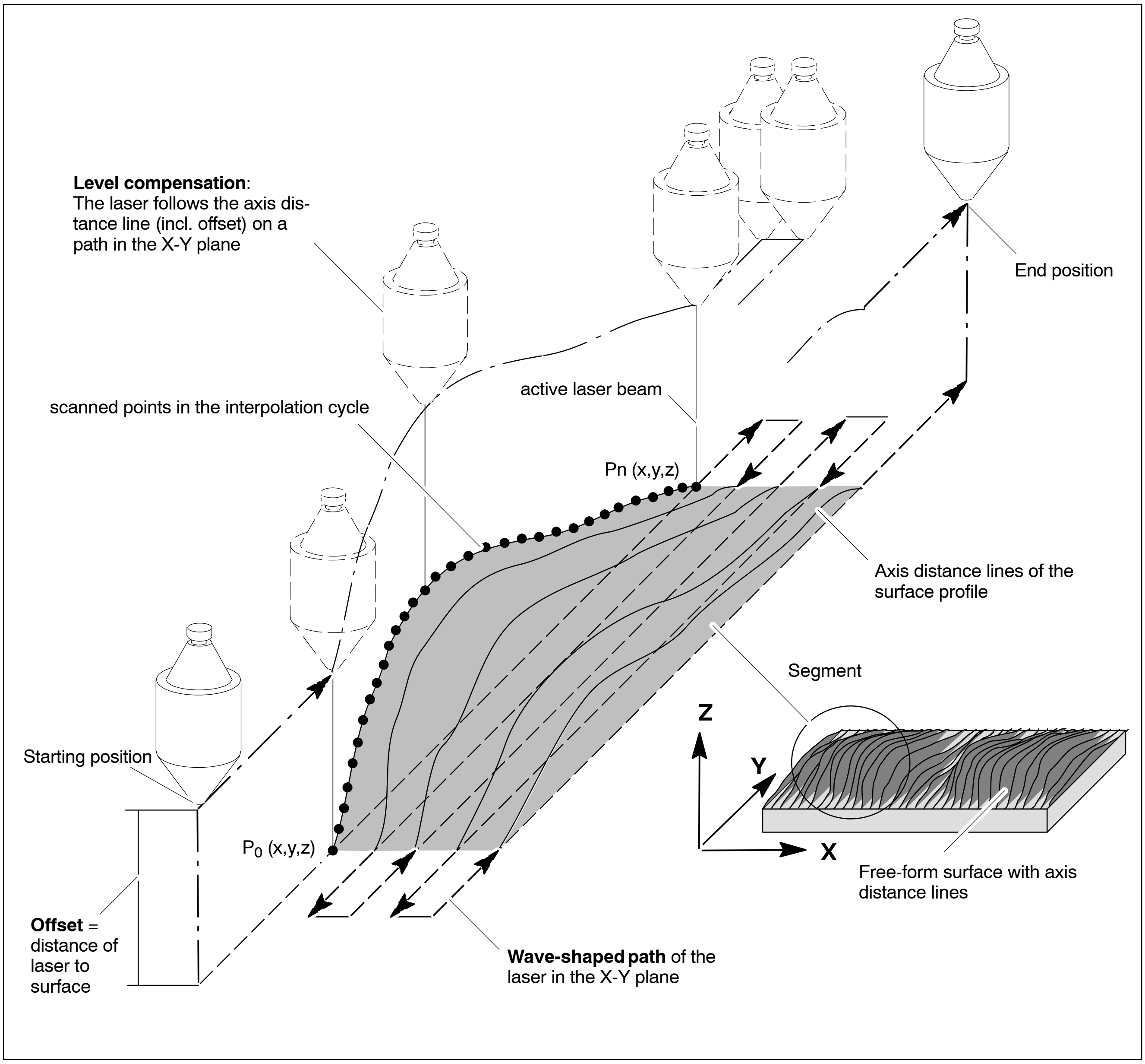
Digitizing process:

- The digitizing process must be initialized in the NC program. The digitizing itself is program-controlled, i.e. the process can be started, interrupted, continued and cancelled in the program.
- For digitizing you need a probe (e.g. laser) which scans the workpiece contactless on a wave-shaped path (see figure below). The laser has its own measuring system which is used to keep a preset distance (offset) to the surface **constant** during the scanning process, i.e. the signals of the laser measuring system are taken into account with the internal measuring system of the laser infeed axis (e.g. Z axis) in real time. Thus it is possible to guide the Z axis along the axis distance line of the surface in the interpolation cycle **(axis distance control)**.

☞ **Extended options in context with the (laser) feed axis (e.g. collision detection, hole detection) are available when you use the "Axis distance control for digitizing" function (see page 12–49 ff.).**

The “external measuring system” (laser) is fed in via a free input of the OM1 (X55) at a Servodyn-D drive module (see figure above).

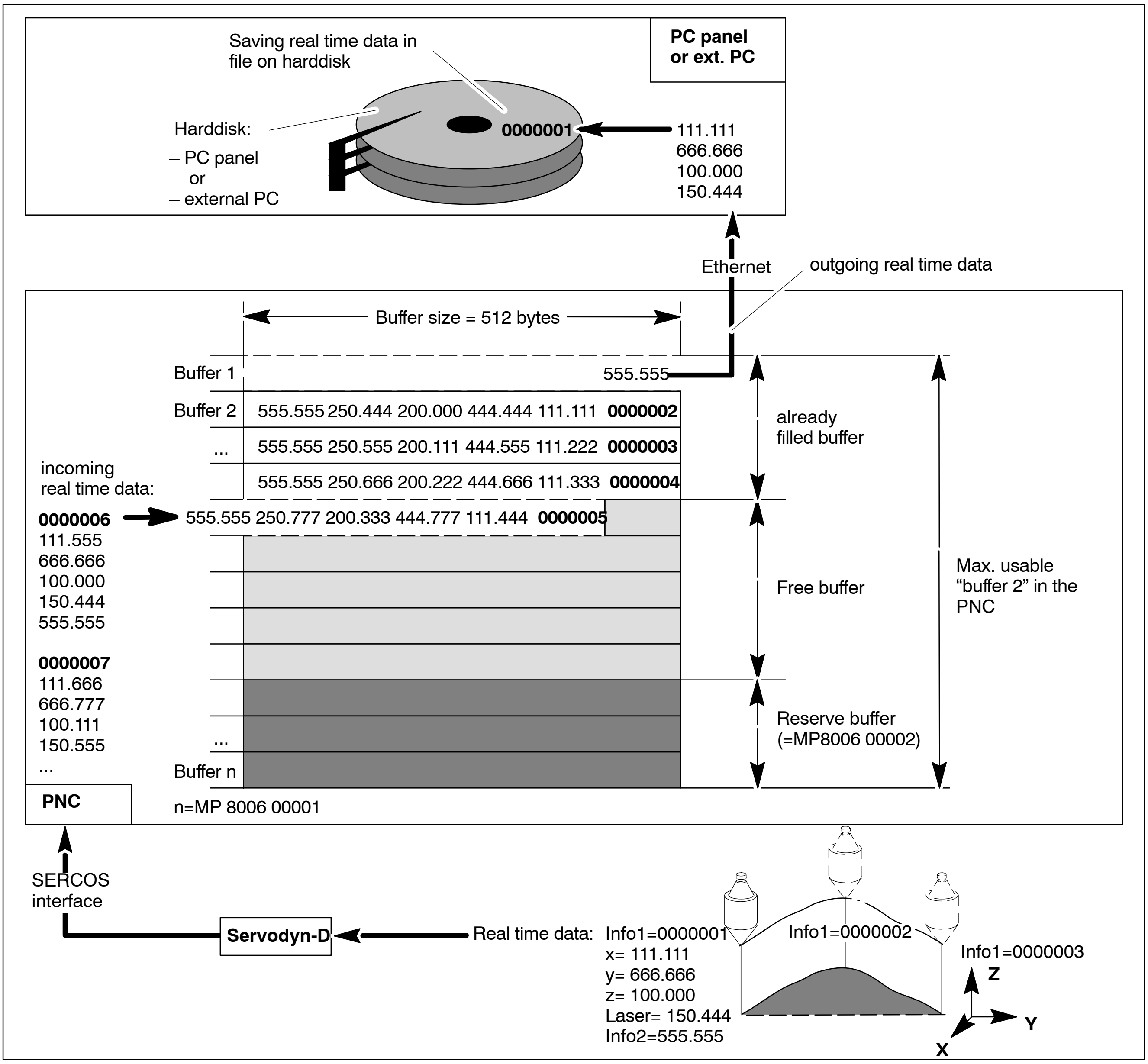
Digitizing



- During scanning, all required axis positions (e.g. X, Y and Z axis) in the interpolator cycle (or a multiple thereof) are recorded and saved on the harddisk of a PC panel, external PC or UNIX computer.
- The saving process is performed via so-called digitizing buffers (memory 2 in the PNC). The real time position data coming from the drive are written into the free RAM of the PNC portion by portion. Already "filled" buffers are saved in the output file (see RECFILE) on the harddisk of a connected PC panel or external PC (via network). The time needed for the saving process depends on the current network load or the system utilization rate of the control unit.
  In this time, several buffers can be filled from the "real time part" (data coming in via SERCOS) (see "already filled buffers" in the figure below).

Digitizing

Example: Buffering of incoming and outgoing real time data



In case of high network load, the data reach the hard disk with delay. This causes a data jam in the digitizing buffer, i.e. more data arrives than can be saved.

If the number of buffers needed is so large that the number of free buffers falls below the ”number of reserve buffers”, the control unit reacts in accordance with the parametrization in command RECFILE(, ,[<File-I/O>]) and RECTIME(, [<AutoStop>],) of the RecordSet:

- The control unit sets the channel feed to zero, so that the loading of real time data is interrupted. This allows full buffers to be saved first (see RECFILE(, ,<File-I/O=1 or2>).
- Each standstill of the channel axes leads to an automatic interruption of the loading of real time data. This allows already filled buffers to be saved (see RECTIME(,<AutoStop=1>,).
- If the ”number of reserve buffers” is not observed and data cannot be saved in time as a result, there is an ”overflow” of the digitizing buffers. Remedy: Increase number of digitizing buffers in MP 8006 00001 (see RECFILE(, ,<File-I/O=0>), or change parameter RECTIME in the ”RECORDSet” NC function.

Restrictions:

- The “Digitizing” function may be active only once in the PNC at any point in time throughout the system. Repeated start (e.g. in different channels) is not permitted.
- Incorrect data entry (e.g. identification number has not been configured in the cyclical axis telegram) or problems with file processing (e.g. insufficient memory available on the hard disk) lead to a runtime error.
- If the desired recording file already exists, it is overwritten by a new digitizing procedure (depending on the parametrization).
- Axis numbers can be selected throughout the system. There is no restriction to the axes of the channel where the digitizing program was started.
- An active digitizing process is cancelled at the end of the program at the latest. The recording file is closed.
- The digitizing procedure is stopped using the RecordBreak instruction, i.e. the control unit interrupts the loading of real time data. The opened recording file stays open.
- If the commanded (setpoint) feedrate of the channel and the actual feedrate of all axes of this channel is equal to 0, the creation of redundant data is avoided (see parameter RECTIME(..,[AutoStop]).
- If saving is not possible because the network load is too high, the feedrate of the channel is set to zero (see above) and the recording of data is aborted at standstill. Buffers which have already been filled with real time data can be saved in spite of possible network load (see RECFILE(, ,[File-I/O])).
- Up to 8 ”measuring points” (axis numbers, axis positions, ..) can be defined (see RECPROBE1 ... RECPROBE8).
- The recorded values can be weighted with a factor for further data processing (e.g. for a CAD system).
- A formatted output of the recorded values is possible.

- Up to 8 additional “measuring points” (e.g. offsets, ..) can be defined (RECPROBE101 ... RECPROBE108) which can be coupled arithmetically (+/–) with RECPROBE1 ... RECPROBE8.
  These measuring points are not saved in the file.
- If the drive (e.g. by other manufacturer) has no backlash compensation, incorrect position values will be output.

Initializing the “digitizing” procedure

The ”Digitizing” function is initialized using the ”RecordSet” and ”RecordSetProbe” NC functions and is located at the start of the NC program which contains the traversing movements needed to measure the workpiece surface.

”**RecordSet**” NC function:

- All parameters of RecordSet have to written in one line.
- The transfer of all parameters is performed using the RecordSet command. Digitizing is thus initialized and can be started.
- In case of RecordSet, the input of RECFILE, RECTIME and RECSEPARATOR is sufficient if the measuring points (position data) have been parametrized using RecordSetProbe **prior to** the RecordSet command.
  This means that the requirement of writing the RecordSet function “in one line” is no longer applicable because the RECPROBEx parameters are defined in the lines ahead of RecordSet.

RecordSet RECFILE(<Name>, {<Rewrite>}, {<File-I/O>})
{RECTIME(<Time>, {<AutoStop>}, {<Potentiometer>})}
RECPROBE1(<Number>,<Name>, {<Factor>},
{<Format>}, {< ± RECPROBEx>})
{. . .}
{. . .}
{RECPROBE8(<Number>, <Name>, {<Factor>},
{<Format>}, {< ± RECPROBEx>})}
{RECPROBE101(<Number>, <Name>, {<Factor>},
{<Format>})}
{. . .}
{. . .}
{RECPROBE108(<Number>, <Name>, {<Factor>},
{<Format>})}
{RECSEPARATOR(<Char>)}

Parameter definition of the “RecordSet” NC function:

RECFILE(<Name>, {<Rewrite>}, {<File-I/O>})

The following applies:

<Name>: **Default: /mnt/<file name>**
Name of the recording file incl. path
(max. 100 characters)

{<Rewrite>}: **Optional, default: 0**,
Options:
”Save file” under the specified <Name> overwrites (=1) the existing file with the same name or is rejected (=0).

{<File-I/O>}: **Optional, default: 1**,
If saving is not possible due to network overload, the feedrate of the channel is set to zero. If the channel axes are at standstill, the loading of the real time data is aborted. This allows already filled buffers to be saved. The ”number of reserve buffers” has to be large enough, so that the data generated during the deceleration process can be saved.
Options:
0: The feedrate is not set to zero.
1: Having saved the filled buffers, the process is continued automatically.
2: Having saved the filled buffers, the process is only continued after cycle start.

{**RECTIME**(<Time>, {<AutoStop>}, {<Potentiometer>})}

RECTIME is **optional**. If RECTIME has not been programmed, <Time> is set to the current NC cycle time, [<AutoStop>] and [<Potentiometer>] to zero.

The following applies:

<Time>: **Default: current interpolation cycle in ms**,
in all other cases 0.5 to 10000.0 ms. Indicating the time cycle in which the position data should be read (e.g. interpolation cycle or an integer multiple thereof). The value is automatically rounded off to an integer multiple of the interpolation, if it is located between an interpolation cycle.

{<AutoStop>}: **Optional, default: 0**
Options:
0: The saving process is not aborted.
1: Digitizing interrupts the saving of data as long as the axes of the channel starting the digitizing process are standing still.

{<Potentiometer>}: **Optional, default: 0**
Options:
0: The value of the channel potentiometer is not effective.
1: Depending on the value of the channel potentiometer, the scanning rate of the digitizing procedure is changed (e.g. potentiometer 50%: double the scanning time).

Digitizing

RECPROBE1(<Number>,<Name>, {<Factor>}, {<Format>},
{< ± RECPROBEx>})
{. . .}
{. . .}
{**RECPROBE8**(<Number>, <Name>, {<Factor>}, {<Format>},
{< ± RECPROBEx>})}

The so-called measuring points are defined using RECPROBE 1..8. These may be position data in the form of SERCOS parameters or "internal measuring points" in the form of "I-ACTPOS" (actual position, compensated by various PNC compensations).
At least 1 measuring point has to be programmed!

The following applies:

<Number>: 0: for measuring points not derived from axes.
1..64: Axis number or specification of the physical axis name.

<Name>: Using the <Name> parameter, the position data of the axis designated under <Number> are transferred.
The following options exist:
– S-0-xxxx: SERCOS parameter (according to the convention e.g.: S-0-0051)
– P-0-xxxx: SERCOS parameter (according to the convention e.g.: P-0-0022)

☞ **It is necessary to agree on the SERCOS parameter in the cyclical axis telegram.**

– I-xxxx: Internal measuring point
– I-ACTPOS
Actual position – LSEC – CCOMP – temporary compensation). Its unit is 0.1 µm.
– I–HSINPUT1 .. I–HSINPUT8
High-speed input of the "PNC highspeed I/O" (PNC-P) or "osa dc I/O" (PNC-R)

{<Factor>}: **Optional**,
the unit of measure of the incoming position data is converted into a different unit using a conversion factor (e.g. 1 µm = 0.001mm). The position data are saved using the new unit of measurement in the recording file. The result is rounded.

{<Format>}: **Optional, default:** " **%d**", formatting the incoming position data into one of the following options:
– %d: 4-byte integer,
– %f: 4-byte float,
– %e: 4-byte float (exponential display)

{< ± RECPROBEx>}: **Optional**: RECPROBEx (x=101, 102, ... ,108). The position value is compensated by "+" or"–" RECPROBE x and the result is saved in the recording file (e.g. offset of laser to surface).
The compensation is indicated by RECPROBE number"**x**" (e.g. RECPROBE1 (1,"I–ACTPOS", , , +101) adds the RECPROBE101 to probe 1). The compensated value is saved in the recording file.

☞ **RECPROBE101 ... RECPROBE108 are not saved in the recording file. These measuring points (e.g. offset) can only be used for taking into account with RECPROBE1 through RECPROBE8.**

{**RECSEPARATOR**(<Char>)}: **Optional, default: "\t"** (Tab)

with the parameters:

(<Char>): Separators between the recorded position data, just in case the data have to be **exported** later (e.g. into a Microsoft Excel table file).

Options: all ASCII characters.
recommended characters:"blanks: " ", Tab: "\t" , comma: ",", semicolon: ";", colon: ":".
Using a "dot" as separator may lead to misinterpretation of the data because the dot is also used in the floating point format (e.g. 10.102).

NC function “**RecordSetProbe**”

- With the aid of RecordSetProbe, the measuring points (position data) can be parametrized **prior to** the RecordSet command. RecordSetProbe can be programmed several times. In this context, RECPROBEx which have already been parametrized cannot be overwritten.

RecordSetProbe RECPROBE1(<Number>, <Name>, {<Factor>}, {<Format>}, {< ± RECPROBEx>})
. . .
. . .
RECPROBE8(<Number>, <Name>, {<Factor>}, {<Format>}, {< ± RECPROBEx>})
RECPROBE101(<Number>, <Name>, {<Factor>}, {<Format>})}
. . .
RECPROBE108(<Number>, <Name>, {<Factor>}, {<Format>})}

Parameter definition of the “RecordSetProbe” NC function:

RECPROBE1(<Number>, <Name>, {<Factor>}, {<Format>}, {< ± RECPROBEx>})
. . .
. . .
RECPROBE8(<Number>, <Name>, {<Factor>}, {<Format>}, {< ± RECPROBEx>})

—> see description of the “RecordSet” NC function:

RECPROBE101(<Number>, <Name>, {<Factor>}, {<Format>})
. . .
. . .
RECPROBE108(<Number>, <Name>, {<Factor>}, {<Format>})

—> see description of the “RecordSet” NC function

Controlling the “digitizing” procedure

The **RecordOn, RecordOff, RecordBreak, RecordContinue** NC functions are used to influence the saving of the real time data during the execution of the NC program.

NC function “**RecordOn**”

RecordOn Starts the “digitizing” process.
The recording file is opened.
RecordOn starts the recording and saving of the measured values (position data). Required parameters have to be set previously using RecordSet. If the “digitizing” function has already been activated, further RecordSet instructions result in a runtime error.

NC function “**RecordOff**”

RecordOff Ends the “digitizing” process.
RecordOff is used to end the recording of the measured values and closes the recording file.
Following the RecordOff command, it is possible to restart digitizing with RecordOn or to perform a further initialization using RecordSet (...).

Other functions to end the digitizing process:
- End of program
- Channel or system control reset

NC function “**RecordBreak**”

RecordBreak Interrupts the “digitizing” process.
RecordBreak interrupts the digitizing process, measured values are no longer recorded and saved, but the recording file stays open.
The feedrate is not affected by this, so that execution of the NC program continues. If “End of program“ (e.g. M30), control reset or RecordOff is detected during the recording break, the recording file is closed.

NC function “**RecordContinue**”

RecordContinue{RECTIME(<Time>, {<AutoStop>}, {<Potentiometer>})}

Continue the “digitizing” process.
RecordContinue is used to continue the digitizing process, i.e. the recording and saving of measured values. A scanning time can be specified when RecordContinue is programmed. It is thus possible to change the time indicated in RECTIME.

For information on the RECTIME parameter, see the “RecordSet” NC function.

Digitizing

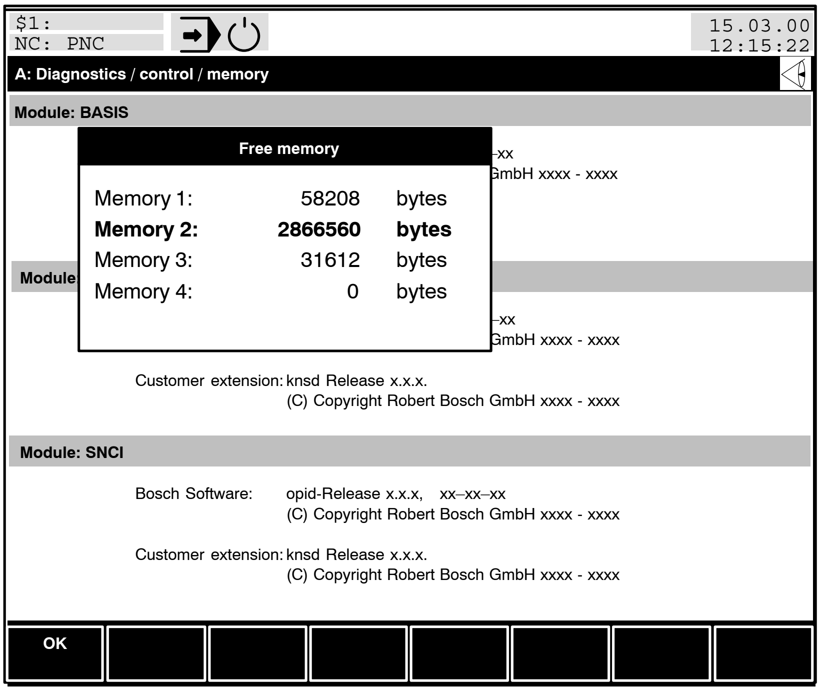
In addition, the digitizing process can be influenced by the following settings:

- Stop “Saving of real time data” automatically when there is no traversing movement:
  —> Setting in RecordSet(..., RECTIME(<AutoStop=**1**>),...)
- Close recording file automatically at the end of the program:
  —> Programming of M30 in the NC program.

Relevant MACODA parameters (MP):

8006 00001: Number of digitizing buffers (buffer size 512 bytes).
The maximum number of digitizing buffers is obtained using the equation:
max. number of buffers=“size of the free memory 2” / 512.

The size of the free memory 2 can be determined via DIAGNOSTICS ▶ CONTROL ▶ MEMORY



8006 00002: Number of reserve buffers.
Recommendation: approx. 30% of MP 8006 00001. In case of data jam, the number of reserve buffers can be increased and/or, it is possible to work with a lower feedrate, if required.

4075 00102: Assignment of digital outputs.
See Section 12.9.1 (page 12–42 ff.) under item 5.

12.9.1 Assigning

The RecordSet and RecordSetProbe NC instructions used to initialize and program the digitizing process partly require data that should be set at the assigning stage already.

Therefore, identify:

- the name or physical number of the axes involved in MACODA (e.g. 1= X axis).
- the relevant SERCOS identification numbers (e.g. actual position of the motor encoder: S-0-0051) or use the "I–ACTPOS" internal measuring point.
- unit and format of the SERCOS identification numbers or of the "I–ACTPOS" internal measuring point, respectively.
- offset of the laser (e.g. SERCOS identification number P-0-0553 "Position actual value 2" of the Z axis)
- format, unit and separators of the real time data, so that these can later be loaded into a CAD or table calculation program for further processing without any problems.
- if required, include external PC in the network in which the PNC is working.
- mount directory and PC for saving of the recording file. If required, include an external PC in the network in which the PNC is working.
- The NC-internal recording of measured values can be synchronized with an external device. If the value 6 is entered in MP 4075 00102 "Assignment of digital outputs", the level of this high-speed output is changed with every internal recording procedure.

Assigning:

1. Determine the number of possible digitizing buffers with the aid of "Memory 2" (DIAGNOSTICS ▶ CONTROL ▶ MEMORY) and enter it in MP 8006 00001.
2. Determine reserve buffer (= approx. 30% of MP 8006 00001) and enter it in MP 8006 00002.
3. If no mount directory for saving the recording file has been defined in the PNC, you have to link a directory:
   to do this, start the **T3Config** program on the PC panel.
   Select the **Select NFS-Mount-Directory** sub-item in the "T3SWConfig" menu item.
   You can choose any directory on a released drive (hard disk) of a networked PC for your mount directory (for more information, please refer to the "Software Installation" manual).
4. Due to an axis telegram (SERCOS) with the respective configuration, both the axis actual value and the sensor actual value are available in the PNC.
   For this purpose, modify the SERCOS file by including the SERCOS parameter for the sensor actual value (e.g. P-0-0553, position actual value 2) of the connected external measuring system in the section for the cyclical drive telegram.

Digitizing

5. If the NC-internal recording of the measuring values is to be synchronized with an external device, one of the high-speed digital outputs of the control unit can be used for this purpose.
   To do this, assign the value ”0006” to the desired digital output in MP 4075 00102. The level of this output will then change with every internal recording process.
6. Restart the control unit and the drives.

Define the parameters of the RecordSet NC function:

7. Determine strategy in case of data jam in the buffer of the PNC (via RECFILE (..)).
8. As an alternative to 7. via RECTIME (...), when the real time data are to be recorded in the interpolation cycle (or a multiple thereof; interpolation cycle: MP 9030 00001).
9. Determine potentiometer reaction RECTIME (..).
10. Selection of the real time data RECPROBE1 ...8.
11. Standardize unit and format of the position data to be saved RECPROBE1 ...8.
12. Enter offset of the laser RECPROBE101 ...108.
13. Determine separators between two real time values, if you intend to export the data.
14. Program “digitizing” test run: A path in the X-Y plane is programmed for the surface area to be digitized, on which the scanning head (e.g. laser) moves to record the workpiece surface in Z direction.
15. Starting the test run.
16. Test the RecordOn, RecordOff, RecordBreak and RecordContinue functions in the digitizing process.
17. Analysis of the recording file:
    – Settings: are all settings from RecordSet available?!
    – Data: have the real time data been recorded?!
    – Statistics: buffer overflow, network failure, ...?!

Example 1: programming the actual position with the ”I–ACTPOS” internal measuring point of the PNC and (S-0-0053)

```
RecordSet RECFILE(”/mnt/digit/example.dig”,1,0) REC-
TIME(8,1,0) RECPROBE1(1,”I-ACTPOS”) RECPROBE2(2, ”I-ACT-
POS”) RECPROBE3(3, ”I-ACTPOS”)  RECPROBE4(3, ”S-0-0053”)
```

RecordSet is programmed in one single line here. The parameters have the following meaning:

- The real time data are stored in the “example.dig” recording file. It is saved in the “/mnt/digit” path. An existing file with the same name is deleted before recording starts.
- The recording takes place every 8 msec.
- Since <File-I/O> is equal to 0, no measures will be taken in case of network problems to prevent a memory overflow.
- The compensated actual positions are recorded of the axes with the physical axis numbers 1, 2 and 3.
- The ”second” actual value of axis 3 is recorded additionally.

- Since <AutoStop> is equal to 1, no data is recorded when the axes are standing still.
- Since <Potentiometer> is equal to 0, the potentiometer does not change the scanning time.

Example 2: programming the actual position with the actual position of the motor encoder (S-0-0051)

```
RecordSet RECFILE("/mnt/digit/example.dig",1) RECTIME(8)
RECPROBE1(1, "S-0-51") RECPROBE2(2, "S-0-0051")
RECPROBE3(3,, "S-0-0051")
```

RecordSet is programmed in one single line here. The parameters have the following meaning:

- The real time data are stored in the “example.dig” recording file. It is saved in the “/mnt/digit” path. An existing file with the same name is deleted before recording starts.
- The recording takes place every 8 msec.
- The actual positions of the motor encoder (S-0-0051) of the drives with the physical axis numbers 1, 2 and 3 are to be recorded.
- Since<File-I/O> is not indicated, the feedrate will be reduced to zero in case of network problems. Having saved the buffers already filled, the part program is continued automatically.
- Since <AutoStop> is equal to 0, data is also recorded when the axes are standing still.
- Since <Potentiometer> is equal to 0, the potentiometer does not change the scanning time.

Example 3: programming the actual position with the ”I–ACTPOS” internal measuring point of the PNC and an offset (S-0-0053) (position actual value 2)

```
RecordSetProbe RECPROBE1("X", "I-ACTPOS", 0.0001, "%9.3f")
RecordSetProbe RECPROBE2("Y", "I-ACTPOS", 0.0001, "%9.3f")
RecordSetProbe RECPROBE3("Z", "I-ACTPOS", 0.0001, "%9.3f",-101)
RecordSetProbe RECPROBE101("Z", "P-0-0553", FACTOR)
RecordSet RECFILE("/mnt/digit/example.dig",1,2) RECTIME(8,1,0)
```

RecordSet is considerably shorter here, because the RECPROBE parameters have been programmed previously using RecordSetProbe.

The parameters have the following meaning:

- The compensated actual positions are recorded of the axes with the physical axis names X, Y and Z.
- RECPROBE101 is assigned the identification number P-0-0553 (position actual value 2) of the Z axis. In the example, the value of the identification number P-0-0553 is normalized to match RECPROBE3 with the aid of the FACTOR CPL variable.
- In case of axis Z, the value of RECPROBE101 is subtracted.

- The measured values are stored in the /mnt/digit/example.dig file. An existing file with the same name is deleted before recording starts.
- Since<File-I/O> is equal to 2, the feedrate will be reduced to zero in case of network problems. Having saved already filled buffers, the part program is only continued after "Cycle start".
- Since <AutoStop> is equal to 1, no data is recorded when the axes are standing still.
- Since <Potentiometer> is equal to 0, the potentiometer does not change the scanning time.

12.9.2 Activating

Precondition:

- RecordSet has been set according to the machine configuration.

Creating an NC program

1. Select name for the NC program.
2. Insert the "parametrized RecordSet" at the start of the NC program.
3. Program the path where the laser is supposed to "scan" the workpiece.
4. By programming the "RecordOn" command, you determine from what point on the position data is recorded.
5. Program the "RecordBreak" command to temporarily interrupt the digitizing process (e.g. for traversing outside of the workpiece surface).
   Program "RecordContinue" to continue the digitizing process.
6. Program "RecordOff" or "M30 end of program" to end the digitizing process.

Executing:

7. Start the digitizing process with recording and saving of the position data.

☞ **If the movement of the laser halts during execution, the cause may be a bottleneck in transferring the real time data via the network. Information on the cause of the fault may be found in the recording file in the "[Statistics]" area.**
In many instance, these problems can be solved by modifying the settings in "RecordSet".

12.9.3 Deactivating

1. The digitizing process can be cancelled via
   – RecordOff,
   – end of program,
   – channel control reset or
   – system control reset.

12.9.4 The recording file

Type of file

The digitizing result is saved as ”recording file” in ASCII format.

File name and path

Indicate file name and path under which the recording file is to be saved in “<Name>” of the **RECFILE** parameter (see RecordSet initialization function).

Example: <Name> corresponds to **/mnt/digit/example.dig**

Opening of file:
(start of the recording process)

Open the recording file: **RecordOn** command.
Incoming position data are saved in the structure indicated below.

Closing of file:
(end of the recording process)

Close the recording file: **RecordOff** command.
Position data which are still in the buffer are saved.

File structure

The structure of the recording file is divided into 3 sections. The following **keywords** are used to identify the start of a section:

1. **[Settings]**:
   Contains information on configuration, recording file and part program used to digitize the workpiece.
2. **[Data]**:
   Contains real time data, organized in lines and columns.
   The columns contain the recorded values (e.g. axis position of the X axis) defined using RECPROBE1..8.
   Each line contains a complete block of RECPROBE1..8 of a single scanning time. Within the data, programmed interruptions are indicated with RecordBreak and a comment. If no comment is entered in connection with the RecordBreak command, this entry is totally deleted.

Column

```
0000001 111.111 222.222 100.000 150.444 555.555
0000002 111.111 444.444 200.000 250.444 555.555
0000003 111.111 666.666 100.000 150.444 555.555
.....
```

Line

3. **[Statistics]**:
   Contains statistical information which occurred during the digitizing process.

Detailed description of the file structure:

[Settings]:

[Settings]:
[DATE]
<Creation date of the output file >
[PROGRAM]
<File name of the part program>
[RECFILE]
(see also RecordSet RECFILE)
Name: <File name>, Rewrite: <Rewrite>, File-I/O: <File-I/O>
[RECTIME]
see also RecordSet RECTIME
RECTIME: <Time>, AutoStop: <AutoStop>, Potentiometer: <Potentiometer>
[RECPROBE1]
see RecordSet RECPROBE1
Axis: <Number>, Name: <Name>, Factor: <Factor>, Format: <Format>
....
[RECPROBE8]
[RECSEPARATOR]

[Data]

[Data]
<RECPROBE1(<Name>)> {<RECPROBE2(<Name>)} ... {<RECPROBE8(<Name>)}

[Statistics]

[Statistics]

NmbOfSamples:	Number of recorded lines
LimitUsedBuffers:	Max. number of buffers available. The real time data of the digitizing process are temporarily stored in buffers before they are written into the file.
MaxUsedBuffers:	Max. buffer distance reached. Saving in the file may be much slower than the recording of the real time data, depending on the system or network load. This leads to a buffer distance between the buffer described in the PNC at present and the buffer being saved on the hard disk.
DownSlopesDetected:	Number of “Exceeding the critical buffer distance”. If the above mentioned buffer distance exceeds a maximum value (=MP 8006 00002, the control unit tries to stop the axis movement in dependence on the ”File-I/O” parameter, in order to subsequently interrupt the digitizing process.

DownslopesExecuted: Number of downslopes executed until standstill (see AutoStop).
When the critical buffer distance is exceeded, a deceleration process of the axes is initiated. When standstill is reached, the digitizing process is interrupted, so that the real time data can be written into the recording file. The NC program can be continued as soon as free buffers are available again.
The output number is a measure for the system or network load.

TimeAtEnd: Time “End of digitizing”.

Info: for internal purposes only

Example: Contents of the “example.dig” recording file

```
[Settings]
[DATE]
6.11.2000 10:35
[PROGRAM]
Example.cnc
[RECFILE]
Name: /mnt/digit/example.dig, Rewrite:  1, File-I/O: 0
[RECTIME]
RECTIME: 8, AutoStop: 0, Potentiometer: 0
[RECPROBE1]
Axis: X, Name: I-ACTPOS, Factor: 1, Format: %d:
[RECPROBE2]
Axis: Y, Name: I-ACTPOS, Factor: 1, Format: %d:
[RECPROBE3]
Axis: Z, Name: I-ACTPOS, Factor: 1, Format: %d:
[RECPROBE4]
Axis: Z, Name: S-0-53, Factor: 1, Format: %d:
[RECSEPARATOR]
<Tab>
[DATA]
111.111 222.222 100.000 150.444
111.111 444.444 200.000 250.444
111.111 546.345 122.000 350.444
...
...
[Statistics]
NumbOfSamples: 1000
LimitUsedBuffers: 50
MaxUsedBuffers: 30
DownslopesDetected: 0
DownslopesExecuted: 0
TimeAtEnd: 7.11.2000 11:50
Info
```

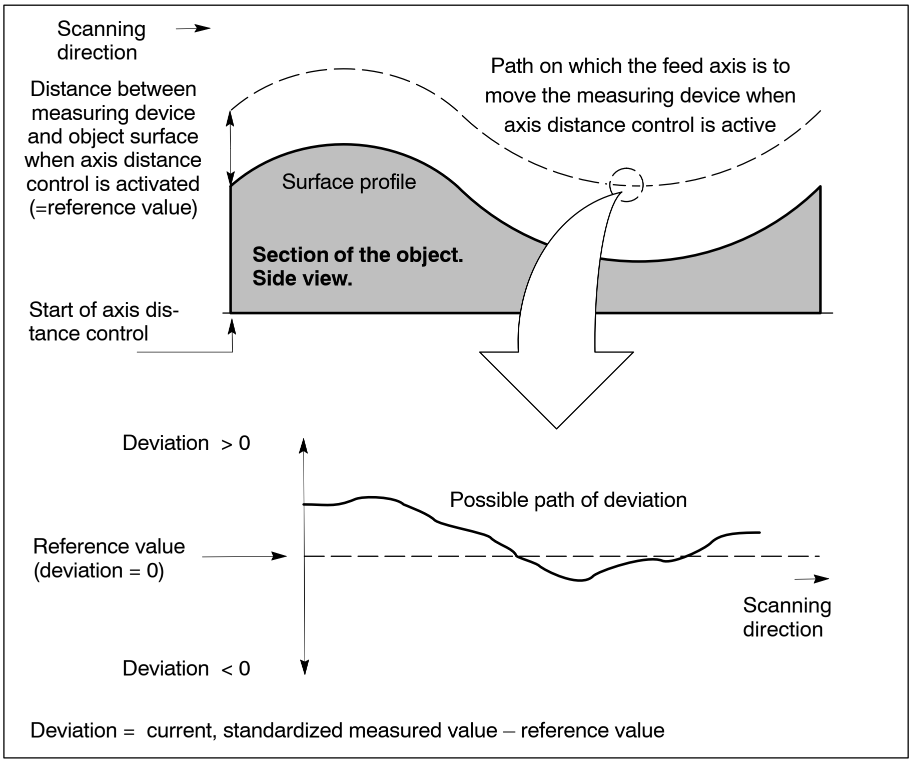
12.10 Axis distance control for the digitizing process

Function:

When digitizing a 3-dimensional contour, it is necessary to determine and save the coordinates of the surface in question. For this purpose, you use a part program which

1. first of all moves a measuring device (e.g. laser) via the feed axis (e.g. Z) to a predefined distance from the contour. This distance serves as reference value in the subsequent process.
2. activates the "Axis distance control for digitizing" function
3. starts the digitizing process (see page 12–29 ff.) and
4. subsequently scans the object to be digitized in wave-shaped movements in the plane defined by the two other axes (e.g. X/Y plane).

Using the measuring system, changes in the distance in relation to the reference value are recorded constantly, transmitted to the NC and converted to the unit mm, if required (for information on standardization, please refer to MP 7050 00720). For this purpose it is necessary to feed the measured data into the external encoder connection (e.g. as incremental encoder signal) and to transmit them to the NC cyclically via SERCOS interface.
Having activated the axis distance control, the NC is able to control the position of the feed axis so that the distance between measuring device and object surface current at the time of activation remains constant (deviation from the reference value = 0). It is thus ensurde that the available working range of the measuring device is not exceeded.



Axis distance control for the digitizing process

The following options are available within the function:

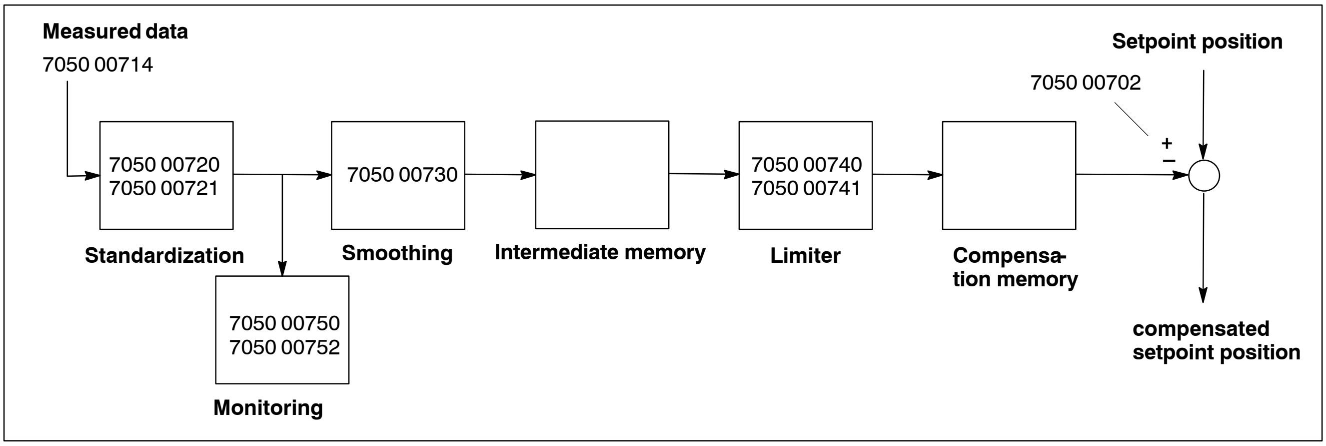
- program-controlled activation, interruption, continuation and deactivation of axis distance control. See ”Relevant NC functions”.

☞ **Axis distance control is always switched off following control unit start-up.**

- program-controlled suppression of pertaining MACODA parameters. See ”Relevant NC functions”.

☞ **Suppressed MACODA configuration data become effective again after program deselection, channel or system control reset only!**

- monitoring of the deviation via programmable tolerance range.
  In this context, a collision and hole detection is possible (see Section 12.10.1, page 12–54 ff.).
- limitation of the current, standardized measured value with regard to speed and acceleration. See MP 7050 00740 and MP 7050 00741.
- Smoothing of the arriving measured values.
  Suppresses ”freak values”, see MP 7050 00730.



Points in the measured data processing path where MACODA parameters take effect

Restrictions:

- Software version V47 (or higher) is required for our Servodyn-D drives.
- The feed axis must not be removed from the channel (e.g. via axis transfer) when axis distance control is active.
- Axis distance control is only possible for axes which are not involved in an axis transformation with active coordinate programming.
- The software limit switches of the feed axis are inactive when axis distance control is switched on.
- The feed axis cannot be moved to a machine-oriented absolute position when axis distance control is active.

Axis distance control for the digitizing process

Relevant MACODA parameters (MP):

7050 00700 Axis distance control available.
0: no; 1: yes

7050 00702 Axis distance control: number of the channel axis and compensation direction.
Determines which axis of the channel is to undergo axis distance control. An axis number

- with or without positive sign has the effect that the compensation values are taken into account in positive movement direction of the distance-controlled axis.
- with negative sign has the effect that the compensation values are taken into account in negative movement direction of the distance-controlled axis.

7050 00708 Axis distance control: interface signal, number of the customer output of the channel.
Determines at which output of the respective channel interface an active axis distance control is to be reported (output set: axis distance is active).

–1: no signal output

0: NC-O22.0
:
7: NC-O22.7

7050 00710 Axis distance control: sensor selection, device group.

7050 00712 Axis distance control: sensor selection, number of the device.

7050 00714 Axis distance control: sensor selection, special identification (S-0-xxxx, P-0-xxxx, @xxxx).
Specifies in which drive parameter the encoder pulses coming from the measuring device are counted. The SERCOS parameter entered here must be included in the drive telegram (S-0-0016) as well.

7050 00720 Axis distance control: sensor increments –> mm – factor.
Serves to convert (together with MP 7050 00721) the data (e.g. encoder pulses) transmitted to the NC to the unit millimeters.

7050 00721 Axis distance control: sensor increments –> mm – divisor.
See MP 7050 00720.

7050 00730 Axis distance control: filter – smoothing time in ms.
Parametrizes a square filter. Entering 0 deactivates the filter.

7050 00740 Axis distance control: limiter – speed in mm/min.
Maximum speed by which the current, standardized measured value is allowed to change. This can be used, e.g. when traversing over a hole, to prevent the feed axis from sinking into the hole at its maximum permitted speed.

Axis distance control for the digitizing process

7050 00741 Axis distance control: limiter – acceleration in m/sec^2
Maximum acceleration by which the current, standardized measured value is allowed to change.

7050 00750 Axis distance control: monitoring – collision detection in mm (0.000=off).
Determines the tolerance limit for the collision detection (in mm). If the deviation from the reference value exceeds the value specified here, the NC reports a collision.
Entering 0.000 deactivates the collision detection.

7050 00752 Axis distance control: monitoring – hole detection in mm (0.000=off).
Determines the tolerance limit for the hole detection (in mm). If the deviation from the reference value exceeds the value specified here, the NC ”freezes” the axis distance control. In this state, the feed axis remains in her momentary position.
Entering 0.000 deactivates the hole detection.

Relevant SERCOS parameters:

S-0-0016: Configuration list DT.
Determines which SERCOS parameters are to be transmitted to the NC cyclically.
The SERCOS parameter specified in MP 7050 00714 has to be contained here (e.g. P-0-0553 ”Position actual value 2 incremental”).

S-0-0115: Position encoder type (external encoder).

S-0-0117: Rotation encoder 2 resolution (external encoder).
Only necessary if encoder type ”rotation encoder” has been set in S-0-0115.

S-0-0118: Linear encoder resolution (external encoder).
Only necessary if encoder type ”linear encoder” has been set in S-0-0115.

S-0-0257: Multiplication 2 (external encoder).

P-0-0552 Evaluation external encoder
Must be assigned the value ”1”.

Relevant NC functions:

DistCtrlOn Starts axis distance control. In addition, the current distance between measuring device and surface is adopted as reference value.
If programmed alone, the configuration data programmed via MACODA will take effect.
It is possible optionally to suppress some configuration data by programming various additional commands.

☞ **Suppressed MACODA configuration data become effective again after program deselection, channel or system control reset only!**

Axis distance control for the digitizing process

DCAXIS(<Axis>,<Comp>)
Suppresses MP 7050 00702.

<Axis>	Name or number of the channel axis which is to undergo axis distance control.	
<Comp>	+1 or 1:	takes into account compensation values in positive movement direction
	–1:	takes into account compensation values in negative movement direction

DCFILTER(<Time>)
Suppresses MP 7050 00730.

<Time>	0:	Filter off
	>0:	filter on, values in ms

DCLIMIT([<Speed>],[<Accel>])
Suppresses MP 7050 00740 and 7050 00741.

<Speed>	Suppresses MP 7050 00740. Input value depending on active unit of measure system (G71,G70) in mm/min or inch/min.
<Accel>	Suppresses MP 7050 00741. Input value depending on active unit of measure system (G71,G70) in m/s^2 or 1000 inch/s^2.

DCMON([<Collision>],[<Hole>])
Suppresses MP 7050 00750 and 7050 00752.

<Collision>	Suppresses MP 7050 00740. Input value depending on active unit of measure system (G71,G70) in mm/min or inch/min.
<Hole>	Suppresses MP 7050 00741. Input value depending on active unit of measure system (G71,G70) in m/s^2 or 1000 inch/s^2.

DistCtrlBreakOn Interrupts axis distance control. The current compensation value remains active.

DistCtrlContinue Continues a previously interrupted axis distance control. The NC adjusts the deviation from the reference value as quickly as possible.

DistCtrlOff Ends axis distance control, adopts the current compensation value and stops the axis movement.
If DistCtrlOff is programmed together with a traversing movement in one block, the NC deactivates axis distance control only after execution of the movement.

Relevant CPL functions:

CPOS, WPOS: Neither function takes into account the compensation value of the axis distance control!

MPOS: Provides the position of the coordinate including the compensation value.

MPOS: Provides the setpoint position of the axis including the compensation value.

APOS, PPOS, PROBE: All CPL functions which are based on the actual position or the measured value of a drive take into account the axis distance control.

Relevant interface signals:

NC-O 22.x see MP 7050 00708.

12.10.1 Monitoring deviations

The data coming from the measuring device via SERCOS interface are converted to the unit mm by the NC using MP 7050 00720 and MP 7050 00721.

Please note that for correct interpretation of the measured data

- you need an appropriate configuration of the external encoder interface in the drive (S-0-0-0115, bit 3 or bit 5),
- you have to take the compensation value into account correctly (via MP 7050 00702: sign, axis number).

As a consequence, the reported position data must

- decrease when the distance between measuring device and surface increases, because the surface position (in relation to the measuring device) is lowered.
- increase when the distance between measuring device and surface decreases, because the surface position (in relation to the measuring device) is raised.

The standardized measured value is now checked for compliance with the maximum permissible tolerance limits.

The basis for the check is the distance between measuring device and object surface active when axis distance control is switched on. In the following, the reference value serves as zero line of a tolerance range. If the current standardized measured value does not correspond to the reference value, this indicates a positive or negative deviation from the zero line.

Traversing a hole becomes apparent by a jump of the deviation in negative direction because the encoder suddenly signals a lower position of the scanned surface.
Opposed to this, the effect of a sudden rise in the contour is that the reported surface position is raised abruptly, which in turn leads to a jump of the deviation in positive direction.

Axis distance control for the digitizing process

Example:

- When the axis distance control is activated, the distance to the surface is 100 mm. The encoder thus reports e.g. the standardized value 100 mm. This distance is now used as reference value:

  Deviation = current, standardized measuring value – reference value = 100 mm – 100 mm = 0 mm

- The surface contour now drops by 60 mm. The encoder reports the position 40 mm:

  Deviation = current, standardized measuring value – reference value = 40 mm – 100 mm = –60 mm

  The NC immediately compensates the deviation by readjusting the feed axis. The deviation is thus reduced to 0. At the same time, the standardized position reported by the encoder increases to 100 mm again.

- The surface contour now abruptly rises from 40 mm to 100 mm. The encoder reports the position 160 mm:

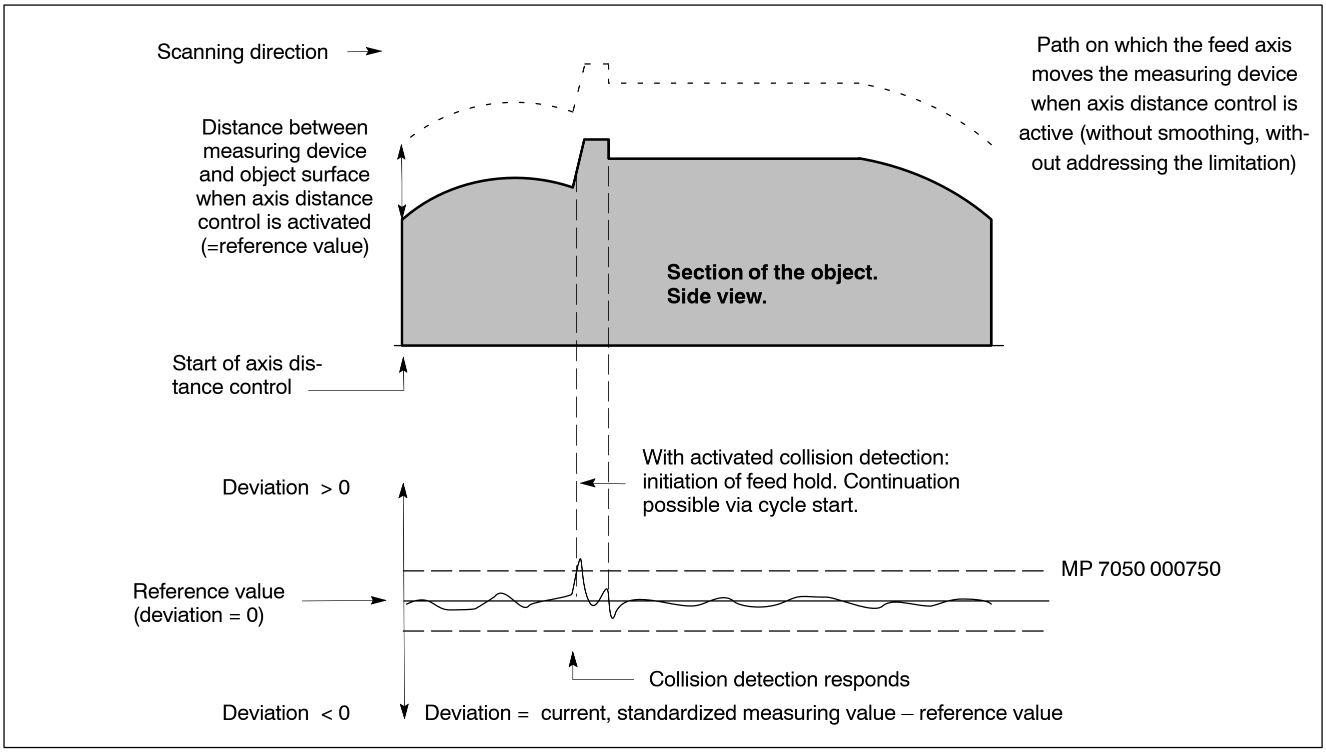
  Deviation = current, standardized measuring value – reference value = 160 mm – 100 mm = +60 mm

  The NC immediately compensates the deviation by readjusting the feed axis. The deviation is thus reduced to 0. At the same time, the standardized position reported by the encoder decreases to 100 mm again.

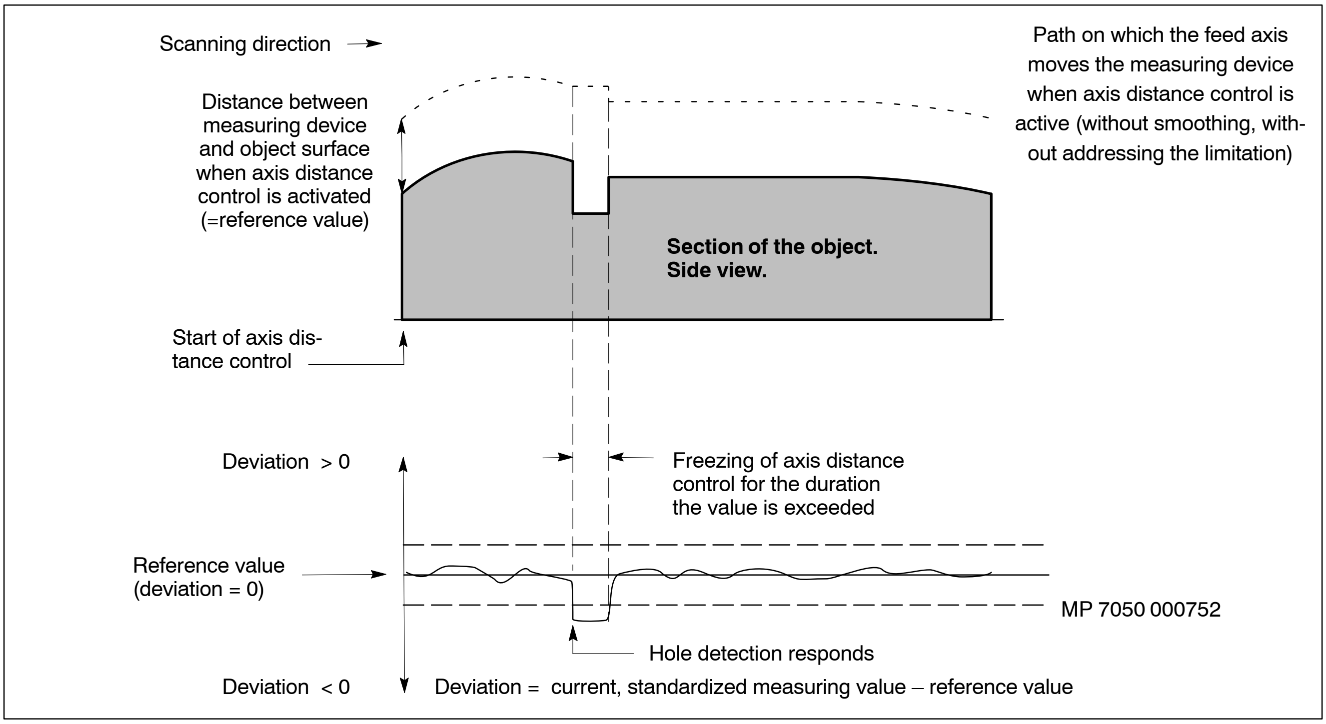
The sudden increase of the deviation in positive or negative direction can be used for **collision and hole detection**. For this purpose, the current deviation is constantly compared with MP 7050 00750 (collision detection) and MP 7050 00752 (hole detection).
If the current deviation exceeds the respective parametrized limit value, the NC responds as follows:

- after detection of a collision:
  the NC generates a warning and stops the movement on the channel via feed hold.
  The operator can, if no collision has occurred, continue the movement with cycle start (possibly with lower feedrate potentiometer setting).
- after detection of a hole:
  the NC interrupts the axis distance control for the duration the value is exceeded (the same effect as DistCtrlBreak). Scanning of the object continues. In this case, the current compensation value remains active, i.e. the feed axis does not change its coordinate.

Axis distance control for the digitizing process



Principle of collision detection



Principle of hole detection

12.10.2 Assigning

1. Make sure that the measured data signals of the measuring device are suited for the external encoder connection of the drive.
2. Set the relevant SERCOS parameters (see above) in the drive correctly and save them in the EEPROM or transfer them via the control unit to the drive during SERCOS initialization (for information on SERCOS, see page 4–67 ff.).
   Make sure that the function ”Position monitoring 2nd encoder” is deactivated (P-0-0510; bit 5=0).
3. Check whether the position actual value in P-0-0553 (pulse of external encoder) behaves correctly in case of changes in distance between measuring device and scanned surface.
   A decrease in distance must result in an increase in the pulse number in P-0-0553. The behavior can be set via S-0-0115 (bit 3, direction of movement or bit 5, counting direction).
4. Enter the SERCOS parameter which contains the position actual value of the measuring device and is to be transmitted cyclically to the NC into the drive telegram (S-0-0016).
   If the measured data are fed into the drive e.g. as incremental encoder signals, enter the SERCOS parameter ”P-0-0553” into the drive telegram (S-0-0016).
   Check whether the correct telegram has been configured in S-0-0015. SERCOS parameters contained in S-0-0016 are only transmitted to the NC if the ”freely configurable telegram” is selected in S-0-0015.
5. Configure the parameters specified under ”Relevant MACODA parameters” according to your application.

CAUTION
Danger of damage to the machine or the workpiece!
Make sure that the compensation direction (MP 7050 00702) has been parametrized correctly.

☞ **MP 7050 00702 and MP 7050 00730 through 7050 752 may also be influenced via the ”DistCtrlOn” function.**

CAUTION
Danger of damage to the machine or the workpiece!
Do not initiate a control RESET during machining.

6. Start up the control unit and the drives again.

12.10.3 Switching on/off

Switching on: Program DistCtrlOn (possibly with additional parameters).

Switching off: Program DistCtrlOff.

12.11 Spline

Function:

Compared to the linear interpolation, spine interpolation achieves approximately the same surface or contour accuracy with fewer interpolation points, because curves are created between the points.

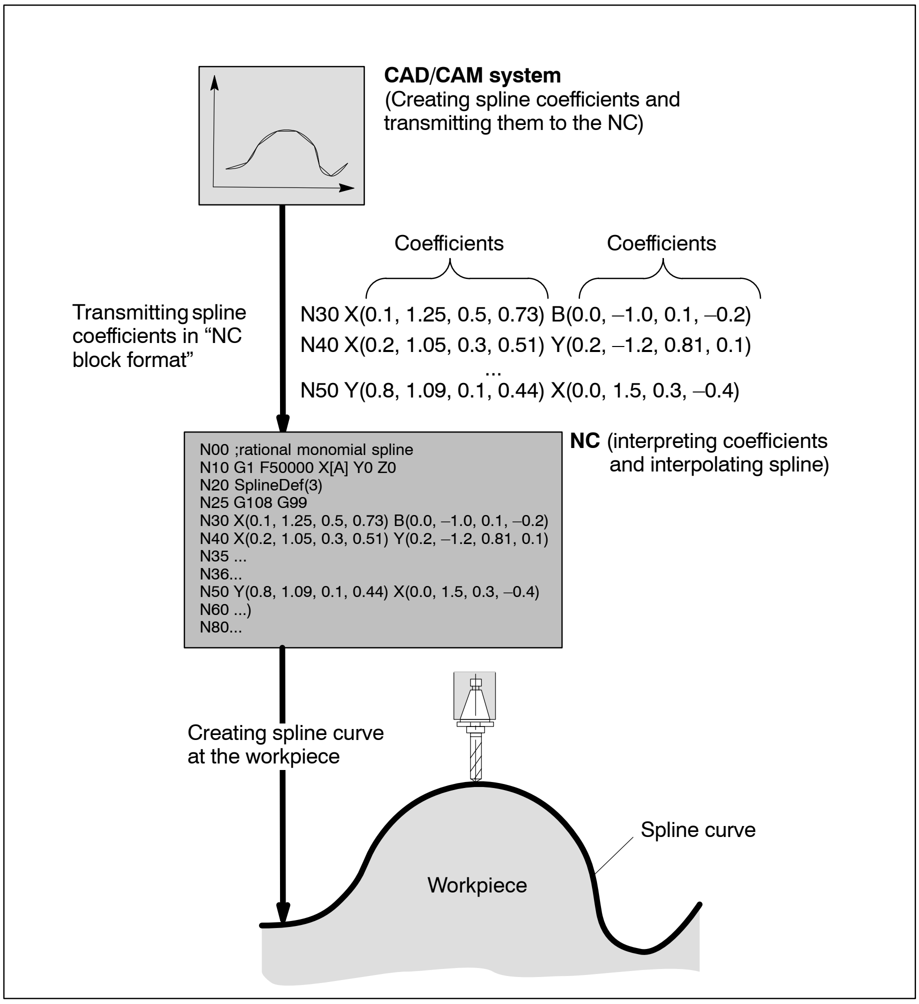
The spline function offers

- different processing options for spline data coming from the CAD/ CAM or digitizing system,
- different types of spline for practice-oriented requirements (e.g. free-form surface machining),
- parametrization of boundary conditions for application-oriented spline machining in the NC program.

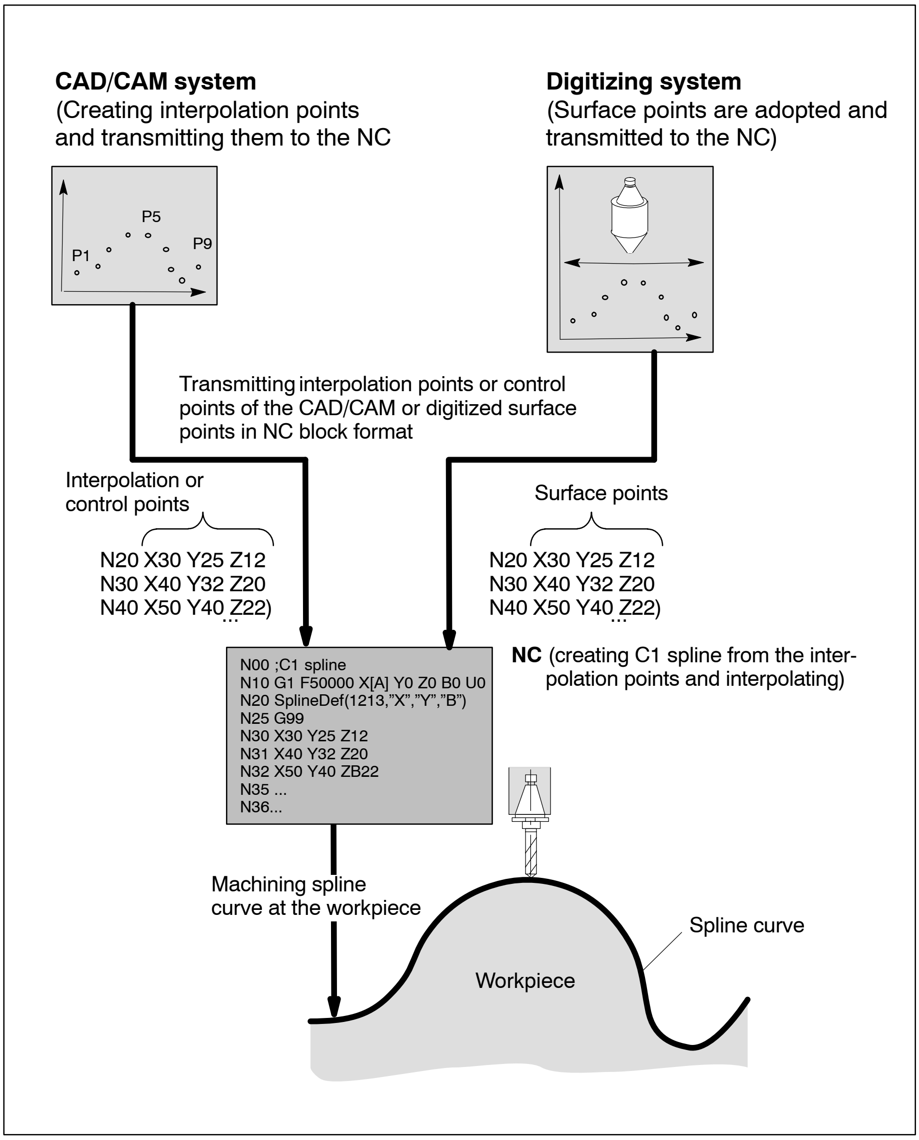
Origin of spline data

Point sequences can be specified as follows:

- manual programming in the NC program
- creating spline via CAD program with post-processor function and transmitting only the spline coefficients in the NC program.



- a CAD/CAM system creates interpolation points or control points for the NC program which are used to generate the NC spline curves (see figure below).
- digitizing: records the surface as individual points (see figure below).



Advantages:
- Softer contour lines and smoother surfaces on the workpiece.
- The number of points created by the CAD/CAM program or the digitizing process may be reduced, maintaining the required accuracy.
- The machine and its moving parts are exposed to **less** load by spline interpolation as opposed to the angular transition of linear interpolation, because the tool changes its direction less abruptly.

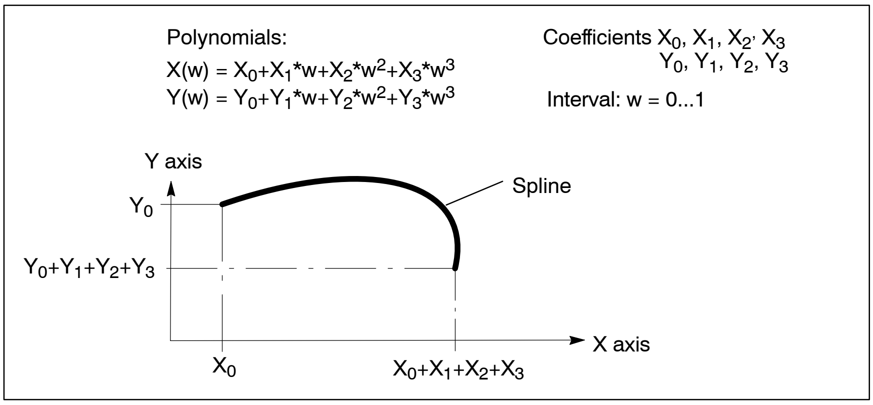
Spline

Types of spline (overview)

The NC supports the following spline programming:

- **Splines with coefficient programming**:

  Every spline can be represented as polynomial (e.g. parabola=square polynomial). Its coefficients determine the shape of the curve, the start and end point. The coefficients are provided by the CAD/CAM system, where they are programmed individually as NC blocks, transmitted to the NC and interpolated in the NC.



  Advantage: the data provided by the CAD/CAM system determine the spline

  Disadvantage: the CAD/CAM system must have a corresponding post-processor interface. Compensation in the NC via parametrization are not possible. Possible deviations only become apparent in the course of the machining process.

- **C^1-continuous cubic splines with point programming**:

  The interpolation points are programmed as NC blocks. From this, the NC generates cubic splines, the interpolation points of which have tangential transitions.
  The **tangential transitions** are calculated according to the Akima, Bessel or chordal method and have an effect on max. 4 interpolation points.

  Advantage: little computation demand in the NC.
  Look-ahead of only one block required (local spline).

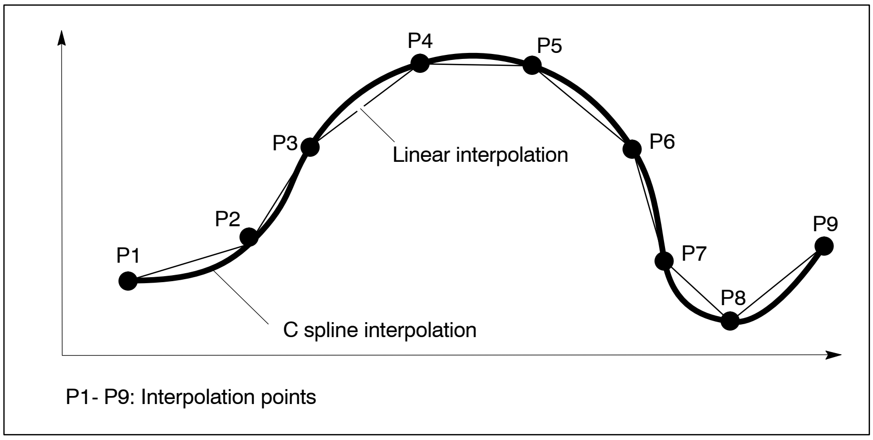
  Disadvantage: The transitions at the support points do not have a continuous bend; that is why jumps occur in the axis acceleration.

- **C^2-continuous cubic splines with point programming**:

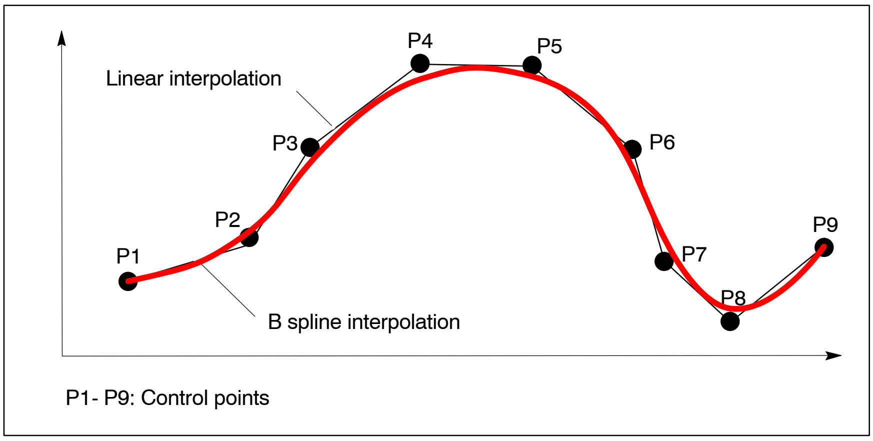
  The interpolation points are programmed as NC blocks. From this, the NC generates cubic splines, the interpolation points of which have transitions with a continuous bend.

  Advantage: no jumps in the axis acceleration.

  Disadvantage: high computation and memory demand in the NC. Great number of blocks in the look-ahead (global spline)



- **B splines with control point programming**:
  In case of B spline programming **(e.g. NURBS)** the spline curve calculated by the NC does not pass through the interpolation points but as close as possible to the control points programmed in the NC block ("the points attract the curve").
  This is used for programming the tool path for free-form surfaces.



Advantage:

- The curve bend is principally less pronounced than in C^2-continuous cubic splines.
- All curves defined by cone sections (ellipse, circle, parabola, hyperbola) can be programmed precisely.
- Less computation required in the NC than for C^2-continuous cubic splines.
- Fewer blocks in the look-ahead (local spline)

Disadvantage: As a rule, the control points are not approached precisely but only approximately.

Spline

12.11.1 Spline with coefficient programming

☞ **The coefficient programming of splines requires a corresponding CAD/CAM programming system capable of generating the coefficients in an NC program.**

Integral monomial splines with coefficient programming:
The spline is represented by a polynomial up to degree n=5. The n+1 coefficients of the polynomial are specified **directly** in the NC block.

$$\overbrace{\vec{r}\,(w)}^{\text{Vector}} = \overbrace{\sum_{j=0}^{n} \vec{r}_j\, w^j}^{\text{Polynomial}} \qquad w \in [0,w_e]$$

Example: Monomial splines (e.g. cubic spline: degree 3) with coefficient programming. It is assumed here that the channel axes X, Y, Z, A and B constitute the spline movement.

$$\overbrace{\vec{r}\,(w)}^{\text{Vector}} = \overbrace{\begin{pmatrix} X(w) \\ Y(w) \\ Z(w) \\ A(w) \\ B(w) \end{pmatrix}}^{\text{Channel axes}} = \overbrace{\underbrace{\begin{pmatrix} X_0 \\ Y_0 \\ Z_0 \\ A_0 \\ B_0 \end{pmatrix} + \begin{pmatrix} X_1 \\ Y_1 \\ Z_1 \\ A_1 \\ B_1 \end{pmatrix} \cdot w + \begin{pmatrix} X_2 \\ Y_2 \\ Z_2 \\ A_2 \\ B_2 \end{pmatrix} \cdot w^2 + \begin{pmatrix} X_3 \\ Y_3 \\ Z_3 \\ A_3 \\ B_3 \end{pmatrix} \cdot w^3}_{\text{3+1 coefficients}}}^{3^{rd}\text{ degree polynomial}}$$

Interval size w
$w \in [0,w_e]$

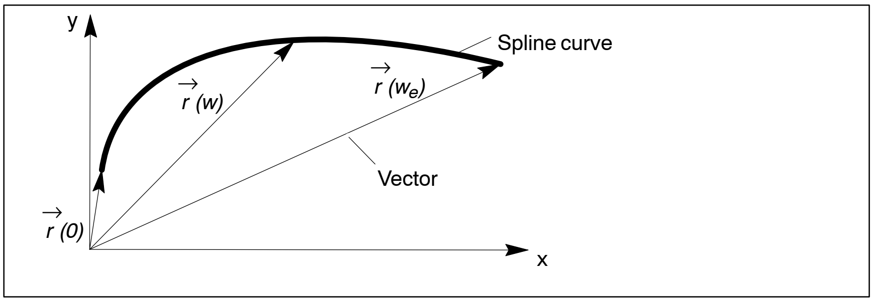
Example: Specification of coefficients in the NC program for channel axes X and B
G99 **X(0.1, 1.25, 0.5, 0.73) B(0.0, –1.0, 0.1, –0.2)**

For the channel axis X and B, this results in:
X(w)= $0.1+1.25w+ 0.5w^2+0.73w^3$
B(w)= $0.0–1.0w+ 0.1w^2–0.2w^3$

The coefficients are used to calculate the vectors creating the spline curve in the definition interval 0 through w_e, the components of which are the channel coordinates. The channel coordinates are used to calculate the axis positions of the physical axes within the NC, taking into account an active axis transformation, if required.

Spline

Example: Spline $\vec{r}(w)$ curve with the two channel coordinates x and y



Spline parameter length

The spline parameter length defines the length of the **definition interval** of **w**. The **definition interval** is programmed in the NC block via the rule **PL...**.

Example:

```
G99 X(0.1, 1.25, 0.5, 0.73) B(0.0, –1.0, 0.1, –0.2) PL0.6
```

Mixed programming

Within each NC block, coordinates may be programmed either as **spline** or as **linear segment**.

Application example:

N10 G1 X100 Y100 Z20 B10 U0 F1000
N20 SplineDef(3)
N30 G2 X200 Y100 Z30 I100 J0
N40 G99 **X(200,1,0.5,–0.1) Y(100,–1,0.2,0.2) B(10,1.0) Z60 U10** PL10
N50 **X(160,20) Y(310,4.0,0.2,–0.01) Z0**
N60 G1 X10 Y20 Z30 B40
N70 G99 Y(...) PL12

Explanation:

N20: The G99 syntax is assigned a cubic spline.

N40: Changing from path type helical to spline.
X, Y and B have coefficient programming and are considered as spline members in this block.
Z and U have end point programming and move on a linear path.
The spline parameter w passes through interval [0,10]. The spline members move correspondingly.

- $X(w)=200+1.0w+0.5w^2-0.1w^3$ End point: X(10)=160
- $Y(w)=100-1.0w+0.2w^2-0.2w^3$ End point: Y(10)=310
- $B(w)=10-1.0w+0.0w^2-0.0w^3$ End point: B(10)=20

N50: Second spline in the NC program. The spline parameter length is 10 (modal effect of PL from N40).
X and Y are spline members, Z moves on a linear path.

N60: Changing to path type linear

N70: Changing to path type spline. PL must be programmed again.

Spline

Rational monomial splines

If the polynomial of the integral monomial spline is extended using a common **denominator polynomial**, the result is a rational monomial spline.

$$\underbrace{\vec{r}(w)}_{\text{Vector}} = \frac{\overbrace{\sum_{j=0}^{n} \vec{r_j} \cdot w^j}^{\text{Polynomial}}}{\underbrace{\sum_{j=0}^{n} g_j \cdot w^j}_{\text{Denominator polynomial}}} \qquad w \in [0,w_e]$$

The denominator polynomial is programmed by coefficient programming in the NC via rule DN(...).

Example: Specification of coefficients in the NC program for channel axes X and B

```
G99 X(0.1, 1.25, 0.5, 0.73) B(0.0, -1.0, 0.1, -0.2) DN(1.0,0.0,1.0)
```

For the channel axis, this results in:

$$X(w)= \frac{0.1+1.25w+ 0.5w^2+0.73w^3}{1+w^2}$$

and for

$$B(w)= \frac{0.0–1.0w+ 0.1w^2–0.2w^3}{1+w^2}$$

Application example:

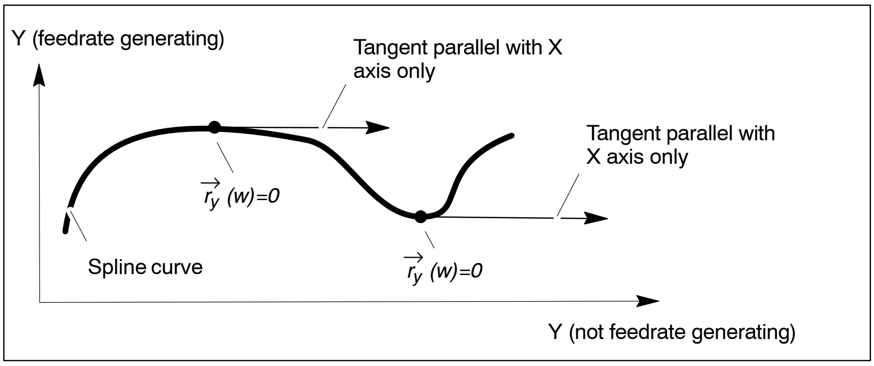
```
N00 ;ellipse as rational monomial spline
0001 A=100
0002 B=50
N10 G1 F50000 X[A] Y0 Z0
N20 SplineDef(2)
N25 G108 G99
N30 X(A,0,-A) Y(0,2*B) DN(1,0,1) PL1
N40 X(0,-2*A,A) Y(2*B,-2*B) DN(2,-2,1)
N50 X(-A,0,A) Y(0,-2*B) DN(1,0,1)
N60 X(0,2*A,-A) Y(-2*B,2*B) DN(2,-2,1)
N80 G1
```

☞ **All cone section (ellipse,..) as well as rational Bezier and B splines (NURBS) can be represented by rational monomial splines**.

In addition, the splines have the following properties:

- A C^0 continuity exists when the starting point of the spline is identical with the end point of the previous block. Inaccurate coefficient programming may result in differences between starting and end point. A tolerance value e_0 can be set in MP 8007 00010 for the block transition.
- The path speed on the spline curve is approximated internally using a "path length function" on the basis of the relationship between the path length and the spline parameter w.
  In MP 8007 00020, it is possible to set a tolerance ε_V in this context as **max. relative path velocity error**. As accuracy requirements increase, the value entered has to be smaller. However, this also leads to a rising NC block cycle time. If the specified accuracy is not reached, the NC generates a warning, but machining is continued. If MP 8007 00020 is set to 100 %, the path length function is approximated by a line.
- Even axes involved in the spline curve can be removed from the feedrate calculation. However, the spline curve must not include a path point in which all feedrate generating coordinates/axes have a so-called zero tangent (i.e. no shift in axis direction). In this case, a runtime error will be generated.

  Example: the space curve in the following figure is considered as curve of the axes X and Y. The spline $\vec{r}(w)$ has two path points with a tangent along the X axis. If X is not feedrate generating, this is exactly where the error occurs.



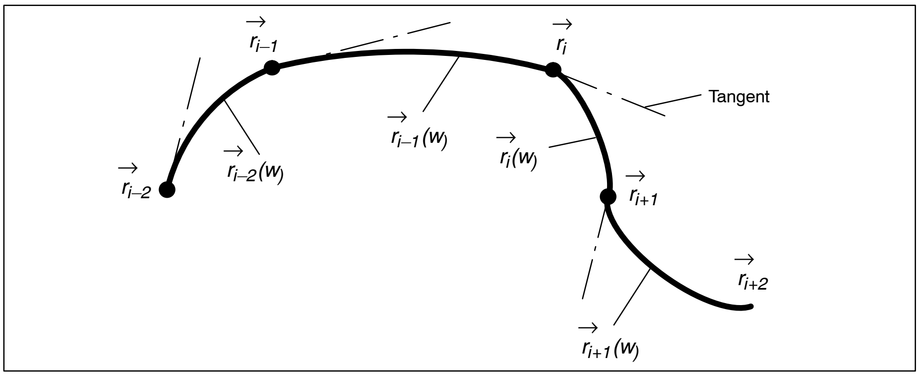
12.11.2 Spline with point programming and tangential transitions (C^{1}-continuous cubic splines)

Function

The splines are calculated from the specified points (n curve points) and the tangents of these points. The points are specified in the NC program.

The tangents in the points $\vec{r}_i$ (i=2,...n–1) are calculated from three consecutive points $\vec{r}_{i-1}$, $\vec{r}_i$, $\vec{r}_{i+1}$.

The tangents at the points $\vec{r}_1$ and $\vec{r}_n$ result from the start and end conditions (see next page).



The splines have the following properties:

- On the basis of n points and a start and end condition of the spline sequence n–1 splines are created. A spline $\vec{r}_i(w)$ connects the points $\vec{r}_i$ and $\vec{r}_{i+1}$.
- They are C^1-continuous (tangential) at the transition points $\vec{r}_i$.
- They are local, i.e. the modification of a point $\vec{r}_i$ influences a restricted number of neighboring splines only. In this case, the splines $\vec{r}_{i-2}(w)$, $\vec{r}_{i-1}(w)$, $\vec{r}_i(w)$, $\vec{r}_{i+1}(w)$ would be affected. In the reverse case, a spline is only determined by the points $\vec{r}_{i-1}$, $\vec{r}_i$, $\vec{r}_{i+1}$, $\vec{r}_{i+2}$.

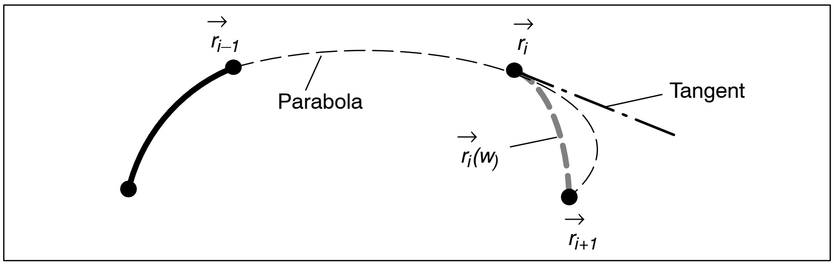
Tangent calculation:

The tangential transition is set via parametrization (for information on spline ID, see Section 12.11.5). The following methods are available for this purpose:

- **Bessel method**

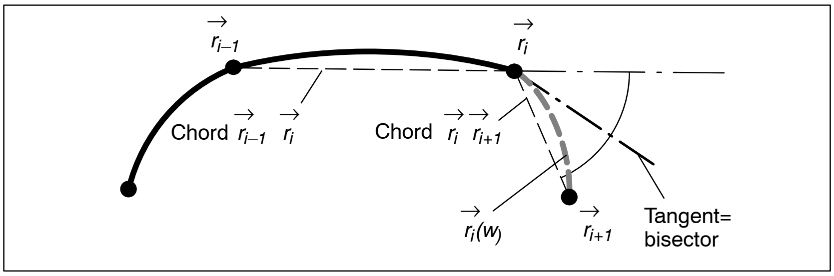
  The programmed points $\vec{r}_{i-1}$, $\vec{r}_i$, $\vec{r}_{i+1}$ determine a parabola (2nd degree spline). This results in the tangent in $\vec{r}_i$.

  Length and direction of the tangent depend on the selected spline parametrization (see page 12–69).



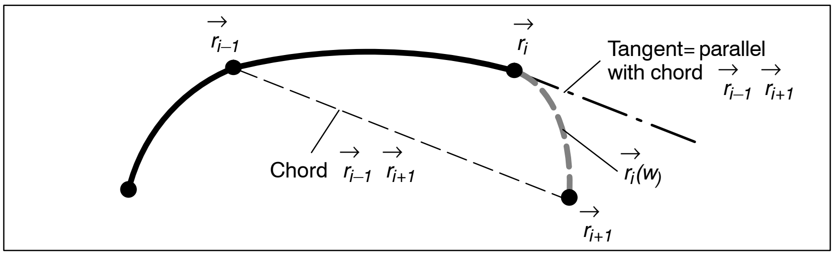
- **Akima method**

  The tangent in $\vec{r}_i$ points in the direction of the bisector of the two chords $\vec{r}_{i-1}\,\vec{r}_i$ and $\vec{r}_i\,\vec{r}_{i+1}$. The tangent length depends on the selected spline parametrization (see page 12–69).



- **Chordal method**

  The tangent in $\vec{r}_i$ points in direction of the chord $\vec{r}_{i-1}\,\vec{r}_{i+1}$.



Start and end condition

A start and end condition is programmed to determine the first and last spline segment. Programming is performed with the parameters:

- “Start Boundary Condition” **SBC**(<Typ>[,<Values>]) and
- “End Boundary Condition” **EBC**(<Typ>[,<Values>]).

For the start and end conditions, it is possible to select as follows:

- Type =1: specification of the tangent direction at the starting/end point of the spline sequence. Only the relationship between the values of the spline members is of significance. The tangent length is calculated internally (for chordal parametrization, the tangent length = 1).
- Type =2: specification of the second derivation $\vec{r}_i''$ following w at the starting/end point of the spline sequence.

Spline

- Type =11: the first/last spline starts tangentially to the linear block of the preceding/subsequent block.

A value has to be specified in the <Values> list for each spline.

SBC and EBC may also be programmed within a spline sequence (G99,..,G1). This divides the original sequence into two partial sequences to specifically generate corners within a spline curve.

Without programmed start and end condition, the natural boundary condition SBC(2,0,...,0) or EBC(2,0,...,0) is used.

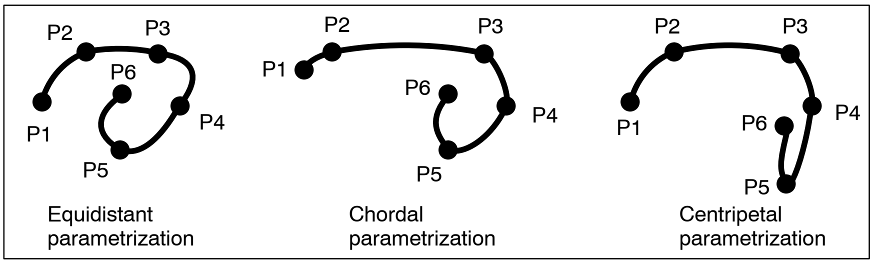
Spline parameter length

The spline parameter length takes into account the point size and spline curve bend. Depending on the application, 3 different parametrizations are allowed:

- **Equidistant** parametrization (w=1) :
  This parametrization is suitable for identical spline point intervals. If the spline interpolation points are located at varying distances, there is an increased tendency, in particular in case of bend continuity, that there will be “bulges” in the curve.
- **Chordal** parametrization $w = |\vec{r}_{i+1} - \vec{r}_i|$:
  If the spline interpolation points are located at varying distances, the distance and the direction of the interpolation points will be taken into account.
- **Centripetal** parametrization $w = \left|\sqrt{(\vec{r}_{i+1} - \vec{r}_i)}\right|$:
  This parametrization is used if the spline interpolation points are located at varying distances and the spline curves have strong bends.

The curve slope will be smoother and display fewer unexpected bends, the better the position and distances of the interpolation points are taken into account. Therefore, a combination of the above mentioned parametrization methods may be required depending on the interpolation points specified.

The spline parameter length can also be specified manually via parameter PL<w> instead of using the parametrization calculated by the NC.

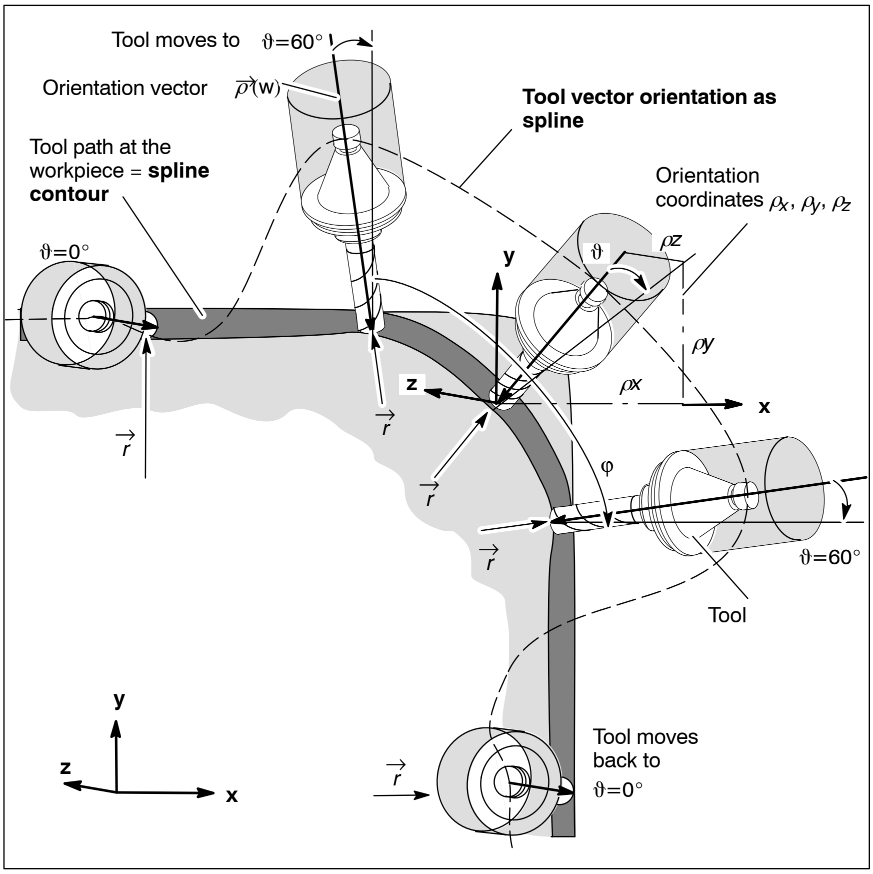


Tool orientation movement as C^1-continuous cubic spline

Not only the tool tip can move on a spline but the **tool orientation** as well, if a rotation-symmetrical tool is involved which is programmed together with a corresponding 5 axis transformation. A distinction is made between vector orientation and linear orientation (see page 5–56 ff.).

☞ **The orientation of non-rotation-symmetrical tools (tensor orientation in connection with the corresponding 6 axis transformation) cannot be programmed as spline.**

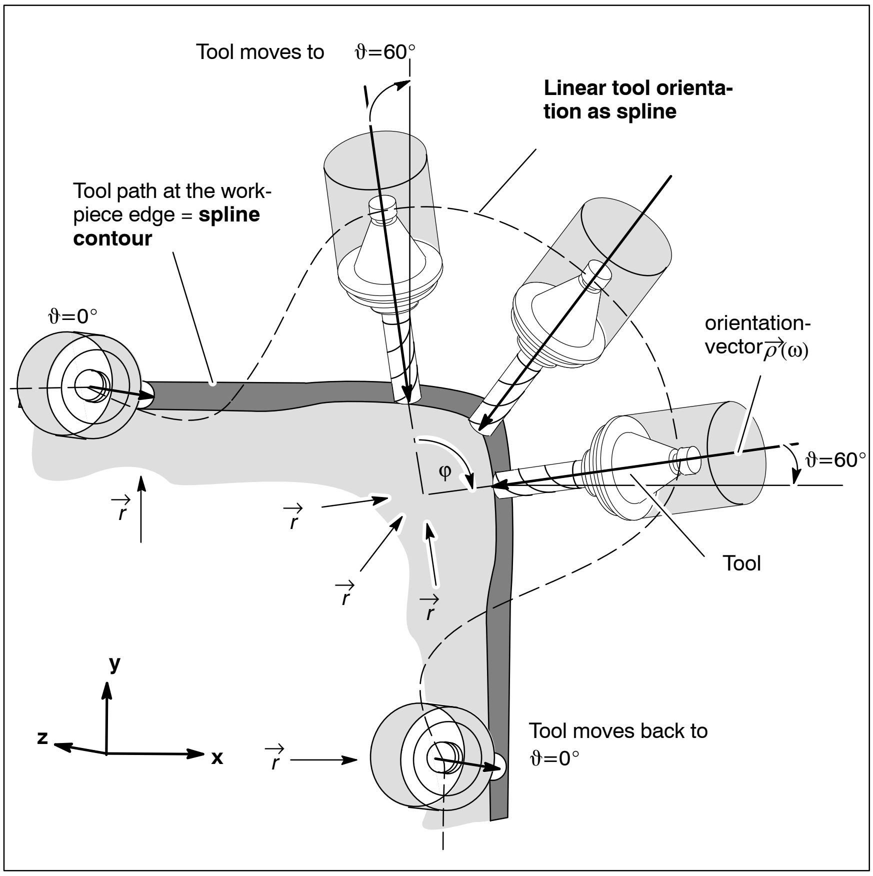
- **Vector orientation as C^1-continuous cubic spline:**
  In case of vector orientation as C^1-continuous cubic spline, a spline is created in each NC block in the six coordinates x, y, z, ρ_x, ρ_y, ρ_z using a common parametrization (equidistant, chordal, centripetal). The tangent method (Bessel, Akima, chordal method) stated in the “SplineDef” function is also used for the orientation vector $\vec{\rho}(w)$. At the block transitions, both the path movement (x, y, z) and the orientation vector are C^1-continuous.



Spline

- **Linear orientation movement as C^1-continuous cubic spline**

  For linear orientation movement, splines are created in the five coordinates x, y, z, phi, theta. The splines are mathematically equivalent to splines for an axis movement in three linear and two rotary axes (see page 12–70).



In addition, the splines have the following properties:

- The $\vec{r}$ movement ($\vec{r} = (x,y,z)$) is dependent on the $\vec{\rho}$ movement and vice versa. This means that e.g. in blocks where only $\vec{\rho}$ is programmed, an additional $\vec{r}$ movement can occur if not all $\vec{r}$ spline-determining points $\vec{r}_{i-1}$,..., $\vec{r}_{i+2}$ are identical.
- Special boundary conditions for $\vec{\rho}$ cannot be programmed.
- The spline parameter $w \in [0,w_e]$ is applicable to $\vec{r}_i(w)$ as well as to $\vec{\rho}_i(w)$.

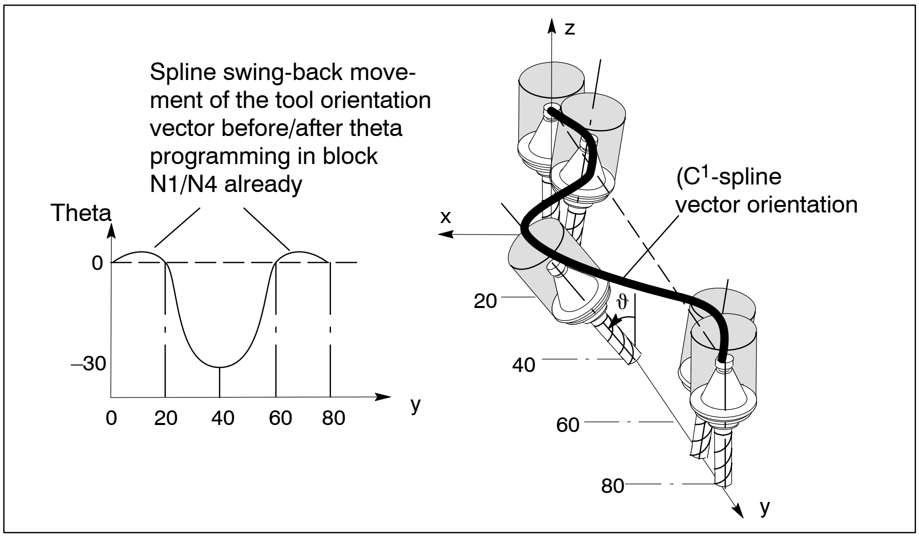
Examples

```
N0 SplineDef(1213,"x","y","z","O")
;Linear coordinates and orientation are spline members (C1)
;5 axis transformation with vector orientation
N0 Coord(1)

;alternatively N0 Coord(3); 5 axis transformation with linear orientation
N0 x0 y0 z0 phi0 theta0

N0 G99 G8
N1 y20
N2 y40 theta30
N3 y60 theta0
N4 y80
N5 ...
```

Explanation:

N1: the theta orientation movement of the tool starts in this block already

N4: the theta orientation movement of the tool only ends in this block



12.11.3 Spline with point programming and continuous bend transitions (C^2-continuous cubic splines)

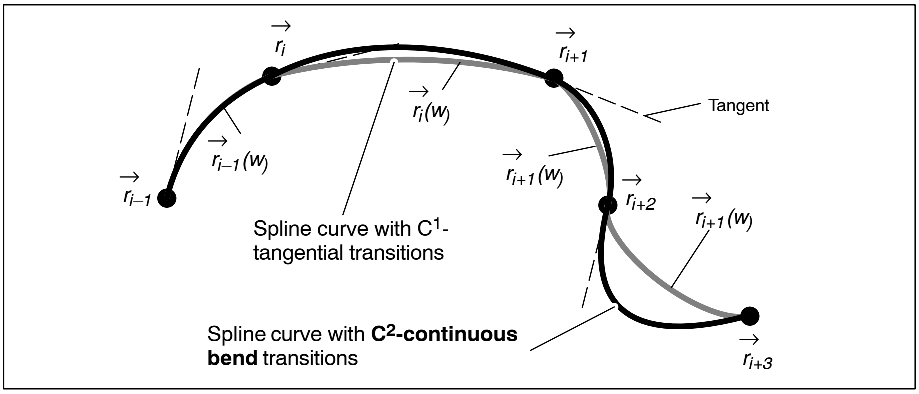
Function

The splines are calculated using the specified points $\vec{r}_i$ with i=1,..,n and a start and end condition of the spline sequence.

The splines have the following properties:

- On the basis of n points and a start and end condition of the spline sequence n–1 splines are created. A spline $\vec{r}_i(w),\ w \in [0, w_e]$ connects the points $\vec{r}_i$ and $\vec{r}_{i+1}$.
- They are C^2-continuous (continuous bend) at the transition points $\vec{r}_i$.
- They are global, i.e. the modification of a point $\vec{r}_i$ influences all splines of the spline sequence. However, the influence decreases strongly with increasing distance from the point $\vec{r}_i$.

The calculation of the splines requires a distinctly higher look-ahead range than the C^1-continuous cubic splines because of the global spline property.



Start and end condition

In addition to the boundary conditions type 1, 2 and 11, which are also possible for C^1-continuous splines (see page 12–68), there are two further conditions:

- Type =3: the **De-Boor boundary condition** connects the two derivations with each other at the first two interpolation points or at the last two interpolation points.
  SBC(3,<v>) or EBC(3,<v>) v is usually =1
- Type =4: the **periodic boundary condition** is used for closed curves (e.g. ellipse). In this case, the last and the first point of the spline sequence have to be identical.
  The periodic boundary condition cannot occur in combination with another boundary condition, i.e. with SBC(4) at the start, EBC(4) at the end is mandatory.
  The spline sequence of SBC(4) through EBC(4) must be located in the look-ahead range of the NC. If that is not the case, a runtime error will be generated.

SBC and EBC may also be programmed within a spline sequence (G99,..,G1). This divides the original sequence into two partial sequences to specifically generate corners within a spline curve.

Without programmed start and end condition, the natural boundary condition SBC(2,0,...,0) or EBC(2,0,...,0) is used.

Spline parameter length

See C^1-continuous cubic spline (see page 12–70).

Tool orientation movement as C^2-cubic spline

The same preconditions as for the C^1-continuous cubic spline are applicable (see page 12–70).
The vector orientation can also be programmed as C^2-continuous cubic spline. In this context, a distinction is made between two options which can be determined in MP 8007 00099 [3]:

- **Vector orientation (option 1) as common C^2-cubic spline Spline with the space spline of the linear coordinates x, y, z.**

  This option uses a spline in the six coordinates x, y, z, ρ_x, ρ_y, ρ_z with a joint parametrization. In this case, the spline vector orientation has the following properties:

  - The $\vec{r}$ movement ($\vec{r} = (x, y, z)$) is dependent on the $\vec{\rho}$ movement and vice versa. This means that e.g. in blocks where only $\vec{\rho}$ is programmed, an additional $\vec{r}$ movement occurs (see figure on 12–76).
  - Special boundary conditions for the orientation vector $\vec{\rho}$ cannot be programmed.
  - The spline parameter $w \in [0, w_e]$ is applicable to $\vec{r}_i(w)$ as well as to $\vec{\rho}(w)$.

- **Vector orientation (option 2) as separate C^2-cubic spline, independent of the space spline of the linear coordinates x, y, z.**

  This option requires a large number of blocks in the look-ahead range. The number of blocks (MP 706000120[3]) should be approximately three times the number of blocks provided for option 1. Option 2 has the following special properties:

  - The $\vec{r}$ movement ($\vec{r} = (x, y, z)$) is independent of the $\vec{\rho}$ movement. A $\vec{\rho}$ movement only occurs in blocks with start orientation unequal to end orientation (see figure on page 12–76).
  - $\vec{\rho}(w)$, $w \in [0, \tilde{w}_e]$ always has an own chordal parametrization, and $\tilde{w}_e$ is the angle between the start and the end orientation vector of the block.
    This $\tilde{w}_e$ of spline $\vec{\rho}_i(w)$ differs from w_e of spline $\vec{r}_i(w)$.
  - Special boundary conditions for the orientation vector $\vec{\rho}$ cannot be programmed.

Examples

If the program is executed using the setting MP 8007 00099[3] = 1, this results in the theta-y path shown in the following figure.

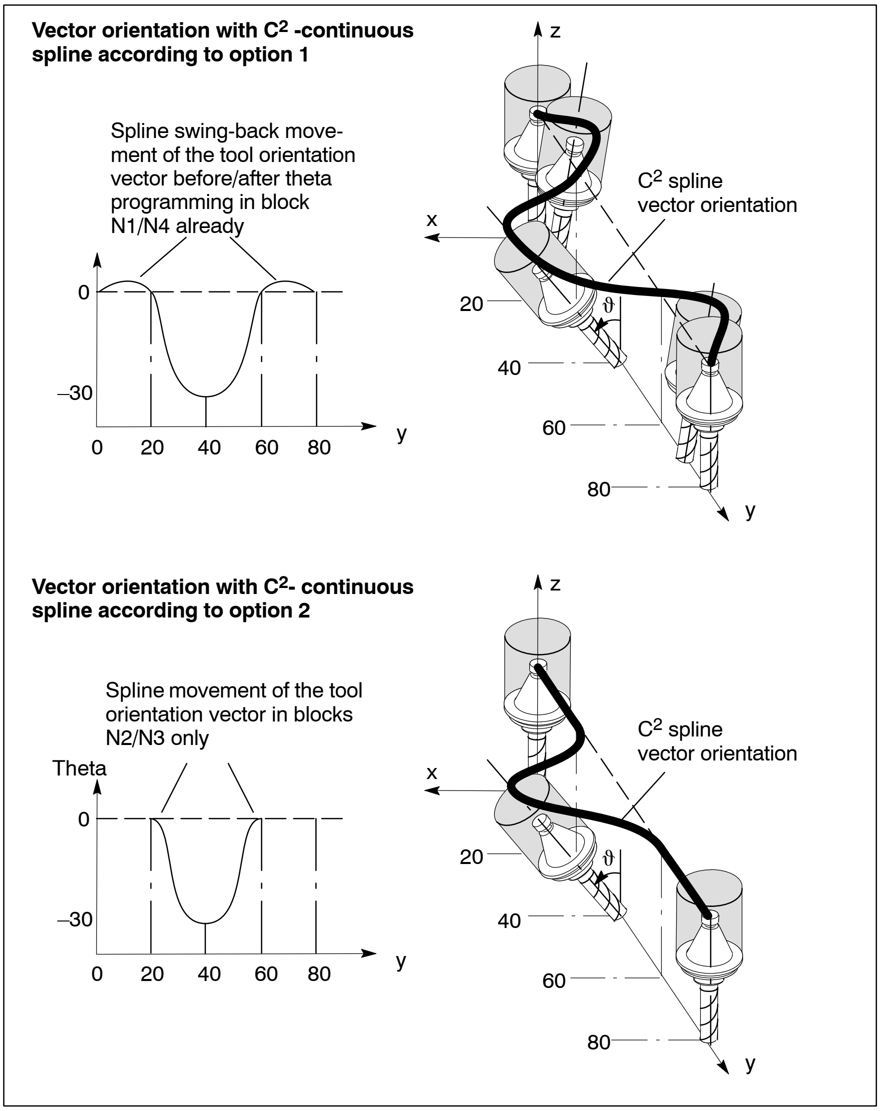
Vector orientation and path movement as joint spline

```
;(Option 1)
N0 SplineDef(2203,"x","y","z","O")
;Linear coordinates and orientation are spline members (C2)
N0 Coord(1);  5 axis transformation with vector orientation
N0 x0 y0 z0 phi0 theta0
N0 G99 G8
N1 y20
;the theta orientation movement of the tool already starts in N1
N2 y40 theta30
N3 y60 theta0
N4 y80
;the theta orientation movement of the tool only ends in N4
N5 ...
```

If the program is executed using the setting MP 8007 00099[3] = 2, this results in the theta-y path shown in the following figure. A $\vec{\rho}$ movement is also generated in the blocks N1 and N4, in which the orientation vector was not programmed.

```
;Vector orientation as separate spline
;(Option 2)
N0 SplineDef(2203,"x","y","z","O")
;Linear coordinates and orientation are spline members (C2)
N0 Coord(1);  5 axis transformation with vector orientation
N0 x0 y0 z0 phi0 theta0
N0 G99 G8
N1 y20
N2 y40 theta30
;the theta orientation movement of the tool starts in N2
N3 y60 theta0
;the theta orientation movement of the tool ends in N3
N4 y80
N5 ...
```

Spline



- **Linear orientation movement as C^2-continuous cubic spline**
  - For the linear orientation movement as C^2-continuous cubic spline the same is applicable as for linear orientation movement as C^1-continuous cubic spline (see page 12–71 ff.).
  - In case of long spline sequences (number of blocks between SBC and EBC), which are not fully located in the look-ahead range (see page 12–80) of the NC, an appropriate method is used to create two or more partial spline sequences which are in turn executed C^2-continuously. Thus you can use splines for contours of any length (free-form surface machining).
    The number of partial sequences should be kept as low as possible, i.e. MP 800700099[2] should be set as high as possible.
  - In case of periodic boundary conditions, sequence interruption is not possible. The entire spline sequence must be located in the look-ahead range in this case.

12.11.4 Spline with control point programming (B spline)

Function

B splines are generated by so-called “control points”. In contrast to the C^1/C^2 splines, the spline curve of B splines does not pass specified interpolation points but “approximates” the specified “control points” (see figure on page 12–62).

☞ **NURBS (NonUniform Rational B Spline) are a special variation of the B splines. They are dealt with here under the designation “B spline”. For information on the special properties of the NURBS, please refer to the relevant literature.**

B splines have the following properties:

- From n points, i.e. from “n–1” NC blocks, “n–p” splines are generated, where p is the degree of the B splines (see example below).
- In general, **smaller** bends occur as compared to the global cubic splines because B splines do not run through the control points with the exception of special cases. They do not tend to overswing and nevertheless have a continuous bend at the transition.
- By programming point weights using the PW function, it is possible to modify splines in the area surrounding a point. A point weight >1 ”attracts” the spline towards the point, a point weight <1 ”pushes” it away from the point. This is referred to as rational B spline or NURBS.
- B splines have a local effect, i.e. the modification of a point influences a number of splines dependent on the degree of the B spline.
- Programming of start and end conditions is not possible (open or periodic node vectors are not supported).

Control points

For a B spline curve with degree p, at least p+1 different control points have to be defined. Within the spline sequence, there have to be at least p NC blocks (with end point unequal to start point).

Double control points are not supported, i.e. they are filtered out by the NC. The result is a spline without traversing movement.

Spline parameter length

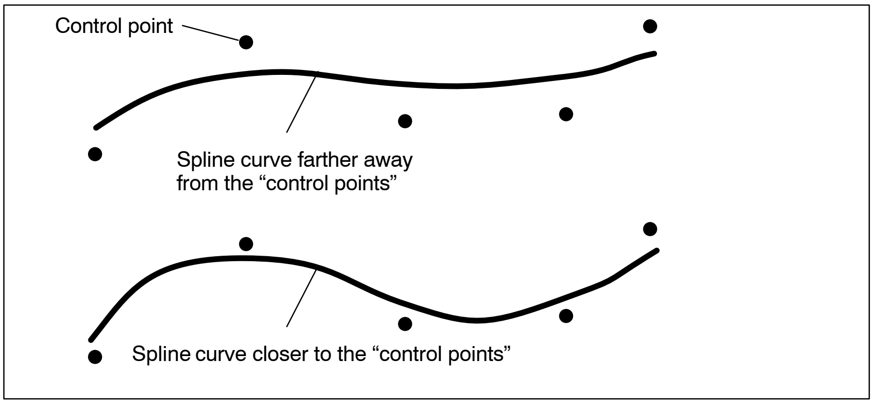
See C^1-continuous cubic spline (see page 12–70). For practical applications, only the equidistant parametrization (uniform B spline) plays a role.

A non-uniform B spline is obtained when the value of the PL spline parameter length is set to a value unequal to 1 in individual blocks.

Point weights

The control points can be given a **P**oint **W**eight using the syntax **PW ...**. By different weighting it is thus possible to influence the path of the spline curve at every control point (depending on the degree of the spline even in the area surrounding the control point). A point weight of 0 is not permitted.

Spline



Example:

3rd degree B spline:

```
;Selection 3rd degree B spline:
N0 SplineDef(3103,"x","y","z","O")
;5 axis transformation with vector orientation:
N0 Coord(1)
N0 x0 y0 z0 phi0 theta0
N0 G99 G108
N1 x10y20z11 O(0.1,0,1.0)
N2 x.. y.. z.. O(..)
;1st spline:
N3 x.. y.. z.. O(..)
;2nd spline, point weight 2.3:
N4 x.. y.. z.. O(..) PW2.3
...  ....
...  ...
;18th spline:
N20 x,.. y.. z.. O(..)
N21 G1
```

Explanation:

N0: Selection of a 3rd degree B spline

N1: Starting point of the spline curve tangentially to the connection line
start point - end point

N1-N2: Non-traversing block

N1-N20: Determining 21 control points

N3-N20: 18 splines which are executed

N20: End point of the spline curve tangentially to the connection line
start point - end point

Spline

12.11.5 Programming spline

All usable splines can be parametrized in the program using: SplineDef(<Id>,<Members>))

The table gives an overview of the available options:

<Id>				**Meaning**	**Recommended combinations**
Spline type	**Parametrization** 1= equidistant 2=chordal 3=centripetal	**Tangent calculation** 1=Bessel 2=Akima 3=chord	**De-grees**		Example: cubic spline SplineDef(<Id>, <Members>)
0	0	0	1, .., 5	Monomial spline with coefficient programming	SplineDef(3)
1	1, 2, 3	1, 2, 3	3	C^1-continuous local cubic spline from specified points with different tangent calculation at the block transitions	SplineDef(1213,...)
2	1, 2, 3	0	3	C^2-continuous global cubic spline from specified points.	SplineDef(2203,...)
3	1	0	1, .., 5	B spline and NURBS from specified control points and weights	SplineDef(3103,...)

Restrictions:

The following functions **cannot** be programmed with splines:

- Tensor orientation: this concerns the tensor syntax Ox(..), Oy(..) and Oz(..) as well as the Eulerian angles phi, theta, psi.
- Polar coordinate interpolation
- G41/G42 path compensation
- Canceling the distance to go
- Punching and nibbling with contour calculation
- Circle or helix with tangential entry to previous spline element
- Chamfers and radiuses
- Tangential tool guidance
- Precision programming
- Area control

Relevant MACODA parameters (MP):

7060 00120[3]: Maximum number of blocks for look-ahead function

Recommended minimum values:

for **spline type 2**:

20 blocks: without the "Vector orientation as spline" function

20 blocks: for option 1 (spline for vector orientation together with space spline)

60 blocks: for option 2 (separate spline for vector orientation)

for **all other spline types**:

10 blocks

7060 00130[3]: Percentage splitting of look-ahead

8007 00010: Tolerance value ε_0 in % for the C^0-continuity for splines with coefficient programming. If the starting point is not identical with the end point of the preceding spline, a deviation exceeding the set tolerance value ε_0 will generate a runtime error.

The value selected should be as low as possible (depends on the accuracy of the programming system), because axis jumps may occur at the segment transitions otherwise.

8007 00020: Maximum admissible relative path speed error $\varepsilon_{v.}$ in %.

The ratio (inverse path length function) of "spline parameter length w" and "path length of the curve" is approached within the NC by means of splines. The path speed error reflects the accuracy of this approximation. The value selected should not be smaller than necessary because the NC cycle time rises with increasing accuracy demands. If the specified accuracy is not reached with eight approximating splines, the NC generates a warning. The machining process is continued.

If the MACODA parameter is set to 100%, the path length function w(s) is approximated by a line.

8007 00099: Spline configuration (only relevant for C^2-continuous cubic spline)

[1] Degree of the spline between two partial sequences:

- 3: cubic spline (C^1-continuous at the starting point only)
- 4: square spline (C^2-continuous relative to the preceding spline)

[2] Overlapping (number of NC blocks) of two partial sequences

[3] Spline option for vector orientation:

- 1: Spline for vector orientation together with space spline
- 2: Separate spline for vector orientation:

[4]...[8] not relevant

Spline

Relevant G functions:

- **Spline with coefficient programming**

  G99 with modal parameters:

  <CoordName >(..)
  : Coefficient programming of individual **coordinates** (CoordName) with their polynomial coefficients.

  O1(..), O2(..), =3(..)
  : Coefficient programming of the **vector orientation** (working range coordinate programming (Coord(..)) has to be activated first)

  DN(..)
  : **Denominator polynomial** in case of coefficient programming for the description of rational monomial splines (Bezier spline, rational B spline (NURBS), all cone sections).

  PL(..)
  : Spline parameter length

- **C^1-continuous cubic splines**

  G99 with modal parameters:

  <CoordName>
  : Programming of coordinates; are only taken into account for calculation of the splines if they are listed under the <Members> parameter in the SplineDEF(..) NC function. All others are carried along linearly.

  <Axes>
  : Programming of axes; see above <CoordName>

  <Orientation coordinates>
  : Programming of vector orientation (5 axis transformation) as spline

  SBC(..)
  : 3 possible boundary conditions for the **starting point** of a spline sequence

  EBC(..)
  : 3 possible boundary conditions for the **end point** of a spline sequence

  PL(..)
  : Spline parameter length

- **C^2-continuous cubic splines:**

  G99 with modal parameters:

  <CoordName>
  : Programming of coordinates; are only taken into account for calculation of the splines if they are listed under the <Members> parameter in the SplineDEF(..) NC function. All others are carried along linearly.

  <Axes>
  : Programming of axes; see above <CoordName>

  <Orientation coordinates>
  : Programming of vector orientation (5 axis transformation) as spline

  SBC(..)
  : 5 possible boundary conditions for the **starting point** of a spline sequence

  EBC(..)
  : 5 possible boundary conditions for the **end point** of a spline sequence

PL(..) Spline parameter length

- **Rational Bezier splines, B splines (NURBS) and cone sections:**
  G99 with the modal parameters:
  <CoordName>
  Programming of coordinates; are only taken into account for calculation of the splines if they are listed under the <Members> parameter in the SplineDEF(..) NC function. All others are carried along linearly.
  <Axes> Programming of axes; see above <CoordName>
  <Orientation coordinates>
  Programming of vector orientation (5 axis transformation) as spline
  PL(..) Spline parameter length
  PW(..) Providing a point weight to control points to either attract the spline curve to the control points or to push it away.

Relevant NC functions:

SplineDef(<Id>,<Members>)
Definition of the spline type

The following applies:

<Id>: Spline type, see table on page 12–79

<Members>: Determines the coordinate or axis names which participate in the spline movement. Coordinates/axes which are not listed can only be moved linearly.
Specifying <Members> is not applicable to type 0.
Default: 0

12.11.6 Assigning

1. Configure the parameters specified under ”Relevant MACODA parameters” according to your application.

12.11.7 Activating

Precondition:

- Before you use G99 in the NC program for the first time, the initialization SplineDef has to be programmed or the initialization stored in the MACODA init string has to be taken over.

1. G99 activates the path type “spline”

12.11.8 Deactivating

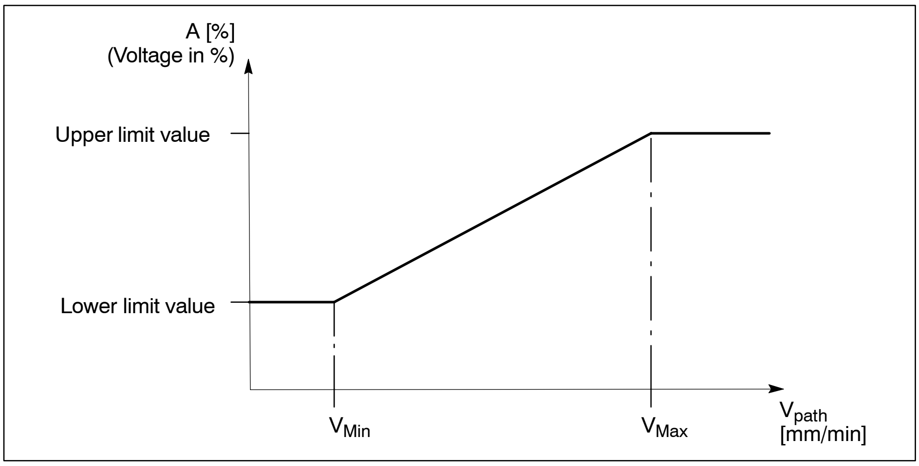
- The path type ”spline” is deactivated by programming a different path type, e.g. using G1, G2, G3 etc. The corresponding function has to be applied.

12.12 Path-speed-dependent laser power control

Function

The function controls the power of a laser, in dependence on the current actual value of the feedrate (V_{path}). For this purpose, a corresponding voltage value is output on an analog output, as defined in MP 4075 00104.

Below a certain path speed V_{path}, the laser power is limited to an adjustable minimum value. Over a certain path speed V_{path}, the laser power is maintained at an adjustable maximum value.
Between V_{Min} and V_{Max} the laser power is adapted in dependence on the path speed, see figure:



The voltage output value is specified in %.
100% output value corresponds to 10 volts at the analog output.

The effective path speed V_{path} results from the speeds of the selected coordinates:

- by selecting the active plane (APL) or the active space (ASP). In this context, a possibly active axis transformation or coordinate transformation (inclined plane) is taken into account.
- by direct selection of coordinates in the part program:
  - no axis transformation active:
    all pseudo coordinates (axes) of a channel can be selected.
  - an axis transformation is active:
    working range and pseudo coordinates can be selected. The selected working range coordinates are linked to the transformation active at this point in time.

Restrictions:

- The available analog outputs limit the number of channels which the function can use.
- When the "Inclined plane" function is used, it is only possible to use the "active space" with PL(ASP) for coordinate selection to generate the speed V_{path} .
- The operation with axes and coordinates is supported with active 5 axis transformation.
- When an error occurs (runtime error, diagnostics class 1 error), when "Drive under control" (no FG (enable), drive off) is deactivated for a drive involved in the path and in case of "Feed hold", no laser voltage is output.

Relevant MACODA parameters (MP):

4075 00104: Assignment of analog outputs
701: Channel 1
..
712: Channel 12

7050 00810: Laser power control: power limit values in %

7050 00815: Laser power control: speed limit values in mm/min

7050 00820: Laser power control: coordinate selection in connection with PL(..)

Relevant G functions:

LFPON **Starts** the path-speed-dependent laser power control.
If programmed alone, the configuration data programmed via MACODA will take effect. Additional parameters can be programmed optionally.

LFP **Parametrizes** the active laser power control via part program.

Parameters for LFPON and LFP:

LL([<%Voltage>], [<VMin>])
Lower power limit: the voltage value determined here is output below the specified path speed.

UL([<%Voltage>], [<VMax>])
Upper power limit: the voltage value determined here is output over the specified path speed.

<%Voltage> 0% .. 100%: corresponds to 0 .. 10 Volt

<VMin> Lower corner value of the path speed in mm/min or inch/min

<VMax> Upper corner value of the path speed in mm/min or inch/min

Path-speed-dependent laser power control

PL(<Plane designation>)
Coordinate selection for the path speed calculation via plane selection

<Plane designation>	"APL":	Current plane (G17, G18, G20)
	"ASP":	Current space
	"MCD":	MACODA values

CD(<Coordinate 1>, [<Coordinate 2>], ... , [<Coordinate n>])
Coordinate selection for the path speed calculation directly via logical name

<Coordinate x>	x = 1..n Logical name of the working range coordinates or pseudo coordinates involved (axes)

LFPOFF	**Ends** the path-speed-dependent laser power control.
LPCOFF	alternatively to LFPOFF

Relevant SERCOS parameters:

S-0-0040	Speed actual value If S-0-0040 is not included in the drive telegram, the speed is obtained from the change in the axis actual position (S-0-0051 or S-0-0053).

Relevant interface signals:

at the channel interface

NC-I 7.5	Feed inhibit
NC-I 7.6	Feed hold

at the channel interface

NC-I 7.5	Drive inhibit
NC-I 7.6	Drive off
NC-I 7.7	Feed inhibit

12.12.1 Assigning

Precondition:

Analog outputs must be available:

- at the PNC-P via PROFIBUS-DP
- at the PNC-R via the osa dc I/O module.

1. Configure the relevant MACODA parameters according to your application.
   Make sure that the assignment of an analog output for the power control of a channel is applied correspondingly in MACODA.
   Example:
   MP 4075 00104 [2] 703 activates the laser power control of the 3rd channel using the 2nd analog output.
2. Enter the power and speed limit values applicable to the machine in MP 7050 00810 and 7050 00815.
3. Parametrize the relevant SERCOS parameters (see above) in the drive:
   The drive telegrams (configured via S-0-0016) of the axes involved should include the speed actual value (S-0-0040).
   If S-0-0040 is not included in the drive telegram, the speed is obtained from the change in the axis actual position (S-0-0051 or S-0-0053).
4. Link the relevant interface signals in the PLC program.

12.12.2 Activating

Following control unit start-up, the power control is deactivated and the settings from the relevant MACODA parameters are selected.

Entries from MP 7060 00010 ”Default state upon start-up” and MP 7060 00020 ”Default state upon a control reset” overwrite this presetting.

The laser power control can be activated using the presettings stored in MACODA and, if required, it can be influenced additionally in the part program using parameters:

1. switching on the laser power control with LFPON (with the default parameters) or
2. switching on the laser power control with additional parameters: LFPON LL(..) UL(..) PL(..).

The active laser power control can be set in the part program using: LFP LL(..) UL(..) PL(..).

Path-speed-dependent laser power control

Example 1 :

Setting the MACODA parameters:

4075	00104	[3]	701	Channel 1 uses the 3rd analog output
7050	00810	[1]	20	Lower power limit value 20% (= 2V)
		[2]	90	Upper power limit value 90% (= 9V)
7050	00815	[1]	500	Lower speed limit value 500 mm/min
		[2]	2000	Upper speed limit value 2000 mm/min
7050	00820	[1]	2	The path actual speed is obtained from the active plane.

Part program:

`N010 G17 G8 G62 F2500`	
`N020 G0 X0 Y0 Z0`	
`N110 G1 X10 LFPON`	Laser on
`N120 Y10`	
`N130 X0`	
`N140 Y0 LFPOFF`	Laser off **Caution**: The axes have not reached their end point
`N200 G54`	
`N210 X10 LFPON`	Laser on
`N220 Y10`	
`N230 X0`	
`N240 Y0`	
`N250 G4 F0.5 LFPOFF`	Laser off after dwell time
`N300 G55`	
`N310 X10 LFPON`	Laser on
`N320 Y10`	
`N330 X0`	
`N340 Y0 G61 LFPOFF`	Laser off when in position window is reached
`M30`	All MACODA data are effective again

Explanation:

The output voltage ranges between 2 and 9 V.
When the path speed in the active plane exceeds 2000 mm/min, 9 V are output.

When the path speed falls below 500 mm/min, 2 V are output, e.g. at the start of the movement, at the corners (depending on the maximum axis step change), at the end of the movement.

Deactivating the laser:

N140: When the laser is deactivated, the axes have not yet reached the end point because of the lag. That is why the machined part will be defective.

N250: Deactivating the laser is delayed by a programmed time.

N340: The laser is deactivated when all axes have reached the in position window.

Example 2 :

Setting the MACODA parameters:

4075	00104	[3]	701	Channel 1 uses the 3rd analog output
7050	00810	[1]	20	Lower power limit value 20% (= 2V)
		[2]	90	Upper power limit value 90% (= 9V)
7050	00815	[1]	200	Lower speed limit value 200 mm/min
		[2]	1500	Upper speed limit value 1500 mm/min
7050	00820	[1]	2	The path actual speed is obtained from the active plane.

Part program:

```
N010 G17 G8 G62 F2500
N020 G0 X0 Y0 Z0
N030 LFP UL( ,1800)       Upper speed limit value 1800 mm/min
N040 LFP CD(”X”,”Y”)      Coordinate selection X and Y
N050 X10 LFPON            Laser on
N060 Y10
N070 X0
N080 Y0 G61 LFPOFF        Laser off when in position window is
                          reached
N090 G62 G54
N100 LFPON PL(”APL”)      Selecting active plane and activating laser
N110 X10
N120 Y10
N130 X0
N140 Y0 G61 LFPOFF        Laser off when in position window is
                          reached
M30                       All MACODA data are effective again
```

Explanation:

The output voltage ranges between 2 and 9 V.

N030 Upper speed limit value 1800 mm/min

N040 When the path speed in the selected coordinates (X and Y) with the active plane (N100) exceeds 1800 mm/min, 9 V are output.

When the path speed falls below 200 mm/min, 2 V are output.

12.12.3 Deactivating

1. LPCOFF (or LFPOFF) deactivates the power control at the end of the NC block.
2. Following control reset and M2/M30, the power control is deactivated and the settings from the relevant MACODA parameters are effective again.

A Appendix

A.1 Abbreviations

Abbreviation	Description
Aux.fct.	Auxiliary function
C:	Drive name, in this case drive C (hard disk drive)
CCOMP	Cross Compensation
CEST	Central European Summer Time
CET	Central European Time
ESD	Electro-Static Discharge Abbreviation for all terms relating to electro-static discharge, e.g. ESD protection, ESD hazards, etc.
Fx	Function key with number x
GOM	Group Operating Mode
HP	Main Program (’Hauptprogramm’)
LSEC	Lead Screw Error Compensation
MDI	”Manual Data Input” mode
MP	MACODA configuration parameter
MSD	Machine-Status Display
MTB	Machine-Tool Builder
NC, CNC	Numerical Control (Computerized Numerical Control)
OI	Operator Interface
PE	Protective Earth
PLC	Programmable Logic Controller
SK	Softkey
SP	Subprogram
UTC	Universal Time Coordinated (formerly GMT)

A.2 Memory demand for function extensions

In case of function extensions, please note that the memory available in the NC (memory 2) is limited. Depending on the hardware used, the following memory is available to you as standard configuration for function extensions:

- SNCI (1,2,3): approx. 2000 KB
- SNCI 4: approx. 5600 KB

☞ **The current size of the free memory 2 can be determined via DIAGNOSTICS ▶ CONTROL ▶ MEMORY.**

Extension / functionality	Set via		Memory demand in bytes (memory 2)	
			Standard configuration	Extension
Additional axis	1001 00001 1003 00001 1003 00002 1003 00004	Drive function type Physical axis designation Channel assignment Axis movement type	9600 (8 axes)	1200 per axis
Additional channel	9040 00001	Number of channels	approx. 920000 (4 channels)	approx. 230000 per channel
Additional block in preparation	7060 00110	Number of blocks in preparation	541200 (Channel 0: 3 blocks, Channel 1: 30 blocks, Channel 2: 30 blocks, Channel 3: 4 blocks)	8200 per block
Data memory for CPL programs	7070 00010	Data areas for CPL programs	49152 (4 channels)	max. 524000
Buffer size required for execution through NFS	3080 00002	Buffer size required for execution through NFS	8192	max. 418000
Digitizing	8006 00001	Number of digitizing buffers	25600 in case of active digitizing (= 50 buffers of 512 bytes each)	max. 1024000 (= 2000 buffers)
Tool administration	Number of tool locations		34650 (105 tools)	26330 per tool

A.3 Index

Numbers
3x3 orientation tensor, 5–59
5 axis transformation
  Definition, 5–80
  Type 3032101, 5–81
  Type 3232101, 5–80
  Type 3232201, 5–80
6 axis transformation
  Definition, 5–90
  Type 3033101, 5–90
  Type 3333301, 5–90

A
Acceleration, max. path, 6–3
ACS, 5–11
ACS axis coordinate system, 5–11
Akima method, Spline, 12–68
Angle of rotation, 5–62
Area control, 9–10
AREADEF, 9–11
AREAVALID, 9–11
Asynchronous axes, 5–2
Automatic program reselection, 7–28
Auxiliary axes, 5–2
Auxiliary functions, 11–1
Axes, 5–2
  C axis, 4–53
  Gantry , 4–9
  linear, 4–1
  linear modulo, 4–5
  modulo, 4–3
  rotary, 4–2
  turret, 4–58
Axes of rotation arrangement, 5–92
Axis and coordinate names, 5–1
Axis compensations, 10–1
Axis connection vectors
  5 axis configuration, 5–83
  6 axis configuration, 5–93
Axis coupling, 4–14
Axis distance control, 12–29
Axis distance control for the digitizing process, 12–49
Axis group, 5–2
Axis index
  logical, 5–2
  physical, 5–2
Axis of rotation orientation tensor
  calculated internally, 5–67
  programmed, 5–68
Axis of rotation orientation vector
  calculated internally, 5–62
  programmed, 5–62
Axis position monitoring, 9–7
Axis transfer, 4–34
  Axis displays, 4–39
  Axis zero shift tables, 5–35
  Changing of axes between axis groups, 4–34
  Changing the axis classification, 4–38
  Removing axis, 4–36
  Renaming axes, 4–38
  Taking over axis, 4–37
Axis transformation, 5–10, 5–16, 5–77
Axis zero shift, 5–10, 5–16, 5–30
  external, 5–39
  Zero shift tables, 5–19
Axis zero shift tables, 5–34
  Axis transfer, 5–35

B
B splines, NURBS, 12–62
Backlash, 10–1
Basics, Tool compensations, 8–1
Basis vectors, Tensor orientation, 5–60
BCS, 5–11
BCS basis workpiece coordinate system, 5–11, 5–53
Bessel method, Spline, 12–67

C
C axis, 4–53
C1 cubic spline
  Linear orientation movement, 12–71
  Vector orientation, 12–70
CAD/CAM, Spline, 12–59
Canceling the distance to go, 7–29
Cartesian components, 5–56
Cartesian coordinate designation, Working range coordinates, 5–8
CCOMP, 10–7
Changing of axes between axis groups, Axis transfer, 4–34
Changing the axis classification, 4–38
Channel, 5–2
Chordal method, Spline, 12–68
Circle compensation at the quadrant transition, 6–18
Coefficient programming, Spline, 12–61
Collision detection, axis distance control, 12–55
Compensation switchover, 8–32
contour–dependent feedrate reduction, jerk–limited velocity profile generation, 6–14
Control reset, 3–6, 7–1

Coord(..), 5–69
Coordinate identifiers, 5–4
Coordinate systems, 5–9
Coordinate transformation, 5–9, 5–17, 5–53
Cross compensation, 10–7
Cutter radius compensation, 8–37
Cutting edge orientation, 8–21
Cutting edge orientation compensation, 8–21
Cutting edge radius, 8–22

D
Dead range, 9–10
Default state, 3–6
Deviations
  geometric, 10–7
  Linearity errors of axis measuring systems, 10–1
  mechanical transmission of movement (motor axis), 10–1
Digitizing, 12–29
  Axis distance control, 12–49
Direct automatic execution, 3–1
Direction of rotation, mathematically positive, 5–5, 5–8
Distance to end point, 9–9
Documentation, 1–7
Drill axis switching, 8–32
Drive–controlled interpolation, 4–31
Drive–controlled oscillation, 4–66
Dynamic limit values, 6–23

E
EMC Directive, 1–1
EMERGENCY–STOP devices, 1–5
End of program, 3–6
Endless axis, 4–3, 12–22
ESD
  Electrostatic discharge, 1–6
  grounding, 1–6
  workplace, 1–6
ESD–sensitive components, 1–6
Eulerian angle, 5–22, 5–61
Execution, direct, 3–1
Explicit tool compensation, TCS, 8–5

F
Feedrate, 5–52
Following distance for coupled axes, 4–14

G
G2/G3, 3–13
G8, 6–1, 6–11
G9, 6–11
G16, 4–41
G21, 4–41
G22, 5–32, 5–37, 5–43, 8–14, 8–30
G32, 12–23
G34, 12–13
G35, 12–13
G36, 12–13
G40, 8–38
G41, 8–38
G42, 8–38
G53, 5–32
G54...G59, 5–32
G60, 5–43
G63, 7–24
G66, 7–24
G67, 5–43
G70, 3–9
G71, 3–9
G74, 4–26
G75, 12–26
G78, 8–34
G79, 8–34
G81...G86, 8–34
G84, 12–19, 12–23
G105, 4–5
G108, 6–2, 6–16
G130, 12–8
G131, 12–8
G134, 12–13
G138, 5–41, 5–43
G139, 5–43
G145...G845, 8–16, 8–34, 8–38
G146, 8–16
G147...G847, 8–20, 8–34, 8–38
G148, 8–20
G150, 4–3
G151, 4–3
G153, 5–32
G154...G159, 5–32
G160, 5–40
G167, 5–40
G168, 5–43
G169, 5–43
G175, 12–26
G186, 8–34
G189, 4–4
G228, 6–2
G253, 5–32
G254...G259, 5–32
G260, 5–40
G268, 5–43
G269, 5–43
G275, 12–26
G301, 12–1
G328, 6–5
G329, 6–5

Appendix

G350, 12–1
G352, 5–43
G353, 5–43
G354...G359, 5–43
G360, 5–40
G374, 4–27
G408, 6–8
G510, 4–40
G511, 4–40
G512, 4–40
G513, 4–40
G515, 4–40
G516, 4–40
G517, 4–54, 4–59
G518, 4–54, 4–59
G520, 4–32
G521, 4–32
G522, 4–32
G523, 4–32
G524, 4–32
G575, 12–15
G595, 3–11
G608, 6–8
G610, 12–5
G611, 12–5
G612, 12–5
G630, 12–10
G631, 12–10
G660, 12–4
G661, 12–4
G662, 12–5
Gantry axes, 4–9
Gantry axis group, Total static torque, 4–14
Gear range switching with ”Spindle idling”, 4–65
geometric deviation, 10–7
Geometry compensation tables, 8–30
Geometry compensations, 8–10
    External (1st external) , 8–14
    General (2nd external) , 8–17
    Standard , 8–13
Grounding bracelet, 1–6
Group assignment, 5–28

H
Handwheel, 7–12
    Online correction, 7–16
Hirth axis, 4–6
Hole detection, axis distance control, 12–55

I
Idling, spindle, 4–65
Inch/metric switching, 3–8
Inclined plane, 5–42
    Tables, 5–45
Indexing, Axis designation, 5–5
Interpolation, drive–controlled, 4–31
Interpolator clock pulse, 6–1

J
Jerk–limited velocity profile generation, contour dependent feedrate reduction, 6–14
Jog
    Manual mode in machine coordinates, 7–3
    Manual mode in workpiece coordinates, 7–7
Jog axis
    in machine coordinates, 7–3
    in workpiece coordinates, 7–7

L
Laser power control, path–speed–dependent, 12–83
Leading gantry axis, 4–14
Leadscrew error compensation , 10–1
Length compensation, 8–24
LFP, 12–84
LFPOFF, 12–85
LFPON, 12–84
Limit switch, electronic, 9–1
Limit values, dynamic, 6–23
Limitation of the dynamics, 6–23
Linear axis, 4–1
Linear modulo axis, 4–5
Linear orientation movement, 5–65
    Programming, 5–70
Linearity error
    Axis measuring system, 10–1
    mechanical transmission of movement (motor axis), 10–1
Logical axes, 5–2
Look–ahead function, 6–13
    Jerk–limited velocity profile generation, 6–14
Low–Voltage Directive, 1–1
LPCOFF, 12–85
LSEC, 10–1

M
Machine zero point, 5–10
Manufacturer, 12–22
Maximum axis step change, 6–1
MCS, 5–10
MCS machine coordinate system, 5–10, 5–53
Metric/inch switching, 3–8
Modules sensitive to electrostatic discharge. *See* ESD–sensitive components
Modulo axis, 4–3
    linear, 4–5

Modulo calculation, 12–22
Monitoring position, 9–7

N
NC functions, (orientation and orientation movement, 5–69
Nibbling, 12–3
Non–rotation symmetrical tools, 5–59

O
O(...), 5–69
  Linear orientation movement, 5–70
Online correction with handwheel, 7–16
Options for Eulerian angle orientation, 5–23
Orientation, 5–22, 5–54
  with Eulerian angles, 5–61
Orientation coordinates, 5–51
  Machine kinematics, 5–54
Orientation movement, 5–52
  of the tool, 5–62
Orientation movement, linear, 5–65
Orientation of the tool coordinate system
  Explicit tool compensation, 8–5
  TCS definition in program coordinates, 8–6
Orientation tensor, 5–59
  Rotation speed, 5–68
Orientation tensor movement, 5–67
Orientation vector, 5–56
  Rotation speed, 5–64
Orientation vector movement, 5–62
Oscillating axis, 12–1
Oscillation, drive–controlled, 4–66
Override, 7–23

P
Path acceleration, limit, 6–3
Path compensation, 8–37
Path slope, 6–1, 6–11
PCS, 5–14
PCS program coordinate system, 5–14
Physical and logical axes , 5–2
Physical axes, 5–2
Polar coordinate transformation, 5–103
Polar coordinates, 5–22
Position coordinates, 5–51
Position monitoring, 9–7
Position of the tool tip, 5–53
Precision programming, 6–4
PREPNUM, 6–13, 6–17
Probe, 12–26
Program coordinate shift, 5–41
Program reselection, 7–28
Program zero point, 5–14
Programmed contour, manipulate, 5–21
Programmed contour shift, 5–41
Properties, 5–51
Pseudo coordinates, 5–4
Punching, 12–3

Q
Qualified personnel, 1–2

R
Radius compensation, 8–24
Rapid travel, limit, 9–5
Rational monomial splines, 12–65
recording a surface profile, 12–29
Recording file, Digitizing, 12–46
Reference orientations of the tool coordinate system (TCS), 6 axis transformation, 5–95
Reference point, 5–11
  approach, 4–24
Reference pose
  5 axis configuration, 5–82
  6 axis configuration, 5–94
  TCS, 8–4
Release, 1–8
Remove and return to the contour, 7–30
Removing axis, 4–36
Renaming axes, 4–38
RESET, 3–6
Reverse transformation, 5–53, 5–77
Rotary axis, 4–2, 12–22
Rotary axis speed scaling, 3–11
Rotation matrix, 5–59
Rotation symmetrical tool, 5–56
ROTAX(..), 5–70
Rounding of corners, 12–12

S
Safety instructions, 1–4
Safety markings, 1–3
Sagging, 10–7
saveph3.scs, 4–76
Scalar orientation, 5–55
SERCOS, 4–67
  Initialization, 4–67
  Logging ID numbers, 4–74
  Logging of SERCOS timing, 4–75
  parameter download, 4–70
  Parameter downloading in phase 3 on demand, 4–76
  SERCOS monitor, 4–73
Serration, 4–6
Shape, 6–7, 6–14
Shifts, 5–28
Singularities, Orientation tensor, 5–68
Space vector, 5 axis configuration, 5–83

Appendix

Spare parts, 1–6
Speeds of rotary/endless axes (Inch/metric), 3–11
Spindle, 4–51
Spindle idling, 4–65
Spindle type, 12–22
Spindle/turret axis, 4–58
Spline, 12–59
  B–, 12–62, 12–77
  C1–continuous cubic, 12–61, 12–67
  C2–continuous cubic, 12–61, 12–73
  Coefficient programming, 12–61, 12–63
  programming, 12–79
Standard operation, 1–1
Start–up, 3–6
Start–up time, 4–76
Surface grinding applications, 12–1
Switching NC blocks via high–speed signal, 12–14
Synchronous axes, 5–2
sysgrph2.scs, 4–70
sysgrph3.scs, 4–70
System control reset, 3–6, 7–1

T

Tables
  Axis zero shift, 5–19, 5–34
  Geometry compensation, 8–30
  Inclined plane, 5–45
Taking over axis, 4–37
Tangential tool guidance, 12–7
Tangential tool orientation, 12–10
Tapping
  with compensation chuck, 12–18
  without compensation chuck, 12–20
TCP, 5–53, 8–1
TCS, 5–14, 8–2
  Reference pose, 8–4
TCS definition in program coordinates, TCS, 8–6
TCS orientation, Programming, 5–71
TCS1, Reference pose, 8–4
TCSc, 8–5
TCSDEF, 8–6
TCSp, 8–6
TCSUNDEF, 8–6
Tensor orientation, 5–59
  Programming, 5–71
Test activities, 1–5
Tool clamping point, 8–1
Tool compensation tables, 8–30
Tool compensations, 8–1
Tool contact point, 8–1
Tool coordinate system, TCS, 5–14, 8–2
Tool coordinate system with active axis transformation, 8–2
Tool guidance, tangential, 12–7
Tool length, 8–1
Tool orientation, 5–10, 5–54
  See also Cutting edge orientation
  tangential, 12–10
Tool radius, 8–1
Tool reference points, 8–1
Tool zero point, 8–1
Torque reduction, 9–3
Total static torque of a gantry axis group, 4–14
Trademarks, 1–8
Turret axis, 4–58
Type and position of the axes of rotation, 6 axis transformation, 5–91
Type identification, Axis transformation, 5–78

V

Vector orientation, 5–56
  Programming, 5–69

W

Working range, 9–10
Working range coordinate programming, 5–51
Working range coordinates, 5–6, 5–51
Workpiece position compensation, 5–41
Workpiece zero point, 5–11

Z

Zero shift, 5–16
Zero shift tables, 5–34

Notes:

Rexroth
Bosch Group

Bosch Rexroth AG
Electric Drives and Controls
Bgm.-Dr.-Nebel-Str. 2
97816 Lohr a. Main, Germany
info@boschrexroth.de
www.boschrexroth.de

Printed in Germany
DOK-PNC***-NC*FUNCTION-FK02-EN-P