

SIEMENS

SINUMERIK

SINUMERIK 802D sl Turning, Milling, Nibbling

Function Manual

Valid for

Control
SINUMERIK 802D sl

| Software | Version |
|----------|---------|
| T/M | 1.4 |
| G/N | 1.4 |

06/2009
6FC5397-1CP10-3BA0

Preface

| | |
|---|----|
| Various Interface Signals (A2) | 1 |
| Axis Monitoring (A3) | 2 |
| Continuous Path Mode, Exact Stop and LookAhead (B1) | 3 |
| Acceleration (B2) | 4 |
| Travel to fixed stop (F1) | 5 |
| Gantry axes (G1) | 6 |
| Velocities, Setpoint/Actual Value Systems, Closed-Loop Control (G2) | 7 |
| Manual and Handwheel Travel (H1) | 8 |
| Auxiliary Function Outputs to PLC (H2) | 9 |
| Operating Modes, Program Operation (K1) | 10 |
| Compensation (K3) | 11 |
| Kinematic Transformation (M1) | 12 |
| Measurement (M5) | 13 |
| EMERGENCY OFF (N2) | 14 |
| Punching and Nibbling (N4) | 15 |
| Transverse Axes (P1) | 16 |

Continued on next page

SINUMERIK 802D sl

Turning, Milling, Nibbling

Function Manual




Continuation

| | |
|---|-----------|
| <u>Positioning Axes (P2)</u> | 17 |
| <u>Reference Point Approach (R1)</u> | 18 |
| <u>Rotary Axes (R2)</u> | 19 |
| <u>Spindle (S1)</u> | 20 |
| <u>Indexing Axes (T1)</u> | 21 |
| <u>Tangential Control (T3)</u> | 22 |
| <u>Speed/ torque coupling, master-slave (TE3)</u> | 23 |
| <u>Feed (V1)</u> | 24 |
| <u>Tool: Compensation and Monitoring (W1)</u> | 25 |
| <u>Appendix</u> | A |

Legal information

Warning notice system

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

| |
|--|
|  DANGER |
| indicates that death or severe personal injury will result if proper precautions are not taken. |
|  WARNING |
| indicates that death or severe personal injury may result if proper precautions are not taken. |
|  CAUTION |
| with a safety alert symbol, indicates that minor personal injury can result if proper precautions are not taken. |
| CAUTION |
| without a safety alert symbol, indicates that property damage can result if proper precautions are not taken. |
| NOTICE |
| indicates that an unintended result or situation can occur if the corresponding information is not taken into account. |


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

Qualified Personnel

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

Proper use of Siemens products

Note the following:

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

Trademarks

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

Disclaimer of Liability

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

Preface

SINUMERIK documentation

The SINUMERIK documentation is organized in three parts:

- General documentation
- User documentation
- Manufacturer/service documentation

Information on the following topics is available at <http://www.siemens.com/motioncontrol/docu>:

- Ordering documentation
Here you can find an up-to-date overview of publications.
- Downloading documentation
Links to more information for downloading files from Service & Support.
- Researching documentation online
Information on DOConCD and direct access to the publications in DOConWEB.
- Compiling individual documentation on the basis of Siemens contents with My Documentation Manager (MDM), refer to <http://www.siemens.com/mdm>.
My Documentation Manager provides you with a range of features for generating your own machine documentation.
- Training and FAQs
Information on the range of training courses and FAQs (frequently asked questions) are available via the page navigation.

Target group

This publication is intended for:

- Project engineers
- Technologists (from machine manufacturers)
- System startup engineers (systems/machines)
- Programmers

Benefits

The function manual describes the functions so that the target group knows them and can select them. It provides the target group with the information required to implement the functions.

Standard scope

This documentation only describes the functionality of the standard version. Additions or revisions made by the machine manufacturer are documented by the machine manufacturer.

Other functions not described in this documentation might be executable in the control. However, no claim can be made regarding the availability of these functions when the equipment is first supplied or in the event of servicing.

For the sake of simplicity, this documentation does not contain all detailed information about all types of the product and cannot cover every conceivable case of installation, operation, or maintenance.

Technical support

If you have any technical questions, please contact our hotline:

| | Europe/Africa |
|---|---|
| Phone | +49 (0) 180 5050 222 |
| Fax | +49 (0) 180 5050 223 |
| €0.14/min. from German landlines, cell phone prices may differ. | |
| Internet | http://www.siemens.com/automation/support-request |

| | America |
|---------------|---|
| Phone | +1 (0) 423 262 2522 |
| Fax | +1 (0) 423 262 2200 |
| E-mail | mailto:techsupport.sea@siemens.com |

| | Asia/Pacific |
|---------------|---|
| Phone | +86 (0) 1064 757575 |
| Fax | +86 (0) 1064 747474 |
| E-mail | mailto:support.asia.automation@siemens.com |

Note

For technical support telephone numbers for different countries, go to:
<http://www.automation.siemens.com/partner>

Questions about the manual

Please send any queries about the documentation (suggestions or corrections) to the following fax number or e-mail address:

Fax +49 (0) 9131 98 2176
E-mail mailto:docu.motioncontrol@siemens.com

A fax form is available in the appendix of this document.

SINUMERIK Internet address

<http://www.siemens.com/sinumerik>

Notes for the Reader

The function descriptions are only valid for the specific software version or up to the software version specified. You should request valid function descriptions for new software versions. Old function descriptions are only partly applicable for new software versions.

Technical information

Notations

The following notation and abbreviations are used in this documentation:

- PLC interface signals -> IS "Signal name" (signal data)
Example: IS "Feedrate override" (VB380x 0000)
The variable byte is located in the "to axis" range, x stands for the axis:
0 Axis 1
1 Axis 2
n Axis n+1.
- Machine data -> MD MD_NR MD_NAME (description)
e.g.: MD30300 IS_ROT_AX (rotary axis)
- Setting data -> SD SD_NR SD_NAME (description)
e.g.: SD41200 JOG_SPIND_SET_VELO (JOG velocity for the spindle)
- The chapter titles are supplemented by a code in brackets (e.g. Chapter 1: EMERGENCY STOP (N2)). This brief description is used in cross references to other chapters.

The machine and setting data are divided into the following areas:

| Range | Data area | Meaning |
|-----------------|-----------|-------------------------------|
| 200 - 399 | \$MM_ | Display machine data |
| 10,000 - 18,999 | \$MN_ | General machine data |
| 20,000 - 28,999 | \$MC_ | Channelspecific machine data |
| 30,000 - 38,999 | \$MA_ | Axis-specific machine data |
| 41,000 - 41,999 | \$SN_ | General setting data |
| 42,000 - 42,999 | \$SC_ | Channel-specific setting data |
| 43,000 - 43,999 | \$SA_ | Axis-specific setting data |

Explanations for the technical data

Protection level:

Protection levels 0 to 7 have been used. The lock for protection levels 1 to 3 (4 to 7) can be cancelled by entering the correct password and 4 to 7 via IS "Protection level" (e.g. keyswitch position). Protection level 0 is not accessible. (refer to Chapter "Various Interface Signals")

The operator only has access to information protected by one particular level and the levels below it. Various protection levels are assigned to the machine data by default and are marked with a write/read value (e.g. 4/7).

Note

Machine and setting data of protection levels 2 to 7 are described in this document. Protection level 1 machine data is only referred to in special cases (expert mode).

Data types:

The following data types are used in the control:

- **DOUBLE**
Floating-point value (64-bit value)
Input limits from $\pm 4.19 \cdot 10^{-307}$ to $\pm 1.67 \cdot 10^{308}$
- **DWORD**
Integer values (32-bit values)
Input limits from -2,147,483,648 to +2,147,483,648 (decimal);
as hexadecimal value: 0000 through FFFF
- **BYTE**
Integer values (8-bit values)
Input limits from -128 to +127 (decimal); as hexadecimal value: 00 through FF
- **BOOLEAN**
Boolean value: TRUE (1) or FALSE (0)
- **STRING**
Consisting of max. 16 ASCII characters (upper-case letters, numbers and underscore)

Detailed explanations

- Detailed explanations for the machine/setting data and interface signals used can be found in the
Reference: "Parameter Manual".
- Detailed explanations of the alarms which may occur can be found in the
References: "Diagnostics Manual".

Table of contents

| | | |
|----------|---|-----------|
| | Preface | 5 |
| 1 | Various Interface Signals (A2) | 21 |
| 1.1 | General | 21 |
| 1.2 | Signals from PLC to NCK | 22 |
| 1.2.1 | Access authorization | 22 |
| 1.2.2 | General signals | 23 |
| 1.2.3 | Signals for digital drives, to axis/spindle | 26 |
| 1.3 | Signals from NCK to PLC | 26 |
| 1.3.1 | General signals | 26 |
| 1.3.2 | Signals for digital drives, from axis/spindle | 28 |
| 1.4 | Signals from PLC to HMI | 29 |
| 1.5 | Signals from HMI to PLC | 31 |
| 1.6 | NC services | 31 |
| 1.6.1 | User interface | 31 |
| 1.6.1.1 | General information | 31 |
| 1.6.1.2 | PI service ASUB | 33 |
| 1.6.1.3 | Reading variables from the NCK area | 34 |
| 1.6.1.4 | Writing variables from the NCK area | 35 |
| 1.6.2 | NC variable | 36 |
| 1.7 | Signals from PLC | 39 |
| 2 | Axis Monitoring (A3) | 41 |
| 2.1 | Overview of monitoring functions | 41 |
| 2.2 | Motion monitoring functions | 41 |
| 2.2.1 | Contour monitoring | 41 |
| 2.2.2 | Position monitoring | 42 |
| 2.2.3 | Standstill monitoring | 44 |
| 2.2.4 | Clamping monitoring | 45 |
| 2.2.5 | Speed setpoint monitoring | 46 |
| 2.2.6 | Actual velocity monitoring | 47 |
| 2.3 | Encoder monitoring functions | 48 |
| 2.3.1 | Encoder limit frequency monitoring | 48 |
| 2.3.2 | Zero mark monitoring | 49 |
| 2.4 | Monitoring of static limits | 50 |
| 2.4.1 | Hardware limit switches | 50 |
| 2.4.2 | Software limit switches | 51 |
| 2.4.3 | Working area limitation | 52 |
| 2.5 | Supplementary conditions | 54 |
| 2.6 | Data lists | 55 |
| 2.6.1 | Machine data | 55 |
| 2.6.2 | Setting data | 56 |
| 2.6.3 | Interface signals | 56 |

| | | |
|----------|--|-----------|
| 3 | Continuous Path Mode, Exact Stop and LookAhead (B1) | 57 |
| 3.1 | Brief description | 57 |
| 3.2 | General..... | 57 |
| 3.3 | Exact stop..... | 58 |
| 3.4 | Continuous-path mode..... | 59 |
| 3.4.1 | General..... | 59 |
| 3.4.2 | Velocity reduction according to overload factor | 61 |
| 3.4.3 | Jerk limiting along the path through velocity reduction | 62 |
| 3.4.4 | Machine axis-specific jerk limiting..... | 62 |
| 3.5 | LookAhead | 63 |
| 3.6 | Data lists..... | 65 |
| 3.6.1 | Machine data..... | 65 |
| 3.6.2 | Interface signals | 66 |
| 4 | Acceleration (B2) | 67 |
| 4.1 | Acceleration profiles..... | 67 |
| 4.2 | Jerk limitation on interpolator level..... | 67 |
| 4.3 | Jerk limitation in JOG mode..... | 68 |
| 4.4 | Percentage acceleration correction, ACC..... | 68 |
| 4.5 | Data lists..... | 70 |
| 5 | Travel to fixed stop (F1) | 71 |
| 5.1 | Brief description | 71 |
| 5.2 | Functionality | 71 |
| 5.3 | Response to RESET and function abort..... | 77 |
| 5.4 | Block-search response..... | 78 |
| 5.5 | Miscellaneous | 78 |
| 5.6 | Data lists..... | 80 |
| 5.6.1 | Machine data..... | 80 |
| 5.6.2 | Setting data | 80 |
| 5.6.3 | Interface signals | 80 |
| 6 | Gantry axes (G1) | 81 |
| 6.1 | Brief description | 81 |
| 6.2 | "Gantry axes" function..... | 82 |
| 6.3 | Referencing and synchronizing gantry axes | 86 |
| 6.3.1 | Introduction | 86 |
| 6.3.2 | Automatic synchronization | 92 |
| 6.3.3 | Points to note | 93 |
| 6.4 | Start-up of gantry axes..... | 94 |
| 6.5 | PLC interface signals for gantry axes | 99 |
| 6.6 | Miscellaneous points regarding gantry axes..... | 100 |
| 6.7 | Example | 101 |
| 6.7.1 | Creating a gantry grouping..... | 101 |

| | | |
|----------|---|------------|
| 6.7.2 | Setting of NCK PLC interface | 103 |
| 6.7.3 | Commencing start-up..... | 104 |
| 6.7.4 | Setting warning and trip limits | 105 |
| 6.8 | Data lists | 107 |
| 6.8.1 | Machine data..... | 107 |
| 6.8.2 | Interface signals | 108 |
| 7 | Velocities, Setpoint/Actual Value Systems, Closed-Loop Control (G2)..... | 109 |
| 7.1 | Velocities, traversing ranges, accuracies | 109 |
| 7.1.1 | Velocities..... | 109 |
| 7.1.2 | Traversing ranges | 110 |
| 7.1.3 | Input/display resolution, computational resolution | 111 |
| 7.1.4 | Standardizing physical quantities of machine and setting data | 111 |
| 7.2 | Metric/inch measuring system | 112 |
| 7.2.1 | Conversion of basic system by part program | 112 |
| 7.2.2 | Manual switchover of the basic system | 114 |
| 7.3 | Setpoint/actual-value system | 117 |
| 7.3.1 | General | 117 |
| 7.3.2 | Drives with DRIVE-CLiQ | 118 |
| 7.3.3 | Speed setpoint and actual-value routing | 120 |
| 7.3.4 | Speed setpoint output | 122 |
| 7.3.5 | Actual-value processing..... | 123 |
| 7.3.6 | Evaluation of internal drive variables | 125 |
| 7.4 | Closed-loop control | 127 |
| 7.5 | Data lists | 130 |
| 7.5.1 | Machine data..... | 130 |
| 7.5.2 | Interface signals | 131 |
| 8 | Manual and Handwheel Travel (H1) | 133 |
| 8.1 | General characteristics of traversing in JOG | 133 |
| 8.2 | Continuous travel | 137 |
| 8.3 | Incremental travel (INC)..... | 138 |
| 8.4 | Handwheel traversal in JOG | 139 |
| 8.5 | Fixed-point approach in JOG | 142 |
| 8.5.1 | Introduction | 142 |
| 8.5.2 | Functionality | 143 |
| 8.5.3 | Parameter setting..... | 145 |
| 8.5.4 | Programming..... | 146 |
| 8.5.5 | Supplementary Conditions..... | 146 |
| 8.5.6 | Application example..... | 147 |
| 8.6 | Data lists | 148 |
| 8.6.1 | Machine data..... | 148 |
| 8.6.2 | Setting data | 148 |
| 8.6.3 | Interface signals | 149 |
| 9 | Auxiliary Function Outputs to PLC (H2) | 151 |
| 9.1 | Brief description | 151 |
| 9.2 | Programming of auxiliary functions..... | 151 |
| 9.3 | Transfer of values and signals to the PLC interface | 153 |

| | | |
|-----------|--|------------|
| 9.4 | Grouping of auxiliary functions..... | 153 |
| 9.5 | Block-search response..... | 155 |
| 9.6 | Description of the auxiliary functions | 156 |
| 9.6.1 | M function..... | 156 |
| 9.6.2 | T function..... | 156 |
| 9.6.3 | D function | 156 |
| 9.6.4 | H function | 157 |
| 9.6.5 | S function | 157 |
| 9.7 | Data lists..... | 157 |
| 9.7.1 | Machine data..... | 157 |
| 9.7.2 | Interface signals | 158 |
| 10 | Operating Modes, Program Operation (K1) | 159 |
| 10.1 | Brief description | 159 |
| 10.2 | Operating modes..... | 159 |
| 10.2.1 | Mode change | 161 |
| 10.2.2 | Functional possibilities in the individual modes | 162 |
| 10.2.3 | Monitoring functions in the individual modes | 163 |
| 10.2.4 | Interlocks in the individual modes..... | 164 |
| 10.3 | Processing a part program..... | 164 |
| 10.3.1 | Program mode and part program selection | 164 |
| 10.3.2 | Start of part program or part program block..... | 165 |
| 10.3.3 | Part program interruption | 166 |
| 10.3.4 | RESET command | 167 |
| 10.3.5 | Program control..... | 167 |
| 10.3.6 | Program status..... | 168 |
| 10.3.7 | Channel status | 169 |
| 10.3.8 | Eventdriven program calls..... | 170 |
| 10.3.9 | Asynchronous subroutines (ASUBs)..... | 178 |
| 10.3.10 | Responses to operator or program actions..... | 180 |
| 10.3.11 | Example of a timing diagram for a program run..... | 182 |
| 10.4 | Program test..... | 182 |
| 10.4.1 | General information on the program test | 182 |
| 10.4.2 | Program processing without axis movements (PRT)..... | 183 |
| 10.4.3 | Program processing in single block mode (SBL)..... | 183 |
| 10.4.4 | Program processing with dry run feedrate (DRY)..... | 185 |
| 10.4.5 | Block search: Processing of certain program sections | 185 |
| 10.4.6 | Skip part program blocks (SKP)..... | 188 |
| 10.4.7 | Graphic simulation | 189 |
| 10.5 | Timers for program execution time | 190 |
| 10.6 | Workpiece counter | 191 |
| 10.7 | Data lists..... | 193 |
| 10.7.1 | Machine data..... | 193 |
| 10.7.2 | Setting data | 194 |
| 10.7.3 | Interface signals | 195 |
| 11 | Compensation (K3)..... | 197 |
| 11.1 | Brief description | 197 |
| 11.2 | Backlash compensation | 197 |

| | | |
|-----------|--|------------|
| 11.3 | Interpolatory compensation..... | 198 |
| 11.3.1 | General | 198 |
| 11.3.2 | LEC | 200 |
| 11.3.3 | Sag compensation and angularity error compensation | 203 |
| 11.3.4 | Special features of interpolatory compensation..... | 214 |
| 11.4 | Following error compensation (feedforward control) | 215 |
| 11.4.1 | General | 215 |
| 11.4.2 | Speed feedforward control..... | 216 |
| 11.5 | Data lists | 217 |
| 11.5.1 | Machine data..... | 217 |
| 11.5.2 | Setting data..... | 217 |
| 11.5.3 | Interface signals..... | 217 |
| 12 | Kinematic Transformation (M1)..... | 219 |
| 12.1 | Brief description | 219 |
| 12.2 | TRANSMIT..... | 220 |
| 12.2.1 | Overview | 220 |
| 12.2.2 | TRANSMIT configuration | 221 |
| 12.3 | TRACYL..... | 226 |
| 12.3.1 | Overview | 226 |
| 12.3.2 | TRACYL configuration | 229 |
| 12.3.3 | Programming example, TRACYL..... | 234 |
| 12.4 | Special features of TRANSMIT and TRACYL | 236 |
| 12.5 | Data lists | 237 |
| 12.5.1 | Machine data..... | 237 |
| 12.5.2 | Interface signals..... | 238 |
| 13 | Measurement (M5) | 239 |
| 13.1 | Brief description | 239 |
| 13.2 | Hardware requirements | 239 |
| 13.2.1 | Probes that can be used..... | 239 |
| 13.2.2 | Probe connection | 241 |
| 13.3 | Channel-specific measuring..... | 241 |
| 13.3.1 | Measuring mode | 241 |
| 13.3.2 | Measurement results | 242 |
| 13.4 | Measurement accuracy and functional testing | 242 |
| 13.4.1 | Measuring accuracy..... | 242 |
| 13.4.2 | Probe functional test..... | 243 |
| 13.5 | Tool measuring in JOG | 244 |
| 13.6 | Data lists | 248 |
| 13.6.1 | Machine data..... | 248 |
| 13.6.2 | Interface signals..... | 248 |
| 14 | EMERGENCY OFF (N2) | 249 |
| 14.1 | Brief description | 249 |
| 14.2 | EMERGENCY STOP sequence | 250 |
| 14.3 | EMERGENCY STOP acknowledgment..... | 251 |
| 14.4 | Data lists | 252 |

| | | |
|-----------|---|------------|
| 14.4.1 | Machine data..... | 252 |
| 14.4.2 | Interface signals..... | 252 |
| 15 | Punching and Nibbling (N4)..... | 253 |
| 15.1 | Brief Description..... | 253 |
| 15.2 | Stroke control..... | 253 |
| 15.2.1 | General information..... | 253 |
| 15.2.2 | High-speed signals..... | 254 |
| 15.2.3 | Criteria for stroke initiation..... | 256 |
| 15.2.4 | Axis start after punching..... | 258 |
| 15.2.5 | PLC signals specific to punching and nibbling..... | 259 |
| 15.2.6 | Punching and nibbling-specific reactions to standard PLC signals..... | 259 |
| 15.2.7 | Signal monitoring..... | 260 |
| 15.3 | Activation and deactivation..... | 260 |
| 15.3.1 | Language commands..... | 260 |
| 15.3.2 | Functional expansions..... | 264 |
| 15.4 | Automatic path segmentation..... | 268 |
| 15.4.1 | General information..... | 268 |
| 15.4.2 | Operating characteristics with path axes..... | 269 |
| 15.4.3 | Response in connection with single axes..... | 273 |
| 15.5 | Rotatable tool..... | 278 |
| 15.5.1 | General information..... | 278 |
| 15.5.2 | Coupled motion of punch and die..... | 279 |
| 15.5.3 | Tangential control..... | 279 |
| 15.6 | Protection zones..... | 283 |
| 15.7 | Examples of defined start of nibbling operation..... | 284 |
| 15.8 | Data lists..... | 289 |
| 15.8.1 | Machine data..... | 289 |
| 15.8.2 | Setting data..... | 289 |
| 15.8.3 | Interface signals..... | 290 |
| 16 | Transverse Axes (P1)..... | 291 |
| 16.1 | Transverse axis definition..... | 291 |
| 16.2 | Diameter programming..... | 291 |
| 16.3 | Constant cutting rate: G96..... | 292 |
| 17 | Positioning Axes (P2)..... | 295 |
| 17.1 | Concurrent positioning axis..... | 295 |
| 17.2 | Permanently assigned PLC axis..... | 296 |
| 17.3 | Data lists..... | 299 |
| 17.3.1 | Machine data..... | 299 |
| 17.3.2 | Interface signals..... | 299 |
| 17.3.3 | Error messages..... | 300 |
| 18 | Reference Point Approach (R1)..... | 301 |
| 18.1 | Fundamentals..... | 301 |
| 18.2 | Referencing with incremental measuring systems..... | 303 |
| 18.3 | Referencing with distancecoded reference markers..... | 306 |

| | | |
|-----------|---|------------|
| 18.3.1 | General information | 306 |
| 18.3.2 | Basic parameter assignment | 307 |
| 18.3.3 | Chronological sequence | 309 |
| 18.3.4 | Phase 1: Travel across the reference marks with synchronization | 309 |
| 18.3.5 | Phase 2: Travel to fixed stop | 311 |
| 18.4 | Referencing with absolute encoders..... | 313 |
| 18.4.1 | General | 313 |
| 18.4.2 | Operator-assisted calibration | 313 |
| 18.5 | Secondary conditions for absolute encoders..... | 314 |
| 18.5.1 | Calibrating absolute encoders | 314 |
| 18.6 | Data lists | 316 |
| 18.6.1 | Machine data..... | 316 |
| 18.6.2 | Interface signals | 317 |
| 19 | Rotary Axes (R2) | 319 |
| 19.1 | General | 319 |
| 19.2 | Modulo 360 degrees | 321 |
| 19.3 | Programming rotary axes..... | 322 |
| 19.3.1 | Rotary axis with active modulo conversion..... | 322 |
| 19.3.2 | Rotary axis without modulo conversion | 323 |
| 19.4 | Data lists | 324 |
| 19.4.1 | Machine data..... | 324 |
| 19.4.2 | Setting data | 325 |
| 20 | Spindle (S1)..... | 327 |
| 20.1 | Brief description | 327 |
| 20.2 | Spindle modes | 328 |
| 20.2.1 | Spindle control mode | 329 |
| 20.2.2 | Spindle oscillation mode | 330 |
| 20.2.3 | Spindle positioning mode..... | 333 |
| 20.2.4 | Spindle axis mode..... | 336 |
| 20.3 | Synchronization..... | 337 |
| 20.4 | Gear stage change | 339 |
| 20.5 | Programming..... | 343 |
| 20.6 | Spindle monitoring | 345 |
| 20.6.1 | Axis/spindle stationary | 345 |
| 20.6.2 | Spindle in setpoint range | 346 |
| 20.6.3 | Maximum spindle speed | 346 |
| 20.6.4 | Minimum/maximum speed for gear stage..... | 346 |
| 20.6.5 | Max. encoder limit frequency | 347 |
| 20.6.6 | Target point monitoring | 348 |
| 20.7 | 2nd spindle / master spindle | 349 |
| 20.8 | Analog spindle | 350 |
| 20.9 | Data lists | 351 |
| 20.9.1 | Machine data..... | 351 |
| 20.9.2 | Setting data | 352 |
| 20.9.3 | Interface signals | 352 |

| | | |
|-----------|---|------------|
| 21 | Indexing Axes (T1) | 355 |
| 21.1 | Brief Description..... | 355 |
| 21.2 | Indexing axes..... | 356 |
| 21.2.1 | General information..... | 356 |
| 21.2.2 | Traversing indexing axes in manual JOG mode..... | 356 |
| 21.2.3 | Traversing indexing axes in AUTOMATIC modes..... | 357 |
| 21.2.4 | Traversing of indexing axes by PLC..... | 358 |
| 21.3 | Parameterization of indexing axes..... | 358 |
| 21.4 | Programming of indexing axes..... | 360 |
| 21.5 | Starting up indexing axes..... | 361 |
| 21.6 | Special features of indexing axes..... | 364 |
| 21.7 | Data lists..... | 365 |
| 21.7.1 | Machine data..... | 365 |
| 21.7.2 | Interface signals..... | 365 |
| 22 | Tangential Control (T3) | 367 |
| 22.1 | Brief description..... | 367 |
| 22.2 | Characteristics of "Tangential control" function..... | 368 |
| 22.3 | Use of "Tangential control" function..... | 369 |
| 22.3.1 | Overview..... | 369 |
| 22.3.2 | Definition of axis coupling: TANG..... | 370 |
| 22.3.3 | Switching on the axis coupling: TANGON..... | 371 |
| 22.3.4 | Response at corners, activation "with intermediate block": TLIFT..... | 371 |
| 22.3.5 | Switching off the axis coupling: TANGOF..... | 372 |
| 22.3.6 | Deactivation of corner response "with intermediate block"..... | 372 |
| 22.3.7 | Deleting the definition of axis coupling: TANGDEL..... | 372 |
| 22.3.8 | Programming examples..... | 373 |
| 22.4 | Limit angle for reversal of path direction..... | 374 |
| 22.5 | Data lists..... | 375 |
| 23 | Speed/torque coupling, master-slave (TE3) | 377 |
| 23.1 | Brief description..... | 377 |
| 23.2 | Coupling diagram..... | 379 |
| 23.3 | Configuring a coupling..... | 380 |
| 23.4 | Torque compensatory controller..... | 380 |
| 23.5 | Tension torque..... | 381 |
| 23.6 | Activating a coupling..... | 382 |
| 23.7 | Response on activation/deactivation..... | 383 |
| 23.8 | Axial interface signals..... | 385 |
| 23.9 | Axial monitoring functions..... | 386 |
| 23.10 | Response in conjunction with other functions..... | 386 |
| 23.11 | Supplementary conditions..... | 389 |
| 23.12 | Examples..... | 390 |
| 23.12.1 | Master-slave coupling between AX1=Master and AX2=Slave..... | 390 |

| | | |
|-----------|---|------------|
| 23.12.2 | Close coupling via the PLC..... | 391 |
| 23.12.3 | Close/separate coupling via part program | 391 |
| 23.12.4 | Release the mechanical brake..... | 392 |
| 23.13 | Data lists | 393 |
| 23.13.1 | Machine data..... | 393 |
| 23.13.2 | Interface signals | 393 |
| 24 | Feed (V1)..... | 395 |
| 24.1 | Path feedrate F | 395 |
| 24.1.1 | Feedrate with G33, G34, G35 (thread cutting) | 397 |
| 24.1.2 | Feedrate for G63 (tapping with compensation chuck)..... | 400 |
| 24.1.3 | Feedrate for G331, G332 (tapping without compensation chuck)..... | 400 |
| 24.1.4 | Feedrate for chamfer/rounding: FRC, FRCM | 401 |
| 24.2 | Rapid traverse G0 | 402 |
| 24.3 | Feedrate control..... | 403 |
| 24.3.1 | Overview | 403 |
| 24.3.2 | Feedrate disable and feedrate/spindle stop..... | 403 |
| 24.3.3 | Feedrate override via a machine control panel..... | 404 |
| 24.4 | Data lists | 406 |
| 24.4.1 | Machine/setting data..... | 406 |
| 24.4.2 | Interface signals | 407 |
| 25 | Tool: Compensation and Monitoring (W1) | 409 |
| 25.1 | Tool and tool compensation overview | 409 |
| 25.2 | Tool | 410 |
| 25.3 | Tool offset | 410 |
| 25.4 | Tool monitoring | 411 |
| 25.4.1 | Tool monitoring overview | 411 |
| 25.4.2 | Tool life monitoring..... | 413 |
| 25.4.3 | Workpiece count monitoring | 414 |
| 25.4.4 | Examples of the service life monitoring | 416 |
| 25.5 | Special handling of tool compensation | 416 |
| 25.6 | Data lists | 419 |
| 25.6.1 | Machine data..... | 419 |
| 25.6.2 | Interface signals | 420 |
| A | Appendix..... | 421 |
| A.1 | List of abbreviations | 421 |
| A.2 | Feedback on the documentation..... | 424 |
| A.3 | Overview | 426 |
| | Glossary | 427 |
| | Index..... | 445 |

Various Interface Signals (A2)

1.1 General

Brief description

This chapter describes the functionality of various interface signals which are of general relevance, but are not described in the function-specific chapters.

Interfaces

The exchange of signals and data between the PLC user program and the NCK (kernel of the numerical control) or HMI (display unit) is performed via various data areas. The PLC user program does not have to handle the exchange of data and signals. From the users point of view, this takes place automatically.

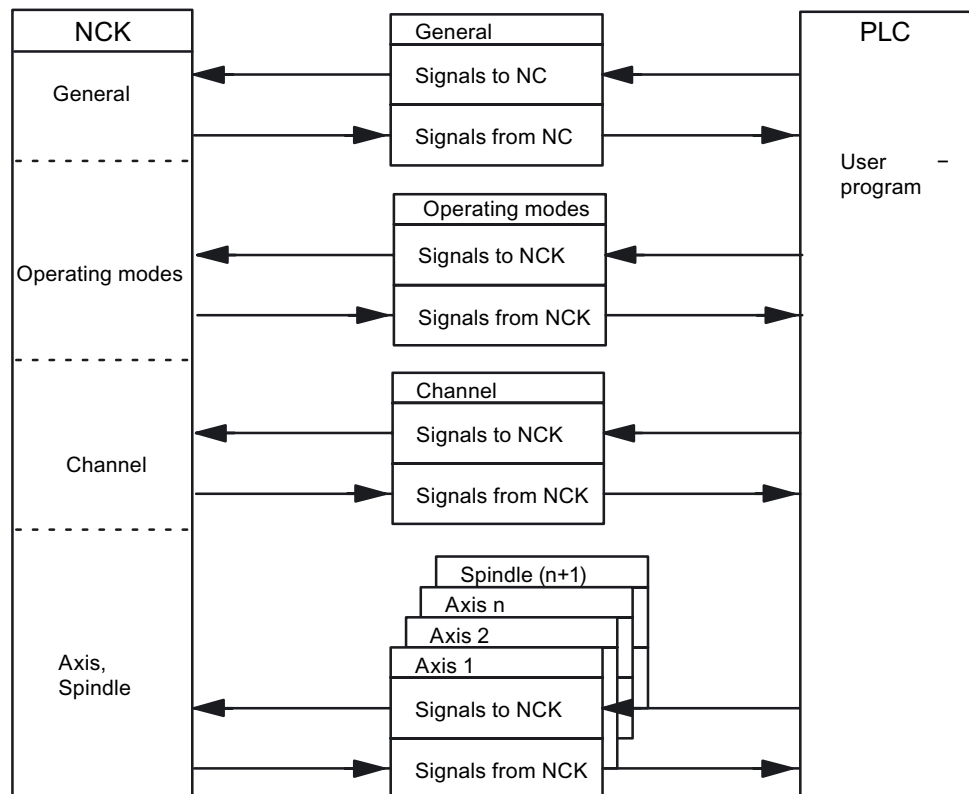


Figure 1-1 PLC/NCK interface

Cyclic signal exchange

The control and status signals of the PLC/NCK interface are updated cyclically.

The signals can be subdivided into the following groups (see previous figure):

- General signals
- Operating mode signals
- Channel signals
- Axis/spindle signals

1.2 Signals from PLC to NCK

1.2.1 Access authorization

Access authorization

Access to programs, data and functions is useroriented and controlled via eight hierarchical protection levels. These are subdivided into:

- Four password levels for Siemens, machine manufacturer (2x) and end user
- Four protection levels for end users (interface signals V2600 0000.4 to .7)

This provides a multilevel safety concept for controlling access rights.

Reference:

/BE/ Operating Instructions; First Commissioning (Commissioning), Chapter: Access levels

| Protection level | Type | User | Access to (examples) |
|------------------|-----------------|--|---|
| 0 | Pass-word | SIEMENS, reserved | |
| 1 | Pass-word | Expert mode | Defined functions, programs and data |
| 2 | Pass-word | Machine manufacturer | Defined functions, programs and data for standard commissioning |
| 3 | Pass-word | End user | Assigned functions, programs and data |
| 4 | IS V2600 0000.7 | End user: programmer, setup engineer | Less than protection level 0 to 3; specified by machine manufacturer or end user |
| 5 | IS V2600 0000.6 | End user: qualified operator, who does not program | Less than protection level 0 to 3; specified by end user |
| 6 | IS V2600 0000.5 | End user: trained operator, who does not program | only program selection, tool wear input and input of zero offsets |
| 7 | IS V2600 0000.4 | End user: semi-skilled operator | No inputs and program selection possible, only operation of machine control panel |

Decreasing access authorization
↓

Figure 1-2 Access protection

1.2.2 General signals

Delete distance-to-go (V3200 0006.2)

IS "Delete distancetogo (channelspecific)" is only active for path axes.

With the rising edge of the interface signal, the distancetogo of all axes in the geometry grouping are deleted and thus brought to a standstill with ramp stop. The next program block is then started.

Axis/spindle disable (V380x 0001.3)

IS "Axis/spindle disable" can be used for test purposes.

Axis disable (for axis):

If IS "Axis disable" is output - for this axis - no more position partial setpoints are output to the position controller; the axis travel is therefore disabled. The position control loop remains closed and the remaining following error is reduced to zero. If an axis is moved with axis disable the actual value position display shows the setpoint position and the actual velocity value display shows the setpoint velocity even though the machine axis is not actually moving. IS "RESET" (V3000 0000.7) sets the position actual value display to the real actual value of the machine. Travel commands continue to be output to the PLC for this axis. If the interface signal is cancelled again the associated axis can again traverse normally. If the

interface signal "Axis disable" is set for a traversing axis, the axis is stopped with a ramp stop.

Spindle disable (for spindle):

If IS "Spindle disable" is set, no more speed setpoints are output to the speed controller in openloop control mode and no more position partial setpoints are output to the position controller in positioning mode. The movement of the spindle is thus disabled. The speed actual value display displays the speed setpoint value. The spindle disable is cancelled via "Reset" or program end (M2) and program restart. If interface signal "Spindle disable" is set while a spindle is turning, the spindle is stopped according to its acceleration characteristic.

Deactivation:

Cancellation of the "Axis/spindle disable" (edge change 1 → 0) does not take effect until the axis/spindle is stationary (i.e. an interpolation setpoint is no longer present). The new movement begins with new specified setpoints. (E.g. new program block with movement specifications in AUTOMATIC operating mode.)

Note: actual values vary between simulated and real axis!

Follow-up mode (V380x 0001.4)

If an axis/spindle is operating in followup mode, then its setpoint position is made to track the current actual value position. The position setpoint in follow-up mode is not defined by the interpolator but derived from the actual position value. Since recording of the actual position value of the axis continues, it is not necessary to re-home the axis when follow-up mode is cancelled.

Standstill, clamping and positioning monitoring are not effective in follow-up mode.

Effect:

The IS "Follow-up mode" is only of relevance if the drive controller enable has been removed (e.g. by IS "Controller enable" = 0 signal or because of a fault in the control), or because controller enable is being re-issued.

IS "Follow-up mode" = 1:

If "Controller enable" is removed the position setpoint of the relevant axis is continuously corrected to the actual value. This state is signaled to the PLC by means of IS "Follow-up mode active" (V390x 0001.3). If the "Controller enable" is enabled again and a part program is active, a controlinternal re-positioning operation is initiated (REPOSA: linear approach with all axes) to the last programmed position. Otherwise, the axis movement starts at the new actual position (which may have changed).

IS "Follow-up mode" = 0:

If "Controller enable" is removed, the old position setpoint is maintained. If the axis is pushed out of position, a following error between position setpoint and actual value results which is corrected when IS "Controller enable" is set. The axis movement starts from the setpoint position valid before the "controller enable" was removed. IS "Followup mode active" (V390x 0001.3) is set to 0 signal during the "Hold" state. Clamping or standstill monitoring is active.

Position measuring system 1 (V380x 0001.5)

A position measuring system may be connected to the spindle. In this case the signal for the spindle has to be set.

Axes always require this signal. In this case, a position measuring system must be installed.

Controller enable (V380x 0002.1)

When the controller enable is activated for the drive, the position control loop of the axis/spindle is closed. The axis/spindle is then subject to position control.

When the controller enable is removed the position control loop and, with a delay, the speed control loop of the axis/spindle are opened.

IS "Position controller active" (V390x 0001.5) is set to 0 signal (checkback).

Activation:

The controller enable for the drive can be set and removed from the following places:

1. From the PLC user program with interface signal "Controller enable" (in normal cases)
Application: Removal of controller enable before clamping an axis/spindle.
2. The controller enable is cancelled internally by the control when certain faults occur in the machine, the drive, the position measuring system or the control (when faults occur)
Application: The traversing axes must be brought to a standstill by a rapid stop due to a fault.
3. By the control if the following event occurs: IS "EMERGENCY STOP" (V2600 0000.1) is active

Removal of controller enable for a moving axis/spindle:

- The spindle is braked to standstill with rapid stop taking account of MD36610 AX_EMERGENCY_STOP_TIME (duration of the braking ramp in error states). Alarm 21612 "Controller enable reset during movement" is then triggered.
- The position control loop of the axis/spindle is opened. Checkback signal to PLC with IS "Position controller active" (V390x 0001.5) = 0 state. The timer for the controller enable delay time (MD36620 SERVO_DISABLE_DELAY_TIME (shutdown delay of controller enable)) is also started.
- As soon as the actual speed has reached the zero speed range, the drive controller enable is removed. Checkback signal to PLC with IS "Speed controller active" (V390x 0001.6) = 0 state. The controller enable of the drive is removed at the latest after the time set in MD36620 SERVO_DISABLE_DELAY_TIME has expired.
- Notice: If the setting for the controller enable shutdown delay is too small the controller enable will be removed even though the axis/spindle is still moving. The axis/spindle is then stopped abruptly with setpoint 0.
- The actual position value of the axis/spindle continues to be acquired by the control.

This axis/spindle state cannot be changed until after "Reset".

Interpolatory axis grouping:

All the axes traversing within the interpolatory axis grouping are stopped as soon as the controller enable signal is cancelled for **one** of the axes.

The axes are brought to a standstill as described above. All axes in the geometry grouping are brought to a standstill with rapid stop. Alarm 21612 "Controller enable reset during movement" is also triggered. Continued processing of the NC program after this event is no longer possible.

1.2.3 Signals for digital drives, to axis/spindle

Drive parameter set selection A, B, C (V380x 4001.0 to .2)

Using bit combination A, B, C, the PLC user program can select up to eight different drive parameter sets.

Speed controller integrator disabled (V380x 4001.6)

The PLC user program inhibits the integrator of the speed controller for the drive. The speed controller is thus switched from PI to P controller.

Pulse enable (V380x 4001.7)

The PLC user program enables the pulses for the axis/spindle. However, the pulse enable is only activated for the drive module if all the enable signals are present.

1.3 Signals from NCK to PLC

1.3.1 General signals

Drives in cyclic operation (V2700 0002.5)

The PLC is signaled via the NCK by means of a cyclical exchange of data that the available drives have reached ramp-up status.

Drive ready (V2700 0002.6)

The PLC is signaled via NCK that all available drives are ready to operate. IS "Drive Ready" (group signal) is active on all axes and spindles.

NCK alarm is active (V2700 0003.0)

The control sends this signal to the PLC to indicate that at least one NCK alarm is active. An enquiry can be made via the channel-specific interface (V3300 0004.7) as to whether a processing stop has been triggered.

Ambient temperature alarm (V2700 0003.6)

The ambient temperature or fan monitoring function has responded.

NCK alarm channel-specific active (V3300 0004.6)

The control system sends this signal to the PLC to indicate that at least one NCK alarm is active for the channel. To what extent this may influence whether the current program run will be interrupted or aborted can be determined from IS "NCK alarm with processing stop is active" (V3300 0004.7).

External language mode active (V3300 4001.0)

The control system sends this signal to the PLC to indicate that the active program language used for the part program is not a SIEMENS language. A language changeover has been made with G291.

NCK alarm with processing stop present (V3300 0004.7)

The control sends this signal to the PLC to indicate that at least one NCK alarm, which has interrupted or aborted the current program run (processing stop), is active for the channel.

Follow-up active (V390x 0001.3)

Follow-up mode for this axis is active.

See Section: Signals from PLC to NCK, follow-up mode (V380x 0001.4)

Axis/spindle stationary (V390x 0001.4)

The current velocity of the axis or actual speed of the spindle is within the range which is defined as standstill. This range is defined with MD36060 STANDSTILL_VELO_TOL (maximum velocity/speed for signal "Axis/spindle stationary").

Position control active (V390x 0001.5)

The position control loop for the axis/spindle is closed; the position control function is active.

Speed control active (V390x 0001.6)

The speed control loop for the axis/spindle is closed; the speed control function is active.

Current control active (V390x 0001.7)

The current control loop for the axis/spindle is closed; the current control function is active.

Lubrication pulse (V390x 1002.0)

The IS "Lubrication pulse" is sent by the NCK and **changes status** once the axis/spindle has traveled a greater distance than that set in MD33050 LUBRICATION_DIST (travel distance for lubrication from PLC)

1.3.2 Signals for digital drives, from axis/spindle

Active drive parameter set A, B, C (V390x 4001.0 to .2)

The drive module sends this checkback to the PLC to indicate which drive parameter set is currently active. With bit combination A, B, C, eight different drive parameter sets can be selected by the PLC.

Drive ready (V390x 4001.5)

Checkback signal indicating that the drive is ready. The conditions required for traversing the axis/spindle are fulfilled.

Integrator for n-controller disabled (V390x 4001.6)

The speed-controller integrator is disabled. The speed controller has thus been switched from PI to P controller.

Pulse enabled (V390x 4001.7)

The pulse enable for the drive module is available. The axis/spindle can now be traversed.

Motor temperature warning (V390x 4002.0).

The drive module signals to the PLC that the motor temperature has exceeded the warning threshold. If the motor temperature remains at this level, the drive will be stopped after a defined time (drive MD) and the pulse enable removed.

Heat sink temperature warning (V390x 4002.1).

The drive module signals to the PLC that the heat sink temperature has exceeded the warning threshold. The pulse enable will be removed for the drive module in question after 20 seconds.

Ramp-up procedure completed (V390x 4002.2)

This signal confirms that the actual speed value has reached the new setpoint allowing for the tolerance band set in the drive. The ramp-up procedure is thus completed. Any subsequent speed fluctuations due to load changes will not affect the interface signal.

$|M_d| < M_{dx}$ (V390x 4002.3)

This signal indicates that the current torque $|M_d|$ is lower than the threshold torque M_{dx} set in the drive:

$|n_{act}| < n_{min}$ (V390x 4002.4)

This signal indicates that the actual speed $|n_{act}|$ is lower than the set minimum speed n_{min} .

$n_{act} < n_x$ (V390x 4002.5)

This signal indicates that the actual speed $|n_{act}|$ is lower than the set threshold speed n_x .

$n_{act} = n_{set}$ (V390x 4002.6)

The PLC receives this signal confirming that the actual speed n_{act} has reached the new setpoint allowing for the tolerance band set in the drive and continues to remain within the tolerance band.

1.4 Signals from PLC to HMI

OP key disable (V1900 5000.2)

IS "OP key disable" can be applied to disable (1 signal) or enable (0 signal) the connected keyboard.

Program number (VB1700 1000)

A declared program number is transferred from the PLC to HMI if a NC program is selected by the PLC. The current NC program selected can be stored via the command interface (see VB 1700 1001) and also selected again.

With SINUMERIK 802D sl, a program with the program name (STRING) is administered. To assign a program name to a program number, the file PLCPROG.LST is available in the control. In this assignment list, the names for a maximum of 255 programs can be declared and assigned.

The use of the numbers is divided into the protection areas of the programs:

- 1 to 100: User area (end user protection level)
- 101 to 200: Machine manufacturer (machine manufacturer protection level)
- 201 to 255: SIEMENS (SIEMENS protection level)

The file PLCPROG.LST - where the minimum protection level for end user has been set - can be edited using the operation: "System" -> "PLC" -> "Program list" or by using the standard editor (program manager -> User cycles). The protection level for the editor should be at least the machine manufacturer protection level.

This file can also be created externally and loaded to the control via the PCIN tool / V.24 interface. The following structure specification must be observed, header entry first:

```
%_N_PLCPROG_LST
;$PATH=/_N_CUS_DIR
```

Each line contains two columns. These are separated via tabs or blanks or " | " characters. The first column contains the program number, the second the program name.

Examples:

```
%_N_PLCPROG_LST
;$PATH=/_N_CUS_DIR
1|BOHR2.MPF
2|PUMPT14.MPF
54|BOHR3.MPF
```

"Program number" (VB 1700 1000) corresponds to the following IS:

- "Program has been selected" (V1700 2000.0)
- "Program selection error" (V1700 2000.1).

When a program number > 0 is written, the program selection is started by the PLC. As soon as the HMI detects a program number > 0, it begins with the internal processing of this job and sets the program number (VB 1700 1000) to 0.

PLC waits until the acknowledgement signal from HMI is received: V1700 2000.0 or V1700 2000.1 and evaluates this immediately. The acknowledge signals are available for one PLC cycle once they have been received and are then automatically deleted by the PLC operating system.

Command (VB1700 1001)

A command job is transferred from the PLC to the HMI.

| Command | Action |
|---------|--|
| 0 | None |
| 1 | Save name of the selected program |
| 2 | Select program with saved program name |

"Command" (VB1700 1001) corresponds to the following IS:

- "Execute command" (V1700 2001.0)
- "Command execution error" (V1700 2001.1)

When a command > 0 is written, the job is started by the PLC. As soon as the HMI detects a command > 0, it begins with the internal processing of this job and sets the command (VB1700 1001) to 0.

PLC waits until the acknowledgement signal has been reached by HMI: V1700 2001.0 or V1700 2001.1 and evaluates this immediately. The acknowledgement signals are available for one PLC cycle once they have been received and are then automatically deleted by the PLC operating system.

1.5 Signals from HMI to PLC

Program has been selected (V1700 2000.0)

Successful selection of the required NC program is signaled back from the HMI to the PLC. This signal is available for one PLC cycle. It corresponds with VB1700 1000.

Program selection error (V1700 2000.1)

Failed selection of the required NC program is signaled back from the HMI to the PLC. This signal is available for one PLC cycle. It corresponds with VB1700 1000.

Execute command (V1700 2001.0)

Successful execution of the required command is signaled back from the HMI to the PLC. This signal is available for one PLC cycle. It corresponds with VB1700 1001.

Command execution error (V1700 2001.1)

Failed execution of the required command is signaled back from the HMI to the PLC. This signal is available for one PLC cycle. It corresponds with VB1700 1001.

1.6 NC services

1.6.1 User interface

1.6.1.1 General information

Communication jobs can be performed via the "NC services" PLC/NCK interface. The following services are available for this:

- Start program invocation services (PI services) in the NCK area (e.g. ASUB)
- Read variables from the NCK area
- Write variables from the NCK area

The activation of the respective service is performed via the global part of the interface. The parameterization of the individual services is described below.

Job, global part

Only one service can run at a time. The service is selected via V1200 0000.1 and V1200 0000.2:

| Service | V1200 0000.2 | V1200 0000.1 |
|-----------------------------------|--------------|--------------|
| Start PI service in the NCK area | 1 | 0 |
| Read variables from the NCK area | 0 | 0 |
| Write variables from the NCK area | 0 | 1 |

Start:

A job is started by setting the signal V1200 0000.0 = 1. A new job can only be started if the previous job has been completed, i.e. the acknowledgement signals ("Job completed" V1200 2000.0 and "Error in job" V1200 2000.1) must be zero.

The execution of a job may take several PLC cycles and vary depending on the utilization; thus, this function is not real-time-capable.

Note

A job already started cannot be cancelled. If the "Start" signal is inadvertently reset before receiving the acknowledgement, the result signals for this job are not refreshed; the job, however, is executed.

Job, global part

The results are written by the PLC operating system; therefore, these signals can only be written by the user.

If the job was completed without errors, the "Job completed" signal V1200 2000.0 is set to 1. If an error occurs while executing a read/write job, the "error in job" signal V1200 2000.1 is set.

The result signals in VB1200 2000 are global bits for the whole job. Possible error causes can be here, e.g.:

- Number of variables (V1200 0001) outside of the valid range
- Variable index (V1200 1000) outside of the valid range

After evaluating the result, the "Start" signal (V1200 0000.0) is reset by the user. The PLC operating system then resets "Job completed" or "Error in job".

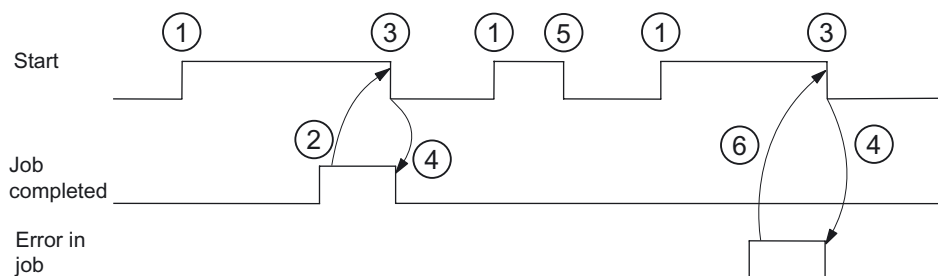


Figure 1-3 Pulse diagram

Explanations regarding the pulse diagram:

1. Starting of the job by setting "Start" ("Job completed" and "Error in job" must be reset)
2. Job completed without errors (the results of the individual variables must still be evaluated)
3. Resetting "Start" after receiving the result
4. Signal change by PLC operating system
5. If the "Start" signal is reset inadvertently before receiving the result, the output signals are not refreshed without influence on the internal execution of the function triggered
6. Error in job

1.6.1.2 PI service ASUB

Initialization

With the ASUB PI service, it is possible to assign the interrupt numbers 1 and 2 fixed program names from the PLC. Prerequisite for this is the existence of the PLCASUP1_SPF or PLCASUP2_SPF programs in the CMA directory.

| PI index | Function |
|-----------------|---|
| VB1200 0001 = 1 | Assignment of Interrupt 1 to the CMA_DIR/PLCASUP1_SPF program. The interrupt has Priority 1. |
| VB1200 0001 = 2 | Assignment of Interrupt 2 to the CMA_DIR/PLCASUP2_SPF program. The interrupt has Priority 2. |

The following must be taken into account during the initialization:

- The PI service ASUB requires executing only once after a restart and is then retained.
- An initialization may only be performed when the channel is not active.
- If a "Ramp-up" program event has been configured, the initialization may only be started after the end of the program event.

Relevant interface signals

| | Address | Name | Valid values |
|--------|--------------|-------------------|--------------|
| Job | V1200 0000.0 | Start | 0/1 |
| | V1200 0000.1 | Write variable | 0 |
| | V1200 0000.2 | PI service | 1 |
| | V1200 0001 | PI index | 1,2 |
| Result | V1200 2000.0 | Request completed | 0/1 |
| | V1200 2000.1 | Error in job | 0/1 |

1.6.1.3 Reading variables from the NCK area

1 to 8 values can be read with a read job (variable x: 0...7). There is a variable-specific part of the interface for this:

- Job: V120x 1000
- Result: V120x 3000

Job, variable-specific part

NC variable:

The NC variable is selected in the variable index (VB120x 1000), see Section: NC variable Area number, column / line index (VB120x 1001 ... VB120x 1005)

Various variables are declared as fields. For flexible addressing, the relevant field index must be specified as a column and/or line index (e.g. R parameter no.).

Values:

The range 120x 1008 ... 11 is not relevant for reading.

Result, variable-specific part

A result is reported for each variable in the job.

If the read process was successful, "Variable valid" (V120x 3000.0) is set to 1; the access result VB120x 3001 is 0.

When reading, the data from VB120x 3004 are entered type-specifically.

In case of error, V120x 3000.0 remains "0", and an entry is made in the access result VB120x 3001:

- 0: No error
- 3: Illegal access to object
- 5: Invalid address
- 10: Object does not exist

Values:

When reading, the read data are in the range 120x 3004...7, in the data type specific for the respective variable (if required, the values are converted from 64-bit to 32-bit REAL).

Relevant interface signals

| | Address | Name | Valid values |
|---------------------------|--------------|---------------------|----------------------------|
| Job, global part | V1200 0000.0 | Start | 0/1 |
| | V1200 0000.1 | Write variable | 0 |
| | V1200 0000.2 | PI service | 0 |
| | VB1200 0001 | Number of variables | 1 ... 8 |
| Job, variable-specific | VB120x 1000 | Variable index | See Section NC variable |
| | VB120x 1001 | Area number | |

| | Address | Name | Valid values |
|--------------------------------------|---|---|----------------------------|
| part | VB120x 1002 | Line index, NCK variable | |
| | VB120x 1004 | Column index, NCK variable | |
| Job, global part | V1200 2000.0 | Request completed | 0/1 |
| | V1200 2000.1 | Error in job | 0/1 |
| Result, variable-specific part | V120x 3000.0 | Invalid variable | 0/1 |
| | VB120x 3001 | Access result | 0/3/5/10 |
| | VB120x 3004/ VW120x 3004/ VD120x 3004 | Value of NCK variable, data type depends on variable index | See Section NC variable |

1.6.1.4 Writing variables from the NCK area

1 to 8 values can be written with a write job (variable x: 0...7). There is a variable-specific part of the interface for this:

- Job: V120x 1000
- Result: V120x 3000

Job, variable-specific part

NC variable:

The NC variable is selected in the variable index (VB120x 1000), see Section: NC variable
Area number, column / line index (VB120x 1001 ... VB120x 1005)

Various variables are declared as fields. For flexible addressing, the relevant field index must be specified as a column and/or line index (e.g. R parameter no.).

Values:

The values to be written must be entered in the range 120x 1008...11 in the data type specific for the appropriate variable.

If necessary, the values are converted (e.g. NCL floating-point values (64-bit) into the PLC format (32-bit) and vice versa). A loss of accuracy results from the conversion from 64-bit to 32bit REAL. The maximum accuracy of 32bit REAL numbers is approximately 10^7 .

Result, variable-specific part

A result is reported for each variable in the job.

If the read process was successful, "Variable valid" (V120x 3000.0) is set to 1; the access result VB120x 3001 is 0.

When reading, the data as of VB120x 3004 is entered type-specifically.

In case of error, V120x 3000.0 remains "0", and an entry is made in the access result VB120x 3001:

- 0: No error
- 3: Illegal access to object
- 5: Invalid address
- 10: Object does not exist

Values:

The range 120x 3004...07 is not relevant for writing.

Relevant interface signals

| | Address | Name | Valid values |
|-----------------------------------|---|---|----------------------------|
| Job, global part | V1200 0000.0 | Start | 0/1 |
| | V1200 0000.1 | Write variable | 1 |
| | V1200 0000.2 | PI service | 0 |
| | VB1200 0001 | Number of variables | 1 ... 8 |
| Job, variable-specific part | VB120x 1000 | Variable index | See Section NC variable |
| | VB120x 1001 | Area number | |
| | VB120x 1002 | Line index, NCK variable | |
| | VB120x 1004 | Column index, NCK variable | |
| | VB120x 3004/ VW120x 3004/ VD120x 3004 | Value of NCK variable, data type depends on variable index | |
| Job, global part | V1200 2000.0 | Request completed | 0/1 |
| | V1200 2000.1 | Error in job | 0/1 |
| Result, variable-specific part | V120x 3000.0 | Invalid variable | 0/1 |
| | VB120x 3001 | Access result | 0/3/5/10 |

1.6.2 NC variable

Variable cuttEdgeParam

Compensation value parameters and cutting edge list with D numbers for a tool.

The meanings of the individual parameters depend on the type of the tool in question. Currently, 25 parameters are reserved for each tool edge (but only a part of them is loaded with values). To be able to remain flexible for future extensions, it is not recommended to use a fixed value of 25 parameters for calculation, but the variable value 'numCuttEdgeParams' (variable index 2).

For a detailed description of the tool parameters, please refer to Chapter "Tool Offset (W1)".

| | Variable cuttEdgeParam [r/w] |
|-------------|---|
| VB120x 1000 | 1 |
| VB120x 1001 | - |
| VW120x 1002 | (EdgeNo - 1) * numCuttEdgeParams + ParameterNo (WORD) |
| VW120x 1004 | T number (1...32000) (WORD) |
| VD120x 1008 | Write: Data to NCK variable x (data type of the variables: REAL) |
| VD120x 3004 | Read: Data from NCK variable x (data type of the variables: REAL) |

Variable numCuttEdgeParams

Number of P elements of an edge

| | Variable numCuttEdgeParams [r] |
|------------|---|
| VB120x1000 | 2 |
| VB120x1001 | - |
| VW120x1002 | - |
| VW120x1004 | - |
| VD120x1008 | - |
| VW120x3004 | Read: Data from NCK variable x (data type of the variables: WORD) |

Variable linShift

Translation of a settable work offset (channel-specific settable frames)

They only exist if MD18601 MM_NUM_GLOBAL_USER_FRAMES > 0.

There are the frame indices:

- 0: ACTFRAME = current resulting work offset
- 1: IFRAME = current settable work offset
- 2: PFRAME = current programmable work offset
- 3: EXTFRAME = current external work offset
- 4: TOTFRAME = current total work offset = total of ACTFRAME and EXTFRAME
- 5: ACTBFRAME = current total base frame
- 6: SETFRAME = current 1st system frame (PRESET, scratching)
- 7: EXTFRAME = current 2nd system frame (PRESET, scratching)
- 8: PARTFRAME = current 3rd system frame (TCARR and PAROT with orientable tool carrier)
- 9: TOOLFRAME = current 4th system frame (TOROT and TOFRAME)
- 10: MEASFRAME = result frame for workpiece and tool gauging
- 11: WPFRAME = current 5th system frame (workpiece reference points)
- 12: CYCFRAME = current 6th system frame (cycles)

The max. frame index is 12.

The value of numMachAxes is contained in the variable with variable index 4.

| | Variable linShift [r] |
|------------|---|
| VB120x1000 | 3 |
| VB120x1001 | - |
| VW120x1002 | Frame index * numMachAxes + axis number |
| VW120x1004 | - |
| VD120x1008 | - |
| VD120x3004 | Read: Data from NCK variable x (data type of the variables: REAL) |

Variable numMachAxes

No. of the highest existing channel axis

If there are no gaps between channels, this corresponds to the number of existing axes in the channel.

| | Variable numMachAxes [r] |
|------------|---|
| VB120x1000 | 4 |
| VB120x1001 | - |
| VW120x1002 | - |
| VW120x1004 | - |
| VD120x1008 | - |
| VW120x3004 | Read: Data from NCK variable x (data type of the variables: WORD) |

Variable rpa

R parameters

| | Variable rpa [r/w] |
|------------|---|
| VB120x1000 | 5 |
| VB120x1001 | - |
| VW120x1002 | R number + 1 |
| VW120x1004 | - |
| VD120x1008 | Write: Data to NCK variable x (data type of the variables: REAL) |
| VD120x3004 | Read: Data from NCK variable x (data type of the variables: REAL) |

Variable actLineNumber

Line number of the current NC block:

- 0: Prior to program start
- -1: Not available due to error
- -2: Not available due to DISPLOF

| | Variable actLineNumber [r] |
|------------|---|
| VB120x1000 | 6 |
| VB120x1001 | - |
| VW120x1002 | - |
| VW120x1004 | - |
| VD120x1008 | - |
| VD120x3004 | Read: Data from NCK variable x (data type of the variables: DINT) |

1.7 Signals from PLC

Commissioning mode

The ramp-up modes are signaled via bit 0 and bit 1 (VB1800 1000) in the user interface.

| Commissioning mode | VB1800 1000.1 | VB1800 1000.0 |
|-----------------------------|---------------|---------------|
| Normal rampup | 0 | 0 |
| Ramp-up with default values | 0 | 1 |
| Ramp-up with saved data | 1 | 0 |

Axis Monitoring (A3)

2.1 Overview of monitoring functions

Overview of monitoring functions

- Motion monitoring functions
 - Contour monitoring
 - Position monitoring
 - Standstill monitoring
 - Clamping monitoring
 - Speed setpoint monitoring
 - Actual velocity monitoring
 - Encoder monitoring functions
- Monitoring of static limits
 - Limit switch monitoring
 - Working area limitation

2.2 Motion monitoring functions

2.2.1 Contour monitoring

Function

The principle on which the contour monitoring function works is the constant comparison of the measured actual position value with that calculated from the NC position setpoint. For the precalculation of the following error, a model is used that simulates the dynamics of the position control including feedforward control.

So that the monitoring function does not respond incorrectly on slight speed fluctuations (caused by changes of load) a tolerance band is allowed for the max. contour deviation.

If the permissible actual value deviation entered in MD36400 CONTOUR_TOL (tolerance band contour monitoring) is exceeded, an alarm is signaled and the axes are stopped.

Effectiveness

Contour monitoring is active for axes and position-controlled spindles.

Effect

If the contour deviation is too large, this has the following effect:

- Alarm 25050 "Contour monitoring" is triggered
- The axis/spindle is brought to a standstill via a speed setpoint ramp with rapid stop (with open position control loop).
The braking ramp time is set in MD36610 AX_EMERGENCY_STOP_TIME (braking ramp time for error states).
- If the axis/spindle is involved in interpolation with other axes/spindles, these are brought to a standstill with rapid stop with following error reduction (position setpoint = constant).

Remedy

- Increase tolerance band of monitoring in MD36400
- The actual "servo gain factor" must correspond to the desired servo gain factor set via MD32200 POSCTRL_GAIN (servo gain factor).
With analog spindles:
MD32260 RATED_VELO (rated motor speed) and
MD32250 RATED_OUTVAL (rated output voltage) must be checked.
- Check optimization of the speed controller
- Check smooth running of the axes
- Check machine data for traversing movements (feed override, acceleration, max. speeds, ...)

2.2.2 Position monitoring

Function

In order to ensure that an axis reaches the required position within a specified time, the timer that can be configured in MD36020 POSITIONING_TIME (time delay exact stop fine) is started at the end of each motion block (setpoint has reached target) and, when the timer runs out, a check is made to ascertain whether the axis has reached its setpoint within the tolerance of MD36010 STOP_LIMIT_FINE (exact stop fine).

For details on "Exact stop coarse and fine" see Chapter "Continuous Path Mode, Exact Stop and Look-Ahead (B1)"

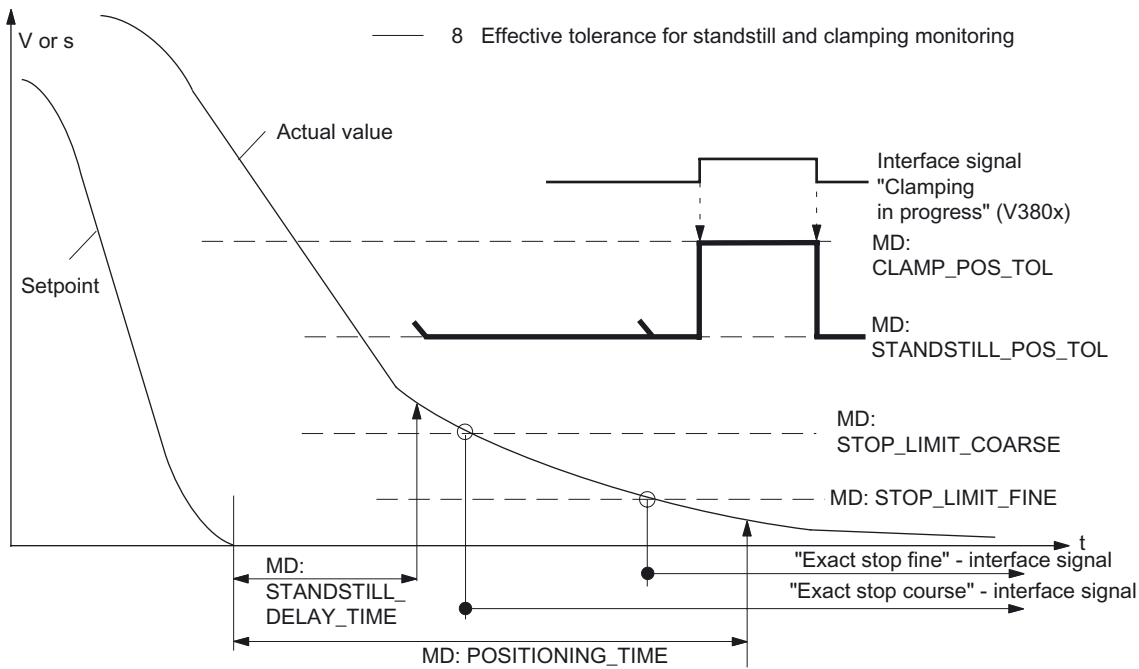


Figure 2-1 Relation between position, standstill, and clamping monitoring

Effectiveness

Positioning monitoring is always activated after the termination of motion blocks "according to the setpoint" (setpoint has reached destination).

Position monitoring is active for axes and position-controlled spindles.

Deactivation

When the programmed "Exact stop limit fine" has been reached or a new setpoint has been output (e.g. for positioning according to "Exact stop coarse" followed by a block change), the position monitoring is deactivated.

Effect

If the limit value for "Exact stop fine" has not yet been reached when the positioning monitoring time has elapsed, the following action is performed:

- Output of alarm 25080 "Positioning monitoring"
- The affected axis/spindle is brought to a standstill using a rapid stop (with open position control loop) along a speed setpoint ramp.
The braking ramp duration is set in MD36610 AX_EMERGENCY_STOP_TIME (braking ramp duration for error states).
- If the axis/spindle is involved in interpolation with other axes/spindles, these are stopped using a rapid stop with following error reduction (default for partial position setpoint = 0).

Cause of error/Remedy

- Position controller gain too low --> change machine data for position controller gain MD32200 POSCTRL_GAIN (servo gain factor)
- Positioning window (exact stop fine), position monitoring time, and position controller gain have not been coordinated --> change machine data:
MD36010 STOP_LIMIT_FINE (exact stop fine),
MD36020 POSITIONING_TIME (exact stop fine delay time),
MD32200 POSCTRL_GAIN (servo gain factor)

Rule of thumb

- Positioning window large --> max. position monitoring time can be set to a relatively short value
- Positioning window small --> max. position monitoring time must be set to a relatively long value
- Position controller gain low --> max. position monitoring time must be set to a relatively long value
- Position controller gain high --> max. position monitoring time can be set to a relatively short value

Note

The size of the positioning window affects the block change time. The smaller the tolerances that are selected, the longer the positioning action will take, which in turn means a longer time before the next command can be executed.

2.2.3 Standstill monitoring

Function

At the end of a motion block (position setpoint has reached target), a check is made as to whether the axis is not more than the distance specified in MD36040 STANDSTILL_DELAY_TIME (standstill monitoring delay time) away from its setpoint after the configurable delay time in MD36060 STANDSTILL_POS_TOL (standstill tolerance) has expired. Otherwise, an alarm will be triggered.

Effectiveness

Standstill monitoring is always active after "Standstill monitoring delay time" active has expired, as long as no new travel command is present.

Standstill monitoring is active on axes and position-controlled spindles.

Effect

When the monitoring function responds, it has the following effects:

- Alarm 25040 "Standstill monitoring" is triggered
- The affected axis/spindle is brought to a standstill with rapid stop (with open position control loop) along a speed setpoint ramp. The braking ramp time is set in MD36610 AX_EMERGENCY_STOP_TIME (duration of the braking ramp for error states).
- If the axis/spindle is involved in interpolation with other axes/spindles, these are stopped by rapid stop with following error reduction (default for position partial setpoint = 0).

Cause of error / remedy

- Position control gain too high (control loop oscillation) --> change machine data for control gain MD32200 POSCTRL_GAIN (servo gain factor)
- Standstill window too small --> change machine data MD36030 STANDSTILL_POS_TOL (standstill tolerance)
- Axis is mechanically "pushed" out of position --> eliminate cause

2.2.4 Clamping monitoring

Function

If the axis must be clamped once it has been positioned, the clamping monitoring function can be activated via IS "Clamping in progress" (V380x 0002.3).

This may be necessary as the axis can be forced further from the setpoint than the standstill tolerance permits during the clamping process. The amount by which the axis may leave the command position is specified in MD36050 CLAMP_POS_TOL (clamping tolerance for interface signal "Clamping active").

Effectiveness

Clamping monitoring is activated by the interface signal "Clamping active". It replaces standstill monitoring during clamping.

Clamping monitoring is active on axes and position-controlled spindles.

Effect

If the axis is pushed out of position beyond the clamping tolerance during clamping the following occurs:

- Alarm 26000 "Clamping monitoring" is triggered
- The affected axis/spindle is brought to a standstill with rapid stop (with open position control loop) along a speed setpoint ramp. The braking ramp time is set in MD36610 AX_EMERGENCY_STOP_TIME (duration of the braking ramp for error states).
- If the axis/spindle is assigned to an interpolatory grouping with other axes/spindles, then these are also braked by rapid stop with following error reduction (default for partial position setpoint = 0).

2.2.5 Speed setpoint monitoring

Function

Speed setpoint monitoring checks whether the setpoint specification does not exceed the maximum permissible drive speed in MD 36210 MD36210 CTRLOUT_LIMIT (maximum speed setpoint). If required, the speed is limited to this value and the axis/spindle stopped and an alarm output.

The maximum speed for the axes (in percent) exceeds the speed at which the velocity in MD3200 MAX_AX_VELO is reached (100%). This also determines the control margin.

On an analog spindle the maximum speed that can be output must not exceed the speed reached at the maximum setpoint output voltage of 10 V (100%).

The speed setpoint consists of the speed setpoint of the position controller and the feedforward control parameter (if feedforward control is active).

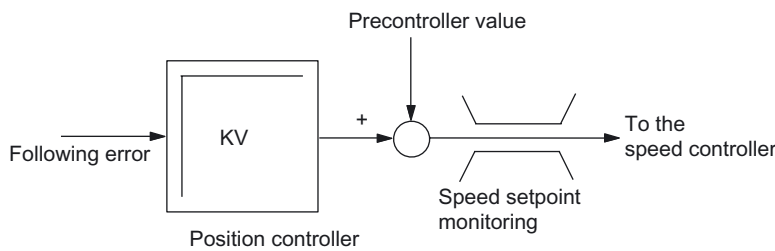


Figure 2-2 Speed setpoint calculation

Effectiveness

Speed setpoint monitoring is always active for axes and spindles.

Effect

The following occurs if the maximum speed setpoint value is exceeded:

- Alarm 25060 "Speed setpoint limiting" is triggered
- The affected axis/spindle is brought to a standstill using a rapid stop (with open position control loop) along a speed setpoint ramp.
The braking ramp duration is set in MD36610 AX_EMERGENCY_STOP_TIME (braking ramp duration for error states).
- If the axis/spindle is involved in interpolation with other axes/spindles, these are stopped using a rapid stop with following error reduction (default for partial position setpoint = 0).

Note

In the "Expert mode" access level (protection level 1), MD36220 CTRLOUT_LIMIT_TIME can be used to set a delay time, after the expiration of which an alarm is output and the axes are brought to a standstill. The default value of this time is zero.

Using speed setpoint limiting will turn the control loop into a non-linear control loop. This generally causes contour deviations if speed setpoint limiting is continued for an axis. A control margin must therefore be set (see the section titled "Speed setpoint output").

Causes of errors

- A measuring circuit error or drive error is present.
- Setpoints are too high (accelerations, velocities, reducing factors).
- Obstacle in work area (e.g. positioning on a working table)
- Tachogenerator compensation has not been performed correctly for an analog spindle, or a measuring circuit error or drive error is present.

2.2.6 Actual velocity monitoring

Function

This function monitors whether the actual velocity exceeds a permissible limit entered in MD36200 AX_VELO_LIMIT (threshold value for velocity monitoring).

Effectiveness

The actual velocity monitor is operative whenever the active measuring circuit activated via "Position measuring system 1" interface signal (V380x 0001.5) is supplying actual values, i.e. still operating below the limit frequency.

The actual velocity monitoring is active for axes and spindles.

Effect

If the "Threshold for velocity monitoring" is exceeded the following occurs:

- Alarm 25030 "Actual velocity alarm limit" is triggered
- The affected axis/spindle is brought to a standstill with rapid stop (with open position control loop) along a speed setpoint ramp. The braking ramp time is set in MD36610 AX_EMERGENCY_STOP_TIME (duration of the braking ramp for error states).
- If the axis/spindle is assigned to an interpolatory grouping with other axes/spindles, then these are also braked by rapid stop with following error reduction (default for partial position setpoint = 0).

Troubleshooting tips

- Check actual values
- Check position control direction (control sense)
- Check MD36200 AX_VELO_LIMIT (threshold value for velocity monitoring)
- Check speed setpoint cable for analog spindles

2.3 Encoder monitoring functions

2.3.1 Encoder limit frequency monitoring

Function

If the permissible limit frequency of a measuring system entered in MD36300 ENC_FREQ_LIMIT (encoder limit frequency) is exceeded, the positional synchronization (reference point) between the machine and control system will be lost. Correct position control is no longer possible. This state is signaled to the PLC.

Effectiveness

The encoder limit frequency monitoring is always active when the encoder is switched on and is active on axes and spindles.

Effect

When the limit frequency of an encoder is exceeded the following occurs:

- The "Encoder limit frequency exceeded 1" interface signal (V390x 0000.2) is set.
- Spindle continues to run with closed-loop speed control.

If the spindle speed is reduced to a value that causes the encoder frequency to drop below the setting in MD 36302 ENC_FREQ_LIMIT_LOW (% value of MD36300), the spindle is automatically re-synchronized with the reference system of the encoder.

- If the limit frequency is exceeded while the measuring system of a position controlled axis/spindle is active, alarm 21610 "Frequency is exceeded" is signaled.
- The affected axis/spindle is brought to a standstill with rapid stop (with open position control loop) along a speed setpoint ramp. The braking ramp time is set in MD36610 AX_EMERGENCY_STOP_TIME (duration of the braking ramp for error states).
- If the axis is assigned to an interpolatory grouping with other axes, then these are also braked by rapid stop with following error reduction (default for partial position setpoint = 0).

Remedy

After the axes have stopped, the position control is automatically resumed.

Note

The axis affected must be re-referenced.

2.3.2 Zero mark monitoring

Function

The zero mark monitoring checks whether pulses were lost between two zero mark passes of the position actual value encoder. Enter the number of identified zero mark errors at which the monitoring function should respond in MD36310 ENC_ZERO_MONITORING (zero mark monitoring).

Effectiveness

The monitoring function is activated with MD36310 ENC_ZERO_MONITORING. The zero mark error count starts with "0" each time the encoder is switched on.

Effect

If the number of zero mark errors entered in MD36310 ENC_ZERO_MONITORING is reached for a measuring system, alarm 25020 "Zero mark monitoring" is triggered.

The affected axis/spindle is brought to a standstill using a rapid stop (with open position control loop) along a speed setpoint ramp. The braking ramp duration is specified in MD36610 AX_EMERGENCY_STOP_TIME.

If the axis is assigned to an interpolatory grouping with other axes, then these are also braked using a rapid stop with following error reduction (default for partial position setpoint = 0).

Causes of errors

- MD36300 ENC_FREQ_LIMIT (encoder limit frequency) set too high.
- Encoder cable damaged.
- Encoder or encoder electronics defective.

Note

When an error occurs in the measuring circuit, interface signal "Referenced/synchronized 1" (V390x 0000.4) is cancelled; i.e. the axis must be re-referenced.

2.4 Monitoring of static limits

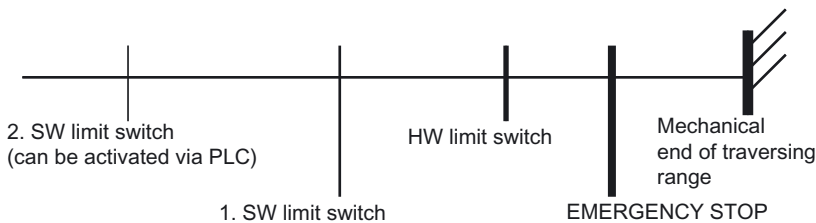


Figure 2-3 Overview of travel limits of a linear axis

2.4.1 Hardware limit switches

Function

Every axis has a hardware limit switch for each traversing direction, which prevents the slide from moving out of the slide bed.

If the hardware limit switch is crossed, the PLC signals this to the NC via IS "Hardware limit switch plus/minus" (V380x 1000.1 or .0) and the movement of all axes is stopped. The braking method can be specified via MD36600 BRAKE_MODE_CHOICE (braking behavior at hardware limit switch).

Effectiveness

HW limit switch monitoring is active after the control has started up in all modes.

Effect

- When a hardware limit switch is passed in either direction, alarm 21614 "Hardware limit switch + or -" is triggered.
- The axis is stopped according to the setting in MD36600 BRAKE_MODE_CHOICE (braking behavior at hardware limit switch).
- If the axis is assigned to an interpolatory grouping with other axes, then these are also stopped according to the method selected in MD36600 BRAKE_MODE_CHOICE (braking behavior at hardware limit switch).
- The direction keys in the approach direction are disabled.

Remedy

- Reset
- Move in the opposite direction (in JOG mode)
- Correct the program

2.4.2 Software limit switches

Function

They are used to limit the maximum traversing range on each individual axis.

There are two pairs of software limit switches for each machine axis. They are defined in the machine axis system using the following machine data:

MD36100 POS_LIMIT_MINUS (1st software limit switch minus)

MD36110 POS_LIMIT_PLUS (1st software limit switch plus)

MD36120 POS_LIMIT_MINUS2 (2nd software limit switch minus)

MD36130 POS_LIMIT_PLUS2 (2nd software limit switch plus)

Effectiveness

- Software limit switch monitoring is activated after reference point approach in all modes.
- The position of the software limit switch can be approached.
- The 2nd software limit switch can be activated via the "2nd software limit switch plus/minus" interface signal (V380x 1000.3 or .2) from the PLC. The change becomes active immediately. The 1st software limit switch plus/minus is then de-activated.
- The SW limit switch monitoring does not function for endlessly turning rotary axes, i.e. if MD30310 ROT_IS_MODULO = 1. (Modulo conversion for rotary axis and spindle)

Effect/reactions

Depending on the mode, different responses to an attempted software limit switch violation are possible:

AUTO, MDA:

- The block that would violate the software limits switches is not started. The previous block is terminated properly.
- Program execution is terminated.
- Alarm 10720 "Software limit switch + or -" is signaled.

JOG:

- The axis stops at the software limit switch position.
- Alarm 10621 "Axis at software limit switch + or -" is signaled.
- The direction keys in the approach direction are disabled.

| |
|---|
| NOTICE |
| Switching over the software limit switch: If the current position lies behind the new software limit switch when the software limit switch is switched over, the axis is decelerated with the maximum permissible axial acceleration. If an axis is involved in interpolation with other axes, these are also decelerated. A contour violation may occur. |

Remedy

- Reset
- Move in the opposite direction (in JOG mode)
- Correct the program

2.4.3 Working area limitation

Function

Working area limits describe the area in which machining is possible. They enable the user to limit the traversing range of the axes in addition to the limit switches.

Reference:

/BP_/ Operation and Programming

A check is made to see whether the tool tip P is within the protected working area. The value entered for the working area limit is the last permissible position of the axis.

MD21020 WORKAREA_WITH_TOOL_RADIUS (allowance for tool radius with working area limits) can be set to determine whether the monitor is to take account of the tool radius.

One pair of values (minus/plus) is possible for each axis to describe the protected working area.

Define working area limits

The working area limits can be defined or changed in two ways:

- Via the operator panel in the "Parameters" operating area using the following setting data:
SD43430 WORKAREA_LIMIT_MINUS (Working area limitation minus)
SD43420 WORKAREA_LIMIT_PLUS (Working area limitation plus)

Changes during Automatic mode are only possible in the reset state and then take effect immediately. Changes during JOG mode are always possible, but only take effect with a new movement.

- In a program with G25/G26. Changes are immediately effective.

A programmed motion has priority, it overwrites the value entered in the setting data and is retained after RESET and End of Program.

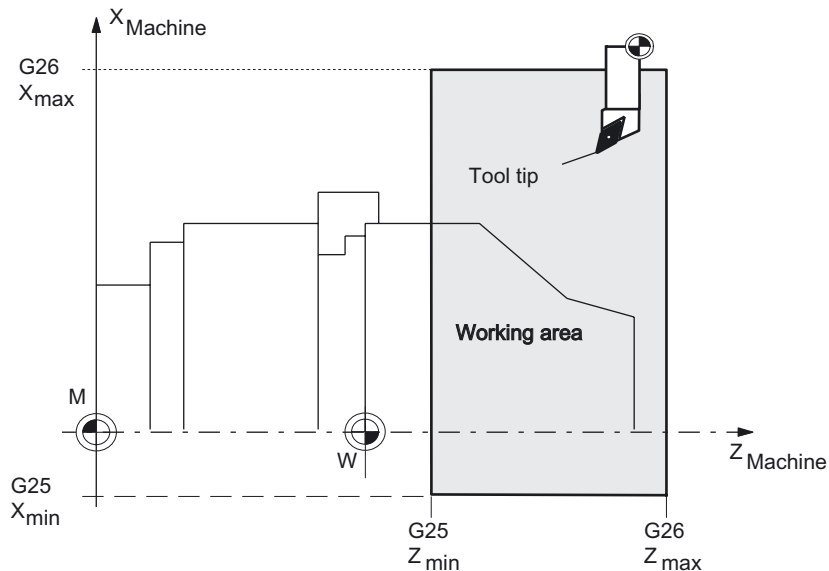


Figure 2-4 Working area limitation using the example of the turning machine

Effectiveness

- The working area limitation can be activated via SD43410 WORKAREA_MINUS_ENABLE, SD43400 WORKAREA_PLUS_ENABLE (working area limitation in the negative or positive direction active) and takes effect after reference point approach.
- During program execution, the working area limitation can be activated with the modal G code "WALIMON" and deactivated with "WALIMOF".
- The working area limitation does not function for endlessly turning rotary axes, i.e. if MD30310 ROT_IS_MODULO = 1. (Modulo conversion for rotary axis and spindle).

Effect/reaction

Depending on the mode, the response to an attempted violation of the working area limitation can be different:

AUTO, MDA

- The block that violates the working area limitation is not started. The previous block is terminated properly.
- Program execution is terminated.
- Alarm 10730 "Working area limitation + or -" is set.

JOG:

- The axis is positioned at the working area limitation and then stopped.
- Alarm 10631 "Axis at working area limitation + or -" is signaled.
- The direction keys in the approach direction are disabled.

Remedy

- Reset
- Check the working area limit in the part program (G25/G26) or in the setting data.
- Move in the opposite direction (in JOG mode)

2.5 Supplementary conditions

To ensure that the monitoring functions respond correctly, it is important that the correct values are entered in the following machine data:

General:

- MD31030 LEADSCREW_PITCH (leadscrew pitch)
- Gear ratio (load gearbox):
 - MD31050 DRIVE_AX_RATIO_DENOM (load gearbox denominator)
 - MD31060 DRIVE_AX_RATIO_NUMERA (load gearbox numerator)
- Gear ratio (encoder), possibly for spindle:
 - MD31070 DRIVE_ENC_RATIO_DENOM (measuring gearbox denominator)
 - MD31080 DRIVE_ENC_RATIO_NUMERA (measuring gearbox numerator)
- MD32810 EQUIV_SPEEDCTRL_TIME
(Equivalent time constant speed control loop for feedforward control)
- Encoder resolution
 - MD31020 ENC_RESOL[0] (encoder pulses per revolution)

The associated machine data are described in Chapter "Velocities, Setpoint/Actual Value Systems, Closed-Loop Control (G2)"

For analog spindle only:

- Output voltage / output speed relation
MD32260 RATED_VELO (rated motor speed)
MD32250 RATED_OUTVAL (rated output voltage)

2.6 Data lists

2.6.1 Machine data

| Number | Identifier | Name |
|-----------------------|--|---|
| Channel-specific | | |
| 21020 | WORKAREA_WITH_TOOL_RADIUS | Allowance for tool radius with working area limitation |
| Axis/spindle-specific | | |
| 30310 | ROT_IS_MODULO | Modulo conversion for rotary axis and spindle |
| 32000 | MAX_AX_VELO | Maximum axis velocity |
| 32200 | POSCTRL_GAIN [n] | Servo gain factor Kv |
| 32250 | RATED_OUTVAL | Rated output voltage |
| 32260 | RATED_VELO | Rated motor speed |
| 32300 | MAX_AX_ACCEL | Axis acceleration |
| 32810 | EQUIV_SPEEDCTRL_TIME [n] | Equivalent time constant speed control loop for feedforward control |
| 35160 | SPIND_EXTERN_VELO_LIMIT | Spindle speed limitation via PLC |
| 36000 | STOP_LIMIT_COARSE | Exact stop coarse |
| 36010 | STOP_LIMIT_FINE | Exact stop fine |
| 36020 | POSITIONING_TIME | Time delay exact stop fine |
| 36030 | STANDSTILL_POS_TOL | Standstill tolerance |
| 36040 | STANDSTILL_DELAY_TIME | Delay time standstill monitoring |
| 36050 | CLAMP_POS_TOL | Clamping tolerance with "Clamping active" interface signal |
| 36060 | STANDSTILL_VELO_TOL | Maximum velocity/speed "Axis/spindle stationary" |
| 36100 | POS_LIMIT_MINUS | 1. Minus software limit switch |
| 36110 | POS_LIMIT_PLUS | 1. Plus software limit switch |
| 36120 | POS_LIMIT_MINUS2 | 2. Minus software limit switch |
| 36130 | POS_LIMIT_PLUS2 | 2. Plus software limit switch |
| 36200 | AX_VELO_LIMIT [n] | Threshold value for velocity monitoring |
| 36210 | CTRL_OUT_LIMIT [n] | Maximum speed setpoint |
| 36300 | ENC_FREQ_LIMIT <input type="checkbox"/> n <input type="checkbox"/> | Encoder frequency limit |
| 36302 | ENC_FREQ_LIMIT_LOW | Encoder limit frequency resynchronization |

| Number | Identifier | Name |
|--------|--------------------------|---|
| 36310 | ENC_ZERO_MONITORING [n] | Zero mark monitoring |
| 36400 | CONTOUR_TOL | Tolerance band contour monitoring |
| 36500 | ENC_CHANGE_TOL | High backlash values / Maximum tolerance for actual position value changeover |
| 36600 | BRAKE_MODE_CHOICE | Braking behavior at hardware limit switch |
| 36610 | AX_EMERGENCY_STOP_TIME | Length of the braking ramp for error states |
| 36620 | SERVO_DISABLE_DELAY_TIME | Cutout delay controller enable |

2.6.2 Setting data

| Number | Identifier | Name |
|-----------------------|-----------------------|--|
| Axis/spindle-specific | | |
| 43400 | WORKAREA_PLUS_ENABLE | Working area limitation active in positive direction |
| 43410 | WORKAREA_MINUS_ENABLE | Working area limitation active in negative direction |
| 43420 | WORKAREA_LIMIT_PLUS | Working area limitation plus |
| 43430 | WORKAREA_LIMIT_MINUS | Working area limitation minus |

2.6.3 Interface signals

| Number | .Bit | Name |
|-----------------------|---------|---|
| Axis/spindle-specific | | |
| V380x 0001 | .5 | Position measuring system 1 |
| V380x 0002 | .3 | Clamping in progress |
| V380x 0003 | .6 | Velocity/spindle speed limitation |
| V380x 1000 | .0 / .1 | Hardware limit switch minus / hardware limit switch plus |
| V380x 1000 | .2 / .3 | 2. Software limit switch minus / software limit switch plus |
| V390x 0000 | .2 | Encoder limit frequency exceeded 1 |
| V390x 0000 | .4 | Referenced/synchronized 1 |

Continuous Path Mode, Exact Stop and LookAhead (B1)

3

3.1 Brief description

For continuous path control, the CNC processes a part program block by block. Only when the functions of the current block have been completed is the next block processed. Various requirements with respect to machining or positioning require different block change criteria. There are two ways that the path axes can behave at block boundaries.

The first way is called "exact stop" and means that all path axes must have reached the set target position depending on an exact-stop criterion before the next block change is initiated. To be able to fulfill the criterion, the path axes must reduce the path velocity at every block change which, however, delays the block change.

The second way is called "continuouspath mode" and it attempts to avoid deceleration of the path velocity at the block boundary in order to change to the next block with as little change of path velocity as possible.

"LookAhead" is a procedure in continuous path mode that achieves velocity control with LookAhead over several NC part program blocks.

3.2 General

Machine axes that are related interpolatively must have the same dynamic response, i.e. the same following error at any given velocity.

The term path axes refers to all machining axes which are controlled by the interpolator calculating the path points in such a manner that:

- All the axes involved start at the same time
- All the axes involved travel with the correct velocity ratios
- All the axes reach the programmed target position at the same time

The acceleration rates of the individual axes may vary depending on the path, e.g. circular path.

Path axes can be geometry axes and special axes (e.g. workpiece turning axes that are involved in the workpiece machining process).

Velocity for zero cycle blocks

The term zero cycle is applied to blocks whose path length is shorter than the distance that can be traveled on the basis of the programmed set feedrate and the interpolator cycle (time). For reasons of precision the velocity is reduced until at least one interpolator cycle is required for the distance. The velocity is then equal to or less than the quotient of the path length of the block and the IPO cycle.

Stop for synchronization

Regardless of whether exact stop or continuous path mode is selected, the block change can be delayed by synchronization processes which can stop the path axes. In exact stop mode, the path axes are stopped at the end of the current block. In continuous path mode, the path axes are stopped at the next block end point at which they can be decelerated without violating their deceleration limits. The following synchronization processes cause axes to stop.

- PLC acknowledgment

If acknowledgment by the PLC is required for an auxiliary function that is output before or after the end of motion, the axes stop at the end of the block.

- Missing following blocks

If following blocks are conditioned too slowly (e.g. "External processing") the axes stop at that last possible block boundary.

- Emptying of the buffer

If the NC part program requests that the run-in be synchronized with the main run (empty the buffer, e.g. STOPRE), this involves an implicit blockrelated velocity reduction or exact stop.

Stopping because of synchronization does not cause contour violations. However, stopping is undesirable, especially in continuous path mode because it can cause backing off.

3.3 Exact stop

With the exact stop function (G60, G9), all the path axes must reach the programmed block end point. Only when all path axes have reached the exact stop criterion is the block change performed. The velocity at the block transition is practically zero.

That is:

- The path axes at the block end point are decelerated almost to rest without overshoot.
- The delay for fulfilling the exact stop criterion prolongs the machining time.
- The delay for fulfilling the exact stop criterion can cause backing off.

The use of the exact stop function is suitable for precise traversing of contours.

Exact stop is not suitable if

- Exact traversing of the contour on the basis of the criterion (e.g. exact stop fine) can deviate from the programmed contour in order to achieve faster machining.
- An absolutely constant velocity is required.

Activate exact stop

The "Exact stop" function can be selected in the NC part program by command G60 or G9. G60 is modal, G9 is non-modal. G9 is used if continuous path mode is to be interrupted. Both exact stop functions only function with the selected exact stop criterion (G601, G602). The "exact stop" function is de-selected with the continuous path mode function (G64).

Exact-stop criteria

- Exact stop fine: G601

This criterion is applied to monitor whether the actual/setpoint position deviation of the axis has remained within a specific distance. The value of the permissible distance is stored in MD36010 STOP_LIMIT_FINE (exact stop fine)

- Exact stop coarse: G602

Functions as exact stop fine, although the monitoring window is stored in MD36000 STOP_LIMIT_COARSE (exact stop coarse). To permit a faster block change than with the exact stop fine criterion, the exact stop coarse criterion is set to be larger than the exact stop fine criterion.

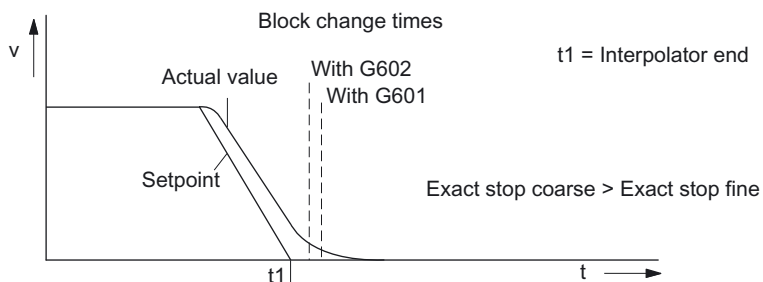


Figure 3-1 Block change depending on exact-stop criteria

Interpolator end

Interpolator end is achieved when the interpolator has calculated the setpoint velocity of the axes from zero for an interpolation cycle. However, the actual positions of the path axes have not reached the target (following error).

Irrespective of continuous-path mode or the active exact-stop criteria for the exact-stop function, "interpolator end" transfers the auxiliary functions present in the block to the PLC if they are to be output after the end of motion.

3.4 Continuous-path mode

3.4.1 General

In continuous path mode, the path velocity is not decelerated for the block change in order to permit the fulfillment of an exact stop criterion. The objective of this mode is to avoid rapid deceleration of the path axes at the block-change point so that the axis velocity remains as constant as possible when the program moves to the next block. To achieve this objective, the "LookAhead" function is also activated when continuouspath mode (G64) is selected.

3.4 Continuous-path mode

Continuouspath mode causes:

- Rounding of the contour.
- Shorter machining times through elimination of braking and acceleration processes that are required to comply with the exact-stop criterion.
- Improved cutting conditions as the velocity is more uniform.

The continuous-path mode is suitable if a contour is to be traversed as quickly as possible.

Continuous-path mode is suitable if:

- A contour is to be traversed precisely.
- An absolutely constant velocity is required.

Implicit exact stop

In some cases, an exact stop needs to be generated in continuouspath mode to allow the execution of subsequent actions. In such situations, the path velocity is reduced to zero.

- If auxiliary functions are output before the traverse motion, the previous block is only terminated when the selected exact-stop criterion is fulfilled.
- If auxiliary functions are to be output after the traverse motion, they are output after the interpolator end of the block.
- If an executable block contains no travel information for the path axes, the previous block is terminated on reaching the selected exact stop criterion.
- A block is terminated on interpolator end if the following block contains the changeover of the acceleration profile BRISK/SOFT.
- If the function "Empty buffer" (STOPRE) is programmed, the previous block is terminated when the selected exact stop criterion is reached.

Velocity = 0 in continuouspath mode

Regardless of the implicit exact stop response, the path motion is braked down to zero velocity at the end of the block in cases where:

- The time taken to position a spindle programmed with SPOS is longer than the travel time of the path axes. The block change is carried out when the "exact stop fine" of the positioning spindle is reached.
- A synchronization process needs to be carried out (see Section "General").

Auxiliary function output during traversal

If the traversal time is not sufficient due to the programmed path length and velocity of the block with auxiliary function output, the path velocity for the block is reduced such that the acknowledgment of the auxiliary function can arrive with a PLC cycle time.

If the acknowledgment is not received within one PLC cycle time, the following prepared block cannot be processed and the axes are braked to rest with setpoint = 0 (without considering the acceleration limits).

If the acknowledgment is not received by the end of the block in long blocks in which the velocity has not needed to be reduced on account of the PLC acknowledgment time, the velocity is maintained until the end of the block and then reduced as described above.

If the acknowledgment arrives while the axis is decelerating, the axis is not accelerated back up to the requested velocity.

3.4.2 Velocity reduction according to overload factor

Function

This function lowers the path velocity in continuous path mode until the nontangential **block transition** can be traversed in one interpolation cycle whilst respecting the deceleration limit and taking an overload factor into account. With the reduced velocity, axis-specific jumps in velocity are produced with a nontangential contour at the block transition. The jump in velocity prevents the path velocity dropping to zero. This jump is performed if the axial velocity was reduced with the axial acceleration to a velocity from which the new setpoint can be reached with the jump.

The magnitude of the setpoint jump can be limited using an overload factor. Because the magnitude of the jump is axial, the minimum jump of the path axes which are active during the block change is considered during block transition. With a practically tangential block transition, the path velocity is not reduced if the permissible axial accelerations are not exceeded. In this way, very small angular changes in the contour can be overtraveled directly.

Overload factor

The overload factor restricts step changes in the machine axis velocity at the block transition. To ensure that the velocity jump does not exceed the maximum load on the axis, the jump is derived from the acceleration of the axis. The overload factor indicates the extent by which the acceleration of the machine axis, which is set in MD32300 MAX_AX_ACCEL (axis acceleration), may be exceeded for an IPO cycle.

The velocity jump is the product of:

axis acceleration * (overload factor-1) * interpolator cycle. The overload factor is 1.2.

Factor 1.0 means that only tangential transitions with finite velocity can be traversed. For all other transitions, the velocity is reduced to zero by changing the setpoint.

Selection and deselection of velocity reduction

Continuous-path mode with velocity reduction according to overload factor can be selected modally in every NC part program block by means of program code G64 (BRISK active, not SOFT).

3.4 Continuous-path mode

Continuous path mode G64 can be

- interrupted non-modally when exact stop G9 is selected,
- de-selected when exact stop G60 is selected.

3.4.3 Jerk limiting along the path through velocity reduction

Introduction

With the jerk limiting along the path, another method of influencing the continuous-path mode is introduced. While the "Velocity reduction according to overload factor" function limits the rate of velocity change, the "Jerk limitation on path" function described here limits the acceleration changes (jerks).

When sections of the contour consisting of blocks (e.g. circle straight line transitions) are machined, step changes in the acceleration rate occur at the **block transition** in continuous path mode.

Reducing jerk

The severity of such jerks can be reduced by decreasing the path velocity at transitions between blocks containing different degrees of curvature. A smoother transition is thus achieved between the contour sections.

Jerk limit

The user specifies the maximum jerk, which may occur on a path axis during a block transition, with MD32432 PATH_TRANS_JERK_LIM (maximum axis-specific jerk of a path axis at the block transition).

Activating

Jerk limiting at block transitions becomes active if continuous path mode is programmed with G64 and SOFT acceleration characteristics. MD32432 PATH_TRANS_JERK_LIM must contain a positive value.

3.4.4 Machine axis-specific jerk limiting

Function

The axis-specific machine data MD32431 MAX_AX_JERK[...] can be used to set individual changes in acceleration for each machine axis, like those that can already be set for acceleration limits in machine data MD32300 MAX_AX_ACCEL.

MD32431 MAX_AX_JERK acts on the axes interpolated by the path if **SOFT** (smooth acceleration curve) is active **within a block**.

A basic distinction is made between the axis acceleration curve within a block and at the transition between two blocks.

Advantages

The deployment of axis-specific machine data for the path offers the following advantages:

- Immediate allowance is made in the interpolation for the dynamic response of the axes, which can then be fully utilized for each axis.
- Jerk limitation for separate axes is performed not just in linear blocks, but also in curved contours.

Please refer to Chapter "Acceleration (B2)" for more information on the subject of "jerk limiting".

3.5 LookAhead

Function

LookAhead is a procedure in continuous path mode (G64) that achieves velocity control with LookAhead over several NC part program blocks beyond the current block.

Without LookAhead: If the program blocks only contain very small paths, a velocity per block is achieved that permits deceleration of the axes at the block end point without violating acceleration limits. This means that the programmed velocity was not actually reached although a sufficient number of prepared blocks with virtually tangential path transitions were available.

With the LookAhead function: It is possible to plan the acceleration and deceleration phase with approximately tangential path transitions over several blocks in order to achieve a higher feedrate with shorter distances. Deceleration to velocity limits is possible with LookAhead such that violation of the acceleration and velocity limit is prevented.

3.5 LookAhead

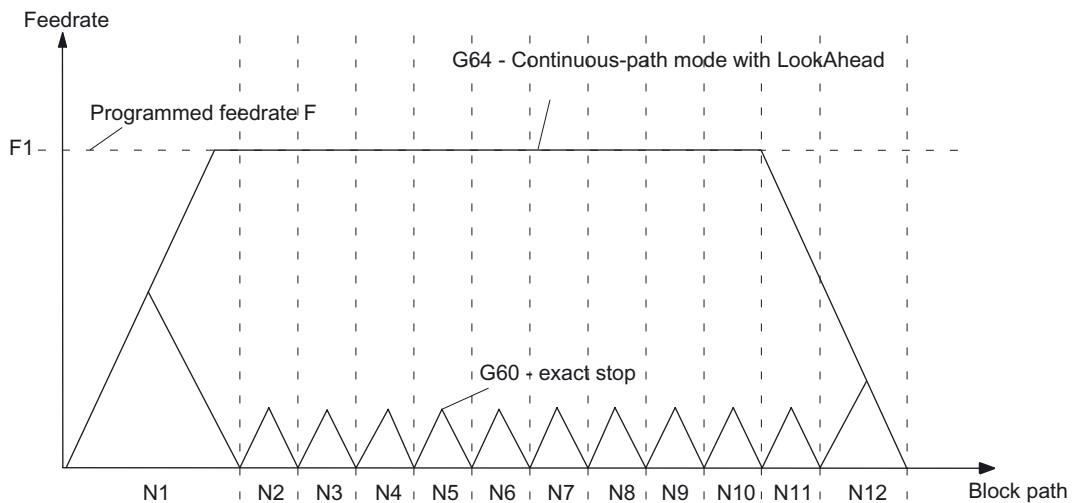


Figure 3-2 Comparison of the G60 and G64 velocity behavior with short travels in the blocks

LookAhead takes plannable velocity limits into consideration such as:

- Velocity limit in the block
- Acceleration limit in the block
- Velocity limit on block transition
- Synchronization with block change at block transition
- Exact stop at block end during termination

Operating principle

LookAhead functionality is available only for path axes, but not for the spindle.

For safety reasons, the velocity at the end of the last prepared block must initially be assumed to be zero because the next block might be very small or be an exact-stop block and the axes must have been stopped by the end of the block. With a series of blocks with high set velocity and very short paths, the speed can be increased in each block depending on the velocity value currently calculated by the LookAhead function in order to achieve the required set velocity. After this it can be reduced so that the velocity at the end of the last block considered by the LookAhead function can be zero. This results in a sawtooth-shaped velocity profile which can be avoided by reducing the set velocity for the number of blocks considered by the LookAhead function (fixed value).

Velocity profiles

In addition to the fixed, plannable velocity limitations, LookAhead can also take account of the programmed velocity. This makes it possible to achieve a lower velocity by applying LookAhead beyond the current block.

Following block velocity

One possible velocity profile contains the determination of the following block velocity. Using information from the current and the following NC block, a velocity profile is calculated from which, in turn, the required velocity reduction for the current override is derived. The calculated maximum value of the velocity profile is limited by the maximum path velocity.

With this function it is possible to initiate a speed reduction in the current block taking override into account such that the lower velocity of the following block can be achieved. If the reduction in velocity takes longer than the travel time of the current block, the velocity is further reduced in the following block. Velocity control is only ever considered for the following block.

Selection and deselection of LookAhead

If the continuous-path mode (G64) is selected LookAhead is called and de-selected/interrupted with G60/G9.

3.6 Data lists

3.6.1 Machine data

| Number | Identifier | Name |
|-----------------------|---------------------------|--|
| Channel-specific | | |
| 29000 | LOOKAH_NUM_CHECKED_BLOCKS | Number of blocks considered by the LookAhead function |
| Axis/spindle-specific | | |
| 32431 | MAX_AX_JERK | Maximum axis-specific jerk for path movement |
| 32432 | PATH_TRANS_JERK_LIM | Maximum axis-specific jerk for path movement at block transition |
| 36000 | STOP_LIMIT_COARSE | Exact stop coarse |
| 36010 | STOP_LIMIT_FINE | Exact stop fine |
| 36020 | POSITIONING_TIME | Delay time exact stop fine |

3.6.2 Interface signals

| Number | Bit | Name |
|-----------------------|-----|---|
| Channel-specific | | |
| V3300 0004 | .3 | All axes stationary |
| Axis/spindle-specific | | |
| V390x 0000 | .6 | Position reached with exact stop coarse |
| V390x 0000 | .7 | Position reached with exact stop fine |

Acceleration (B2)

4.1 Acceleration profiles

Abrupt acceleration changes

With the v/t-linear control of the axis velocity that is normally applied, the motion is controlled such that the acceleration rate changes abruptly over time. With the discontinuous, stepped acceleration, jerkfree starting and braking of the axes is not possible, but a timeoptimized velocity/time profile can be implemented.

Acceleration with jerk limitation

The jerk is the change of acceleration over time. For jerk-limited acceleration the maximum acceleration is not abrupt, but is specified by a ramp. Because of the softer acceleration progression, the traverse time is longer than with abrupt acceleration for the same distance, velocity and acceleration. This time loss can be compensated for by setting a higher acceleration for the axes.

However, it has the following advantages:

- Reduced wear to mechanical parts of the machine
- Reduction of the excitation of highfrequency, difficult to control vibrations of the machine.

4.2 Jerk limitation on interpolator level

Selection and deselection of jerk-limited acceleration

MD32431 MAX_AX_JERK (maximum axis-specific jerk during path motion) can limit the change in acceleration per machine axis individually. It only acts on the axes interpolated by the path when SOFT is active. Jerk limitation is implemented entirely on the interpolator level.

Acceleration with jerk limitation is activated by:

The program code **SOFT** in the part program. SOFT is modal and causes deselection of the abrupt acceleration profile (BRISK). If SOFT is programmed in a block with path axes, the previous block is ended with exact stop.

Acceleration with jerk limitation (SOFT) is deactivated by:

The program code **BRISK** in the part program. BRISK is modal. If path axes are programmed in a block with BRISK, the previous block is ended with exact stop. BRISK activates the profile with abrupt acceleration changes associated with v/tlinear velocity control.

Applicability

Path-related jerk limitation is available for interpolating path axes in operating modes AUTO and MDA. The SOFT and BRISK acceleration profiles can be used in traverse modes exact stop G9, G60, continuouspath modes G64, and with LookAhead. The profiles are also active with the dry run feedrate function. With alarms that trigger a rapid stop, both acceleration profiles are inactive.

Further information about velocity, acceleration and jerk whilst traversing in continuous path mode and at block transitions can be found in Chapter "Continuous Path Mode, Exact Stop and LookAhead (B1)".

Note

We recommend setting the following machine data for each axis with the same values: MD32431 MAX_AX_JERK and MD32432 PATH_TRANS_JERK_LIM (maximum axis-specific jerk for path movement at block transition)

4.3 Jerk limitation in JOG mode

The jerk limitation is active for axes in JOG mode during

- jogging
- handwheel jogging
- repositioning.

The jerk limitation is not active during

- reference point approach with
- alarms that initiate a rapid stop.

Jerk limitation can be determined for specific axes. The acceleration response corresponds with the SOFT acceleration profile of path-related jerk limitation. This limitation cannot be deselected for the axes in the relevant modes.

The axes for which jerk limitation is to be programmed can be selected with MD32420 JOG_AND_POS_JERK_ENABLE. The permissible axis-specific maximum jerk is stored in MD32430 JOG_AND_POS_MAX_JERK.

4.4 Percentage acceleration correction, ACC

Function

Certain program sections can require the axis and spindle acceleration set via the machine data to be changed using the program. This programmable acceleration is a percentage acceleration correction.

ACC[channel axis name] = percentage

The **ACC** command in the program allows you to set a percentage for each axis (e.g. X) or spindle (S1) of > 0% and ≤ 200%. The axis interpolation is then carried out with this proportional acceleration.

The acceleration defined in the axis-specific MD32300 MAX_AX_ACCEL represents the reference for one axis (100%).

For the spindle the reference (100%) corresponds to the active spindle mode and gear stage (n = 1 ... 5):

- MD35200 GEAR_STEP_SPEEDCTRL_ACCEL[n] in open-loop control mode
- MD35210 GEAR_STEP_POSCTRL_ACCEL[n] in positioning mode.

Examples: N10 ACC[X]=80; 80% acceleration for the X axis

Effectiveness

The limitation is effective in all interpolation types of the AUTOMATIC and MDA modes. The limitation is not active in JOG mode and during reference point approach.

The value assignment ACC [...] = 100 deactivates the correction (100% of the MD values) as well as RESET and program end.

The programmed value is also active with dry run feedrate.

Error states

The acceleration limitation is not active in error states that cause a rapid stop with an open position control loop (because the axis is stopped according to a braking ramp of the set speed).

Note

A programmable value > 100% can only be carried out if the drives have appropriate reserves - otherwise alarm messages are output.

4.5 Data lists

Machine data

| Number | Identifier | Name |
|---------------|-------------------------|---|
| Axis-specific | | |
| 32300 | MAX_AX_ACCEL | Axis acceleration |
| 32420 | JOG_AND_POS_JERK_ENABLE | Enabling axis-specific jerk limitation |
| 32430 | JOG_AND_POS_MAX_JERK | Axis-specific jerk |
| 32431 | MAX_AX_JERK | Maximum axis-specific jerk during path movement |
| 32432 | PATH_TRANS_JERK_LIM | Maximum axis-specific jerk during path movement at the block transition |

Travel to fixed stop (F1)

Note

This function is not available with version T/M value.

5.1 Brief description

Application range

The travel to fixed stop (FXS = Fixed Stop) function can be used to establish defined forces for clamping workpieces, such as those required for sleeves and grippers. The function can also be used for the approach of mechanical reference points. With sufficiently reduced torque, it is also possible to perform simple measurement operations without connecting a probe.

Travel to fixed stop can be performed simultaneously for several axes and parallel to the movement of other axes.

The fixed stop can be approached on a path (straight line, circle).

Availability

The "travel to fixed stop" function is available when MD37000 FIXED_STOP_MODE (travel to fixed stop mode) = 1 is set. The function can then be started from the NC program with command "FXS[x]=1".

5.2 Functionality

Programming

Travel to fixed stop is selected with command

- **FXS**[machine axis identifier] = 1 selected.
- **FXS**[machine axis identifier] = 0 de-selected.

The clamping torque is set with the command: **FXST**[Machine axis identifier] = <torque>. It is given as a percentage of the static torque of the drive or a percentage of the nominal motor torque for MSD.

The width of the fixed stop monitoring window is set with the command **FXSW**[Machine axis identifier] = <window width>. Unit: mm, inches or degrees - depending on the basic measurement system, linear or rotary axis.

The commands are modal. The travel path and the command which activates the function must be programmed **in the same block**

Note

The programming of the machine axis identifiers according to MD10000 AXCONF_NAME_TAB is permitted for FXS... and should be used preferentially.

The machine axis identifiers according to MD20070 AXCONF_CHANAX_NAME_TAB are permitted for FXS... provided this is assigned to precisely one axis - e.g. no rotation is activated in the coordinate system.

Axis-specific setting data/machine data

If no **monitoring window** is programmed, the value from SD43520 FIXED_STOP_WINDOW applies.

If a value is programmed, this is applied and adopted by the SD43520. Initially, the SD is loaded using the value from MD37020 FIXED_STOP_WINDOW_DEF.

If no **clamping torque** is programmed, the value from SD43510 FIXED_STOP_TORQUE applies.

If a value is programmed, this is applied and adopted by the SD43510. Initially, the SD is loaded using the value from MD37010 FIXED_STOP_TORQUE_DEF.

The **selection/de-selection** of the function is made in the program via FXS[X1]=1 / =0 only. This value is also written in SD43500 FIXED_STOP_SWITCH (travel to fixed stop selection).

Programming examples

```
X250 Y100 F100 FXS[X1]=1           ; FXS selected for X1 machine axis
                                   ; clamping torque and window width from SDs
X250 Y100 F100 FXS[X1]=1           ; FXS selected for X1 machine axis
      FXST[X1] = 12.3                ; clamping torque 12.3% and window width from SDs
X250 Y100 F100 FXS[X1]=1           ; FXS selected for X1 machine axis
      FXST[X1]=12.3 FXSW[X1]=2       ; clamping torque 12.3% and window width 2 mm
X250 Y100 F100 FXS[X1]=1           ; FXS selected for X1 machine axis
      FXSW[X1] = 2                    ; clamping torque from SD, window width 2 mm
```

Note

It is not permissible to program a new position for an axis if the "Travel to fixed stop" function has already been activated for an axis/spindle (not an analog spindle). The spindle must be switched to position-controlled mode before the function is selected.

System variable \$AA_FXS[X1] for status

This system variable provides the "Travel to fixed stop" status for the axis specified:

- Value =
- 0: Axis not at fixed stop.
 - 1: Fixed stop successfully approached (axis is in fixed stop monitoring window).
 - 2: Approach to fixed stop has failed (axis is not at fixed stop).
 - 3: Travel to fixed stop activated.
 - 4: Fixed stop detected.
 - 5: Travel to fixed stop will be de-selected, the de-selection is not yet completed.

Query of the system variables in the part program initiates a preprocessing stop.

As a result of the status query in the part program, it is possible, for example, to react to an erroneous operational sequence of the "Travel to fixed stop" function.

Example of function

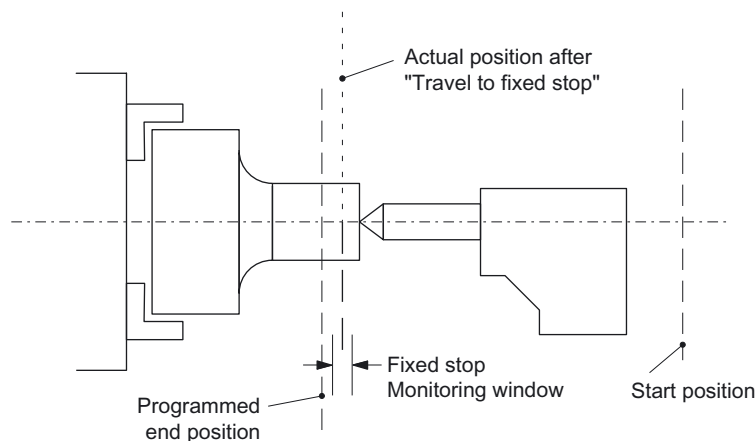


Figure 5-1 Example for travel to fixed stop - sleeve is pressed onto the workpiece

Selection

The NC detects when preparing the block that the function "Travel to fixed stop" is selected via the command `FXS[x]=1` and signals the PLC via the IS "Activate travel to fixed stop" that the function has been selected.

If MD37060 `FIXED_STOP_ACKN_MASK` (register PLC acknowledgments for travel to fixed stop) is set accordingly, the NC waits for acknowledgment by the PLC through IS "Activate travel to fixed stop".

The programmed target position is then approached from the start position at the programmed velocity. The fixed stop must be located between the start and target positions of the axis/spindle. Any programmed torque limit is effective from the block start, i.e. the fixed stop is also approached at a reduced torque. Allowance for this limitation is made in the NC through an automatic reduction in the acceleration rate.

If no torque was programmed in the block or since the beginning of program, the value entered in the axis-specific MD37010 FIXED_STOP_TORQUE_DEF (default setting for clamping torque) applies.

Fixed stop reached

As soon as the axis comes into contact with the mechanical fixed stop (workpiece), the closedloop control in the drive raises the torque so that the axis can move on. The torque rises up to the programmed limit value and then remains constant.

The status "fixed stop reached" can, depending on the MD37040 FIXED_STOP_BY_SENSOR (fixed stop recognition by sensor), be determined in the following ways:

- FIXED_STOP_BY_SENSOR = 0

The status "fixed stop reached" is active when the axis-specific contour deviation (= difference between actual and expected following error) has exceeded the value of the MD37030 FIXED_STOP_THRESHOLD (threshold for fixed stop recognition).

- FIXED_STOP_BY_SENSOR = 1

External sensor transmits "Fixed stop reached" status to NC via the PLC by means of IS "Sensor fixed stop".

- FIXED_STOP_BY_SENSOR = 2

The status "Fixed stop reached" is provided if either this status has been determined by the contour monitoring function or if the external sensor signals this status in response to signal edge change 0 → 1.

Internal processes

Once the NC has detected the "Fixed stop reached" status, it deletes the distancetogo and the position setpoint is made to follow. The controller enabling command remains active.

The PLC is then informed of the state by means of IS "Fixed stop reached".

If MD37060 FIXED_STOP_ACKN_MASK is set accordingly, then the NC waits for acknowledgment by the PLC in the form of IS "Acknowledge fixed stop reached".

The NC then executes a block change or considers the positioning motion to be completed, but still leaves a setpoint applied to the drive actuator to allow the clamping torque to take effect.

The fixed stop monitoring function is activated as soon as the stop position is reached.

Monitoring window

If no torque was programmed in the block or since the beginning of program, the value entered in the MD37010 FIXED_STOP_TORQUE_DEF (default setting for clamping torque) applies.

If the axis leaves the position it was in when the fixed stop was detected, then alarm 20093 "Fixed stop monitoring has responded" is displayed and the "Travel to fixed stop" function deselected.

The window must be selected by the user such that the alarm is activated only when the axis leaves the fixed stop position.

Enabling the fixed stop alarms

MD37050 FIXED_STOP_ALARM_MASK can suppress the output of the following alarms:

- 20091 "Fixed stop not reached"
- 20094 "Fixed stop aborted"

Fixed stop is not reached

If the programmed end position is reached without the "Fixed stop reached" status being recognized, then the torque limit in the drive is cancelled and IS "Activate travel to fixed stop" is reset.

Depending on MD37060 FIXED_STOP_ACKN_MASK, the NC waits for the PLC to acknowledge by resetting IS "Activate travel to fixed stop"; the program then advances to the next block.

Function abort

If the "Travel to fixed stop" function is aborted owing to a pulse cancellation command, cancellation of PLC acknowledgments or a Reset in the approach block, the display or suppression of alarm 20094 can be controlled via MD37050 FIXED_STOP_ALARM_MASK.

Abort without alarm

The PLC can initiate travel to fixed stop without an alarm in the approach block (for example, when the operator actuates a key) if alarm 20094 is suppressed in MD37050 FIXED_STOP_ALARM_MASK.

The Travel to fixed stop function is deselected in response to both "Fixed stop not reached" and "Fixed stop aborted".

Interrupts

- If the fixed stop position is not reached when the function is active, alarm 20091 "Fixed stop not reached" is output and a block change executed.
- If a traversing request (e.g. from the part program or from the operator panel) is provided for an axis after the fixed stop has been reached, the alarm 20092 "Travel to fixed stop still active" is output and the axis is not moved.
- If after reaching the fixed stop an axis is pushed out of position more than the indicated value in SD43520 FIXED_STOP_WINDOW (fixed stop monitoring window), the alarm 20093 "fixed stop monitoring has responded" is output; the "travel to fixed stop" function is then deselected for this axis and system variable \$AA_FXS[x] is set to 2.

Sequence in case of a fault or abnormal termination

The IS "Activate travel to fixed stop" is reset.

Depending on MD37060 FIXED_STOP_ACKN_MASK, the NC waits for the PLC to acknowledge by resetting IS "Activate travel to fixed stop". The torque limitation is then cancelled and a block change executed.

Deselection

The NC detects deselection of the function on the basis of command FXS[x]=0. Then, a preprocessing stop (STOPRE) is initiated internally, since it can not be foreseen where the axis will be after deselection.

The torque limitation and monitoring of the fixed stop monitoring window are cancelled. IS "Activate travel to fixed stop" and "Fixed stop reached" are reset.

Depending on MD37060 FIXED_STOP_ACKN_MASK, the NC waits for the PLC to acknowledge by resetting IS "Activate travel to fixed stop" and /or "Acknowledge fixed stop reached".

The axis will then change to position control. The followup mode of the position setpoints is ended and a synchronization to the new actual position is carried out.

A programmed traverse motion must then be executed. This motion must lead away from the fixed stop or else the stop or even the machine may sustain damage.

A block change is executed after the target position has been reached.

Multiple selection

A selection may only be carried out once. If the function is called once more due to faulty programming (FXS[axis] = 1), the alarm 20092 "Travel to fixed stop still active" is initiated.

Changing clamping torque and monitoring window

The commands FXST[x] and FXSW[x] can be used to change the clamping torque and the fixed stop monitoring window in the part program. The changes take effect before traversing movements in the same block.

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

Ramp for the torque limit

A ramp for changing the torque is specified in MD37012 FIXED_STOP_TORQUE_RAMP_TIME. This determines the length of the ramp until the **new** torque limit is reached.

Without ramp

The torque limit is changed without taking into account the ramp if:

- FXS with (FXS[]=1) is **activated** to make sure that the reduction takes effect immediately
- The drive needs to be de-energized as quickly as possible in the event of a fault.

Response to pulse inhibit for the drive

MD37002 FIXED_STOP_CONTROL can be used to control the response in the case of pulse inhibit at a fixed stop. When the pulses are cancelled, e.g. by using the IS "Pulse enable", the function is not aborted. As a result, the drive will press against the fixed stop again without any further operating action when the machine is restarted.

The rise time of the torque corresponds to the time needed by the current controller of the drive to reach the limitation again.

If the pulses are deleted when a deselection is active (waiting for PLC acknowledgments), the torque limit will be reduced to zero. If the pulses are reactivated during this phase, torque is no longer built up. Once the deselection has been completed, you can continue traversing as normally.

5.3 Response to RESET and function abort


Response to Reset

During selection (fixed stop not yet reached) the function FXS can be aborted with RESET. The termination is carried out such that an "almost achieved" fixed stop (setpoint already beyond the fixed stop, but still within the threshold for the fixed stop detection) will not result in damage.

This is achieved by synchronizing the position setpoint to the new actual position. As soon as the fixed stop is reached, the function remains operative even after RESET.

Function abort

In the case of an **EMERGENCY STOP**, the NC and drive cannot respond, i.e. the PLC must respond.

| |
|--|
|  CAUTION |
|--|

| |
|---|
| <p>Make sure that after canceling the function "travel to fixed stop" by EMERGENCY STOP, no dangerous machine situation (with MD37002 FIXED_STOP_CONTROL e.g. canceling pulse disable) can arise.</p> |
|---|

The fixed stop monitoring function responds in the case of:

- Exit from fixed stop position by axis
- Tool breakage
- Pulse inhibit

5.4 Block-search response

Block search with calculation

No fixed stop may be approached before the target block.

Remedy: Switch off blocks with fixed stops using the script function

Block search without calculation

The commands FXS, FXST and FXSW are ignored.

5.5 Miscellaneous

Inapplicable interface signals

The following IS signals (PLC → NCK) have no effect for axes at endstop until deselected (incl. traversing motion):

- IS "Axis/spindle disable"
- IS "Controller enable"

Actual position at fixed stop

System variable \$AA_IM[x] can determine the actual position of the machine axis, e.g. for test purposes after successful travel to fixed stop.

Combination with measuring function

"Measure with deletion of distancetogo" ("MEAS" command) and "Travel to fixed stop" cannot be programmed at the same time in one block.

Contour monitoring

Axis-specific contour monitoring is not performed while "Travel to fixed stop" is active.

Selection for G64

In MD37060 FIXED_STOP_ACKN_MASK, bit 0 must be 0 (do not wait for PLC input signal "Travel to fixed stop not activated"), since the selection of FXS may not initiate a motion stop. If this bit is programmed, however, the alarm 20090 "Travel to fixed stop not possible - check programming and axis data" is output.

Diagram

The following diagram shows the sequence of the motor current, following error and interface signals for "Activate travel to fixed stop" with a digital drive.

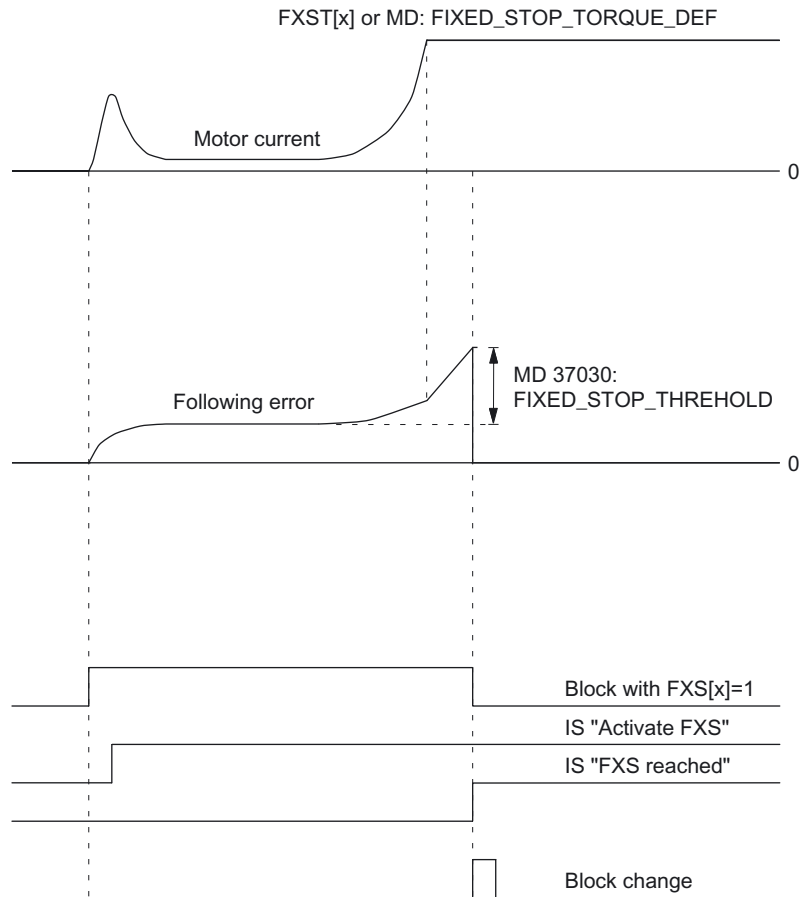


Figure 5-2 Diagram for FXS with a digital drive

5.6 Data lists

5.6.1 Machine data

| Number | Identifier | Name |
|---------------|-----------------------------|--|
| Axis-specific | | |
| 37000 | FIXED_STOP_MODE | Travel to fixed stop mode |
| 37002 | FIXED_STOP_CONTROL | Special function when traveling to fixed stop |
| 37010 | FIXED_STOP_TORQUE_DEF | Default for clamping torque |
| 37012 | FIXED_STOP_TORQUE_RAMP_TIME | Virtual time until reaching the new clamping torque when traveling to fixed stop |
| 37020 | FIXED_STOP_WINDOW_DEF | Default for fixed stop monitoring window |
| 37030 | FIXED_STOP_THRESHOLD | Threshold for fixed stop detection |
| 37040 | FIXED_STOP_BY_SENSOR | Fixed stop detection via sensor |
| 37050 | FIXED_STOP_ALARM_MASK | Enabling the fixed stop alarms |
| 37060 | FIXED_STOP_ACKN_MASK | Monitoring of PLC acknowledgments for travel to fixed stop |

5.6.2 Setting data

| Number | Identifier | Name |
|---------------|-------------------|--|
| Axis-specific | | |
| 43500 | FIXED_STOP_SWITCH | Selection of travel to fixed stop |
| 43510 | FIXED_STOP_WINDOW | Fixed stop monitoring window |
| 43520 | FIXED_STOP_TORQUE | Clamping torque when traveling to fixed stop |

5.6.3 Interface signals

| Number | Bit | Name |
|-----------------------|-----|--------------------------------|
| Axis/spindle-specific | | |
| V380x 0001 | .1 | Acknowledge fixed stop reached |
| V380x 0001 | .2 | Sensor for fixed stop |
| V380x 0001 | .3 | Axis/spindle disable |
| V380x 0002 | .1 | Controller enable |
| V380x 0003 | .1 | Enable travel to fixed stop |
| V390x 0002 | .4 | Enable travel to fixed endstop |
| V390x 0002 | .5 | Fixed stop reached |

Gantry axes (G1)

6.1 Brief description

Note

This function is under license and is not available with version T/M value.

If the corresponding option is activated without a valid license, alarm 8081 "%1 option(s) that has (have) not been licensed using a license key was (were) set" is output. It will not be possible to operate the machine as normal.

For information on operations relating to "Setting (an) option(s)", please refer to the section titled "Licensing in SINUMERIK 802D sl" in the "Turning, Milling, Grinding, Nibbling" Operating Instructions.

Gantry axes

Gantry axes are mechanically grouped machine axes. Because of this mechanical coupling, gantry axes are always traversed in unison. The control occurs through the "gantry axes" function.

The machine axis that is directly traversed is called the leading axis. The machine axis that is traversed in synchronism with it is called the synchronized axis. Together, the leading axis and synchronized axis form a gantry axis grouping.

The difference between the actual positions of the leading axis and synchronized axis is monitored continuously. When the actual position value of the synchronized axis deviates too much from that of the leading axis, the control automatically brings all axes in the gantry grouping to a standstill in order to prevent any damage to the machines.

Application

Two feed drives are required to traverse the gantry on large gantry-type milling machines, i.e. one drive with its own position measuring system on each side. Owing to the mechanical forced coupling, both drives must be operated in absolute synchronism to prevent canting of mechanical components.

Configuration

A gantry axis grouping consisting of a leading axis and synchronized axis can be defined.

6.2 "Gantry axes" function

Application

On large gantry-type milling machines, various axis units (e.g. gantry or crossbeam) are moved by a number of drives, which are mutually independent. Each drive has its own measuring system and thus constitutes a complete axis system. When these mechanically rigidly-coupled axes are traversed, both drives must be operated in **absolute synchronism** in order to prevent canting of mechanical components (resulting in power/torque transmission).

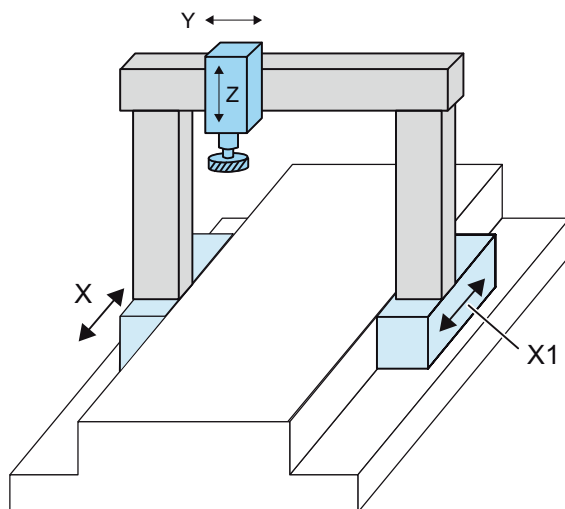


Figure 6-1 Example: Gantry-type milling machine with a gantry axis grouping (X and X1)

The purpose of the "gantry axes" function is to control and monitor machine axes which are rigidly coupled in this way.

Terms

The following terms are frequently used in this functional description:

Gantry axes

Gantry axes comprise one pair of axes, the leading axis and the synchronized axis. As these axes are mechanically coupled, they must always be traversed simultaneously by the NC. The difference between the actual positions of the axes is monitored continuously. The axes in a gantry grouping are either all linear axes or all rotary axes.

Gantry axis grouping

A total of **one** gantry connection can be defined. Each gantry grouping consists of **one** leading axis and **one** synchronized axis.

The gantry axis grouping defines which synchronized axis is controlled by which leading axis, based on machine data settings. The leading axis and synchronized axis cannot be traversed separately.

Leading axis

The leading axis is the gantry axis that exists from the point of view of the operator and programmer and, thus, can be influenced like a standard NC axis. The axis name of the leading axis identifies all axes in the gantry axis grouping.

Synchronized axis

A synchronized axis is the gantry axis whose set position is continuously derived from the motion of the leading axis and is, thus, moved synchronously with the leading axis. From the point of view of the programmer and operator, the synchronized axis "does not exist".

Conditions for a gantry grouping

- A gantry grouping must not contain a spindle.
- A synchronized axis must not be a PLC axis.
- A synchronized axis must not be addressed by a transformation.
- A synchronized axis must not be the slave axis in another type of axis coupling.
- A synchronized axis must not be defined as the leading axis in another axis grouping.

Note

Each axis in the gantry grouping must be set so that it can take over the function of the leading axis at any time, i.e. matching velocity, acceleration and dynamic response settings.

The control performs a plausibility check on the axis definition.

Components

The "gantry axes" function can be subdivided into the following functional units:

1. Setpoint generation of synchronized axis
2. Monitoring of actual value difference
3. Referencing and synchronizing the leading axis and synchronized axis

Setpoint generation of synchronized axis

From the point of view of the operator, all coupled gantry axes are traversed as if only one axis, i.e. the leading axis, were programmed in the NC. Analogously, only the leading axis is programmed in the part program. The commands and traverse requests from the operator, the PLC interface or via the part program therefore apply in equal measure to all axes in the gantry grouping.

When the "gantry axes" function is active, the synchronized axis setpoint is generated directly from the setpoint of the leading axis in all operating modes.

Note

The dynamic control response settings for the leading and synchronized axes must be identical.

Monitoring the actual value difference

The position actual values of the leading and synchronized axes are continuously compared with one another in the interpolation clock cycle and monitored to check that they are still within the permissible tolerance range.

Machine data can be set to specify the following limit values for alarm output and termination of the traversing motion for specific axes:

Gantry warning limit

The "Warning limit exceeded" warning will be relayed to the operator when the actual position value difference exceeds the gantry warning limit:

MD37110 GANTRY_POS_TOL_WARNING (gantry warning limit)

In addition, the following interface signal will be output to the PLC:

V390x 5005.3 (gantry warning limit exceeded)

When below the warning limit, the message and interface signal will automatically be cancelled.

When MD37110 = 0 the message will be disabled.

Gantry trip limit

Alarm 10653 "Error limit exceeded" will be signaled when the machine's maximum permissible actual position value deviation is exceeded:

MD37120 GANTRY_POS_TOL_ERROR (gantry trip limit)

In order to prevent any damage to the machines, the gantry axes will be immediately shut down via the break ramp.

The value of MD37120 is applied when the gantry grouping is synchronized.

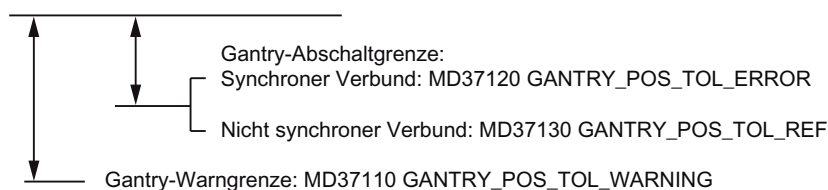
The alarm must be acknowledged with "RESET".

In addition, the following interface signal will be output to the PLC:

V390x 5005.2 (gantry trip limit exceeded)

If the gantry axis grouping has not yet been synchronized, the limit value for the gantry trip limit is derived from the following machine data:

MD37130 GANTRY_POS_TOL_REF (gantry trip limit for referencing)



The "Gantry trip limit exceeded" IS is also output if the gantry grouping is jammed (no servo enable, gantry grouping in "Hold" state).

The monitoring functions are deactivated while the grouping is operating in "Follow-up" mode.

Extended monitoring

An extended monitoring function can be activated with the following machine data:
MD37150 GANTRY_FUNCTION_MASK (gantry functions)

Referencing and synchronization of gantry axes

As the example "Gantry-type milling machine" shows, the forced coupling between gantry axes must remain intact in all operating modes as well as immediately after power ON. In cases where an incremental measuring system is being used for the leading or the synchronized axis, the reference point must be approached while maintaining the axis coupling immediately after machine power ON.

After every axis in the gantry grouping has approached its reference point, any misalignment that may exist between the axes must be eliminated (this is known as the gantry synchronization process). As soon as this has been carried out, the following interface signal is sent to the PLC:

V390x 5005.5 (gantry grouping is synchronized)

The operational sequence for referencing and synchronizing gantry axes is described in detail under the topic "Referencing and synchronizing of gantry axes".

Closed-loop control

The dynamic control response settings for the coupled gantry axes must be identical (see Start-up of gantry axes (Page 94)). This ensures that in normal operation, the leading and synchronized axes move in positional synchronism, even during acceleration and braking.

Load effects are compensated by the appropriate drive of the leading or synchronized axis.

Disturbance characteristic

When a disturbance occurs which causes shutdown of one gantry axis owing, for example, to cancellation of the controller enabling signal (example: EMERGENCY OFF), the coupled gantry axes are also shut down.

Separation of forced coupling

In certain situations (e.g. one gantry axis is no longer referenced owing to an encoder failure), it may be necessary to correct or reduce the misalignment between the gantry axes prior to referencing. To do this, it must be possible to traverse the leading or the synchronized axis **manually in the uncoupled state**.

The forced coupling between the gantry axes can be separated by making the following MD setting and then performing a RESET:

MD37140 GANTRY_BREAK_UP = 1 (break up gantry grouping)

The gantry axes can then be traversed separately by hand; the monitoring of the warning and trip limits is not operative in this state.

| |
|--|
|  CAUTION |
|--|

| |
|---|
| If the gantry axes remain mechanically coupled, there is a risk of damage to the machine when the leading or synchronized axes are traversed in this operating state! |
|---|

6.3 Referencing and synchronizing gantry axes

6.3.1 Introduction

Misalignment after starting

Immediately after the machine is switched on, the leading and synchronized axes may not be ideally positioned in relation to one another (e.g. misalignment of a gantry). Generally speaking, this misalignment is relatively small so that the gantry axes can still be referenced.

In special cases (e.g. gantry axes were brought to a standstill because of a disturbance, power failure, or EMERGENCY OFF), before the axes can be traversed the dimensional offset must be checked for permissible tolerance values and a compensatory motion executed if necessary.

To execute this compensatory motion, the gantry grouping must be separated by means of the following machine data:

MD37140 GANTRY_BREAK_UP (break up gantry grouping)

Gantry synchronization

All gantry axes must first be referenced and then synchronized after the control system is switched on. During gantry synchronization, all gantry axes approach **the reference position of the gantry grouping in the decoupled state**.

The reference position of the gantry grouping for referencing the gantry axes corresponds to the **reference position of the leading axis**:

MD34100 REFP_SET_POS (reference point value/destination point for distance-coded system)

Otherwise, the reference position is the **current actual position of the leading axis**.

These operations for referencing and synchronizing the gantry axes are executed automatically in accordance with a special flowchart.

Referencing process

The flowchart for referencing gantry axes using an incremental measuring system is as follows:

Section 1:

Referencing of the leading axis

Axis-specific referencing of the gantry axes is started by the active machine function REF when the leading axis interface signal is sent from the PLC user program:
V380x 0004.7/6 (traversing key plus/minus)

The leading axis approaches the reference point (operational sequence as for reference point approach).

The appropriate synchronized axis traverses in synchronism with the leading axis. Interface signal "Referenced/synchronized" of the leading axis is output to indicate that the reference point has been reached.

Section 2:

Referencing the synchronized axis

As soon as the leading axis has approached its reference point, the synchronized axis is **automatically** referenced (as for reference point approach).

The dependency between the leading axis and synchronized axis is inverted in the control for this phase so that the leading axis now traverses in synchronism with the synchronized axis. IS "Referenced/synchronized" of the synchronized axis is output to indicate that the reference point has been reached. The gantry axis dependency then reverts to its previous status.

Section 3:

Gantry synchronization

Once all axes in the gantry grouping have been referenced, they must be synchronized with the defined reference position. The actual position of each gantry axis is first compared to the defined reference position of the leading axis.

The next step in the operating sequence depends on the difference calculated between the actual values of the leading and synchronized axes:

- difference is **smaller** than the gantry warning limit:

Gantry synchronization is started **automatically** (provided that IS "Automatic synchronization locking" has not been set). The message "Synchronization in progress gantry grouping x" is output during this process.

All gantry axes traverse at a specific position value **in the decoupled state** at the velocity set in the machine data:

MD34040 REFP_VELO_SEARCH_MARKER (shutdown speed)

The position value is defined by the leading axis:

MD34100 REFP_SET_POS (reference point for incremental system)

The absolute encoders and distanced-coded encoders of the leading axis will be set to the current actual position of the leading axis or to the reference point by the following machine data:

MD34330 REFP_STOP_AT_ABS_MARKER (distancecoded linear measuring system without destination point)

For this operation, the axes traverse at the same velocity as set for reference point approach:

MD34070 REFP_VELO_POS (reference point positioning velocity)

As soon as all gantry axes have reached their target position (ideal position), IS "Gantry grouping is synchronized" is set to "1" followed by re-activation of the gantry axis coupling. The position actual value of all axes in the gantry grouping must now be identical. The gantry synchronization process is now complete.

- Difference is **higher** than the gantry warning limit for the synchronized axis:

IS "Gantry synchronization read to start" is set to "1" and the message "Wait for synchronization start of gantry grouping x" is output. The gantry synchronization process is not started automatically in this case, but must be started explicitly by the operator or from the PLC user program. The process is initiated by IS "Start gantry synchronization" on the leading axis. The signal is set on the leading axis. The operational sequence is then the same as that described above.

The following flowchart illustrates the referencing and synchronization processes.

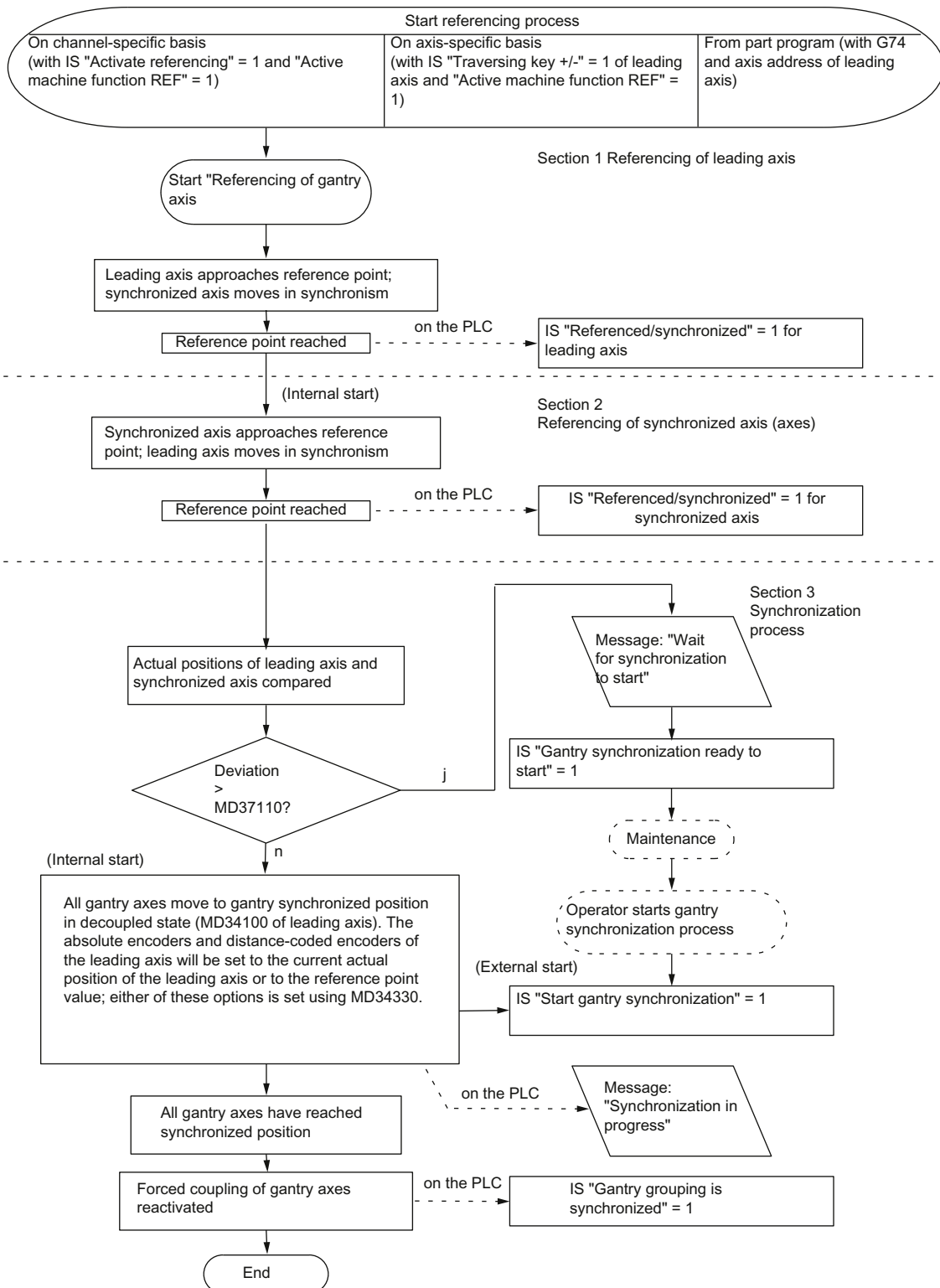


Figure 6-2 Flowchart for referencing and synchronization of gantry axes

Synchronization process

A synchronization process is always required in the following cases:

- after the reference point approach of all axes included in a grouping,
- if the axes become desynchronized (see below).

Operational sequence failure

If the referencing process described above is interrupted as a result of disturbances or a RESET, proceed as follows:

- **Abort within section 1 or 2:**

Restart reference point with leading axis (see section 1)

- **Abort in section 3:**

In cases where the gantry axes have not yet been referenced (IS "Referenced/Synchronized" = 1), the gantry synchronization process can be started again with IS "Synchronize gantry grouping".

Restart gantry synchronization

Synchronization of the gantry axes can be started with IS "Start gantry synchronization" under the following conditions only:

- JOG/REF mode must be active. The following interface signal must be set:
V3100 0001.2 (active machine function REF)
- V390x 5005.5 = 0 (gantry grouping is synchronized)
- All grouping axes operate within the tolerance windows:
V390x 5005.4 = 1 ("Gantry synchronization ready to start")
- Axis is not referenced in the NC channel
V3300 0001.0 = 0 (referencing active)

If gantry synchronization is **not started from the referencing process** by means of IS "Start gantry synchronization", then **the current actual position of the leading axis** - rather than the referencing position from MD34100 - will be specified as the target position and approached in the decoupled state.

Note

Automatic synchronization can be locked by the leading axis by means of the following interface signal:

V380x 5005.5 (automatic synchronization locking)

This always makes sense if no axis enabling signal has yet been issued for the axes. In this case, the synchronization process should also be started explicitly with the interface signal:
V380x 5005.4 = 1 (start gantry synchronization)

Loss of synchronization

The gantry grouping becomes desynchronized ($V390x\ 5005 = 0$) if:

- The gantry axes were in "Follow-up" mode
- The reference position of a gantry axis is lost, e.g. during "Parking" (no measuring system active)
- A gantry axis is re-referenced ($V390x\ 0000 = 0$)
- The gantry grouping was separated ($MD37140 = 1$)

In cases where the gantry grouping has lost synchronization during operation as the result of a disturbance, then the gantry synchronization process can be restarted directly by means of IS "Start gantry synchronization" (condition: $V390x\ 0000 = 1$ for all axes in the gantry grouping). In this case, the synchronizing axes traverse the current actual position of the leading axis in the decoupled state.

If an EMERGENCY OFF occurs while a gantry grouping is moving, and is then rescinded, and both axes have drifted apart less than the standstill tolerance of the following axis, then the gantry grouping will automatically synchronize. It no longer needs to go in the BA REF.

Reference point selection

To ensure that the shortest possible paths are traversed when the gantry axes are referenced, the reference point values for the leading and synchronized axes should be the same ($MD34100$).

Allowance for deviations in distance between the zero mark and the reference point must be made for specific axes via the machine data:

- $MD34080$ REFP_MOVE_DIST (reference point distance)
- $MD34090$ REFP_MOVE_DIST_CORR (reference point offset/absolute offset)

Referencing direction selection

The zero mark leveling function of the following axis can be defined using the following machine data:

$MD37150$ GANTRY_FUNCTION_MASK bit 1

| Bit | Value | Meaning |
|-----|-------|--|
| 1 | 0 | The zero mark leveling function of the following axis is similar to $MD34010$ REFP_CAM_DIR_IS_MINUS. |
| | 1 | The zero mark leveling function of the master axis is the same as the slave axis |

During referencing, the reference point value of the leading axis is specified as the target position for all axes in the grouping for the synchronization compensatory motion. This position is then approached without axis coupling. The absolute encoders and distance-coded encoders of the leading axis will be set to the current actual position of the leading axis or to the reference point value; either of these options is set using the following machine data:

$MD34330$ REFP_STOP_AT_ABS_MARKER
(distance-coded linear measuring system without destination point)

If only one reference cam is used for the leading and synchronized axes, then this must be taken into account in the PLC user program.

6.3.2 Automatic synchronization

Automatic synchronization can take place:

- In referencing mode (see the section titled "Introduction")
- In the manner described below

If a gantry grouping is switched to follow-up mode, monitoring of the actual values between the leading and synchronized axes is deactivated. The grouping is no longer synchronized as a result. The leading axis sets IS "Gantry grouping is synchronous" to "0" regardless of the positions of the axes.

If the gantry grouping is switched from follow-up mode to position control mode, axis synchronism is automatically restored, provided the actual-value monitoring function does not detect a difference between the positions of the leading and synchronized axes that is greater than the setting in MD36030.

In this case, a new setpoint is specified for the synchronized axis without interpolation. The positional difference detected earlier is then corrected by the position controller. The correction causes only the synchronized axis to move.

The motion sequence of the synchronized axis is similar to the situation in which the grouping switches from the "Hold" state to position control mode. In this case, the position specified by the position controller before the grouping is halted is set again on condition that the zero speed monitor has not activated alarm 25040 (with follow-up as alarm reaction) in the meantime.

The same tolerance window is used for this type of automatic synchronization as for the standstill monitoring function:

MD36030 STANDSTILL_POS_TOL (standstill tolerance)

Parameter set-dependent loading takes place with the following machine data:

MD36012 STOP_LIMIT_FACTOR (exact stop coarse/fine factor and standstill)

Note

The following interface signal blocks automatic synchronization in all modes except referencing mode:

V380x 5005.5 (automatic synchronization locking)

If automatic synchronization is to be activated, V380x 5005.5 must be set to "0".

Following this, one of the axes in the gantry grouping must be switched from follow-up mode to position-controlled mode. This is achieved with the interface signals:

V380x 0001.4 = 1 (follow-up mode)

V380x 0002.1 = 1 (servo enable)

6.3.3 Points to note

Channelspecific referencing

Gantry axes can also be referenced on a channel-specific basis:
V3200 0001.0 (activate referencing)

The value of the leading axis machine data is used for the axis sequence during channel-specific referencing:

MD34110 REFP_CYCLE_NR (axis sequence for channel-specific referencing)

After the reference point of the leading axis has been reached, the synchronized axis is referenced first as described above.

Referencing from part program with G74

The referencing and synchronization process for gantry axes can also be initiated from the part program by means of command G74. In this case, only the axis name of the leading axis may be programmed. The operational sequence is analogous to that described for axis-specific referencing.

Position measuring system with distancecoded reference marks

In order to ensure that large distances do not need to be traversed while approaching the reference point, it is possible to use a position measuring system with distance-coded reference markers as the only measuring system for gantry axes. In this way the measuring system is referenced after traversal of a short path (e.g. 20 mm). The procedure for referencing the gantry axes is the same as that described for the normal incremental measuring system.

Absolute encoder

During the synchronization compensatory motion, all the axes in the gantry axis grouping (in the decoupled state) also traverse to the reference point value of the leading axis, which is defined in the following machine data:

MD34100 REFP_SET_POS (reference point value/destination point for distance-coded system)

The absolute encoders and distance-coded encoders of the leading axis will be set to the current actual position of the leading axis or to the reference point value; either of these options is set using the following machine data:

MD34330 REFP_STOP_AT_ABS_MARKER
(distance-coded linear measuring system without destination point)

Activation of axis compensations

Compensation functions can be activated for both the leading axis and the synchronized axis. Compensation values are applied separately for each individual gantry axis. These values must therefore be defined and entered for the leading axis and the synchronized axes during start-up.

The compensations do not become operative internally in the control until the axis is referenced or the gantry grouping synchronized. The following applies:

| Compensation type | Takes effect when | PLC interface signal |
|-----------------------|---------------------------------|-----------------------------------|
| Backlash compensation | Axis is referenced | "Referenced/Synchronized" |
| LEC | Axis is referenced | "Referenced/synchronized" |
| Sag compensation | Gantry grouping is synchronized | "Gantry grouping is synchronized" |

If active compensation causes the synchronized axis to move, a traverse command is displayed for the synchronized axis, independently of the leading axis.

Monitoring functions effective

Analogous to normal NC axes, the following monitoring functions do not take effect for gantry axes until the reference point is reached (IS "Referenced/Synchronized"):

- Working area limits
- Software limit switch
- Protection zones

The axial machine data values are used as monitoring limit values for the synchronized axis as well.

6.4 Start-up of gantry axes

General information

Owing to the forced coupling which is normally present between leading and synchronized gantry axes, the gantry axis grouping must be commissioned as if it were an axis unit. For this reason, the axial machine data for the leading and synchronized axes must always be defined and entered jointly.

If the synchronized axis is being overloaded by the leading axis due to reduced dynamics, this is acknowledged with alarm 10656.

Special points to be noted with regard to starting up gantry axes are described below.

Axis traversing direction

As part of the start-up procedure, a check must be made to ensure that the direction of rotation of the motor corresponds to the desired traversing direction of the axis. Correct by means of axial machine data:

MD32100 AX_MOTION_DIR (traversing direction)

Activation of the axis grouping

MD37100 GANTRY_AXIS_TYPE[a,b] (gantry axis definition)

This machine data is determined for the following gantry axis:

- Which gantry grouping (1) the axis is to be assigned to
- Whether it is to act as a leading or synchronized axis

Table 6- 1 Possible values for MD37100

| a | b | Gantry axis | Gantry grouping |
|---|---|-------------------|-----------------|
| - | 0 | None | - |
| 0 | 1 | Leading axis | 1 |
| 1 | 1 | Synchronized axis | 1 |

For commissioning purposes, the axes in a gantry grouping must be defined as either all linear axes or all rotary axes:

MD30300 IS_ROT_AX (rotary axis/spindle)

Entering gantry trip limits

For the monitoring of the actual position values of the synchronized axis in relation to the actual position of the leading axis, the limit values for termination, as well as for the leading and synchronized axes, should be entered corresponding to the specifications of the machine manufacturer:

MD37120 GANTRY_POS_TOL_ERROR (gantry trip limit)

MD37130 GANTRY_POS_TOL_REF (gantry trip limit for referencing)

Note

The control must then be switched off and then on again because the gantry axis definition and the trip limit values only take effect after power ON.

Response to setpoint changes and disturbances

The gantry axes can only operate in exact synchronism if the parameters for the control circuits of the leading and synchronized axes are set to the **same dynamic response value**.

The axial control loops (position, speed and current controllers) should each be set to the **optimum** value so that disturbances can be eliminated as quickly and efficiently as possible. The **dynamic response adaptation** function in the setpoint branch is provided to allow differing dynamic responses of axes to be matched without loss of control quality.

The following control parameters must be set to the optimum axial value for both the leading axis and the synchronized axis:

- MD32200 POSCTRL_GAIN (servo gain factor)
- MD32620 FFW_MODE (precontrol parameter)
- MD32610 VELO_FFW_WEIGHT (precontrol factor for acceleration/speed precontrol)
- MD32810 EQUIV_SPEEDCTRL_TIME (equivalent time constant speed control loop for precontrol)

The following control parameters must be set to the same value for the leading axis and synchronized axis:

- MD32400 AX_JERK_ENABLE (axial jerk limitation)
- MD32410 AX_JERK_TIME (time constant for the axial jerk filter)
- MD32420 JOG_AND_POS_JERK_ENABLE (basic setting for axial jerk limitation)
- MD32430 \$MA_JOG_AND_POS_MAX_JERK (axial jerk)

Dynamics matching

The leading axis and the coupled synchronized axis must be capable of the same dynamic response to setpoint changes. The same dynamic response means: The following errors are equal in magnitude when the axes are operating at the same speed.

The dynamic response adaptation function in the setpoint branch makes it possible to obtain an excellent match in the response to setpoint changes between axes, which have different dynamic characteristics (control loops). The difference in equivalent time constants between the dynamically "weakest" axis and the other axis in each case must be specified as the dynamic response adaptation time constant.

Example

When the speed feedforward control is active, the dynamic response is primarily determined by the equivalent time constant of the "slowest" speed control loop.

Leading axis:

MD32810 EQUIV_SPEEDCTRL_TIME [n] = 5 ms

Synchronized axis:

MD32810 EQUIV_SPEEDCTRL_TIME [n] = 3 ms

Time constant of dynamic response adaptation for synchronized axis:

MD32910 DYN_MATCH_TIME [n] = 5 ms - 3 ms = 2 ms

(time constant of dynamic response adaptation)

Dynamic response adaptation must be activated axially with the following machine data:

MD32900 DYN_MATCH_ENABLE (dynamic response adaptation)

Note

Checking dynamic response adaptation:

For the purpose of fine tuning, it may be necessary to adjust servo gain factors or feedforward control parameters slightly to achieve an optimum result.

Referencing gantry axes

The positions of the reference points for the leading and synchronized axes must first be set to almost identical values.

To ensure that the synchronization compensatory motion of the gantry axes is not started automatically, during first commissioning the gantry warning limit must be set to 0 before referencing:

MD37100 GANTRY_POS_TOL_WARNING (gantry axis definition)

This will prevent a warning message being output during traversing motion.

In cases where an excessively high additional torque is acting on the drives due to misalignment between the leading and synchronized axes, the gantry grouping must be aligned before the axes are traversed. After this, the gantry axes must be referenced as outlined in the section titled "Referencing and synchronizing gantry axes":

Once the leading and synchronized axes have been referenced, the difference between them must be determined (by comparing the actual position value display on the "Service Axes" screen in the "System" → "Service Display" operating area). This difference must be applied as the reference point offset:

MD34080 REFP_MOVE_DIST (reference point distance)

MD34090 REFP_MOVE_DIST_CORR (reference point offset/absolute offset)

Synchronizing gantry axes

The gantry synchronization process must be activated with IS "Start gantry synchronization" (see Section "Referencing and synchronizing of gantry axes"). Once the axes have been synchronized (IS "Gantry grouping is synchronized" = 1), the dimensional offset between the leading and synchronized axes must be checked to ensure that it equals 0. Corrections may need to be made in the machine data mentioned above.

Input of gantry warning limit

Once the reference point values for the leading and synchronized axes have been optimized so that the gantry axes are perfectly aligned with one another after synchronization, the warning limit values for all axes must be entered in the following machine data:

MD37110 GANTRY_POS_TOL_WARNING (gantry warning limit)

To do this, the value must be increased incrementally until the value is just below the alarm (limit exceeded) response limit. It is particularly important to check the acceleration phases.

This limit value also determines the position deviation value at which gantry synchronization is automatically started in the control.

Calculating and activating compensations

In cases where the gantry axes require compensation (sag or leadscrew error), the compensation values for the leading axis **and** the synchronized axis must be calculated and entered in the appropriate parameters or tables.

Refer to the section titled "Compensations (K3)".

Function generator/measuring function

The activation of the function generator and measuring function using the startup tool will be aborted on the synchronized axis with an error message.

When an activation of the synchronized axis is absolutely necessary (e.g. to calibrate the machine), the leading and synchronized axes must be temporarily interchanged.

Note

Generally, the start of the function generator, measuring functions and AM setup triggers the virtual axes to abort upon error recognition.

Special cases

If **individual** axes have to be activated, the gantry groups must be temporarily canceled. As the second axis no longer travels in synchronism with the first axis, the activated axis must not be allowed to traverse beyond the positional tolerance.

If the gantry grouping is canceled, the following points must be noted:

- Always activate the traversing range limits and set them to the lowest possible values (position tolerance)
- Synchronize the gantry grouping first if possible and then execute a POWER-ON-RESET **without** referencing the axes again. This ensures that the traversing range limits always refer to the same position (i.e. that which was valid on power ON).
- Avoid using the step-change function. Position step changes are only permissible if they stay within the permitted tolerance.
- Always use an offset of 0 for the function generator and measuring function in contrast to the recommendations for normal axes.
- Set the amplitudes for function generator and measuring function to such low values that the activated axis traverses a shorter distance than the position tolerance allows. Always activate the traversing range limits as a check (see above).

Note

As a supplement to the more general description given here of features of start-up and dynamic control response of drives, a complete example of a concrete constellation defined on the basis of its machine data can be found in Chapter "Example".

Start-up support for gantry groupings

The commissioning functions of the function generator and measuring functions are assigned parameters via PI services. All parameterized axes commence traversing when the NC Start key on the MCP panel is pressed in JOG mode.

A window is displayed in the "Measuring function and function generator in gantry grouping" operator interface. Two amplitude values, each with an offset and bandwidth, must be entered in this window. The first amplitude value applies to the measuring axis and the second to the other coupled axes.

6.5 PLC interface signals for gantry axes

Special IS for gantry axes

The special PLC interface signals of the coupled gantry axes are taken via the axial PLC interface of the leading or synchronized axes. Table below shows all special gantry-PLC interface signals along with their codes and indicates whether the IS is evaluated on the leading axis or the synchronized axis.

Table 6- 2 Assignment of gantry-PLC interface signals to leading and synchronized axes

| PLC interface signal | PLC ↔ NCK | Address | Leading axis | Synchronized axis |
|---------------------------------------|-----------|--------------|--------------|-------------------|
| Start gantry synchronization | → | V380x 5005.4 | x | |
| Automatic synchronization locking | → | V380x 5005.5 | x | |
| Gantry axis | ← | V390x 5005.7 | 1 | 1 |
| Gantry leading axis | ← | V390x 5005.6 | 1 | 0 |
| Gantry grouping is synchronized | ← | V390x 5005.5 | x | |
| Gantry synchronization ready to start | ← | V390x 5005.4 | x | |
| Gantry warning limit exceeded | ← | V390x 5005.3 | | x |
| Gantry trip limit exceeded | ← | V390x 5005.2 | | x |

x ≙ relevant for ...

Effect of axial interface signals on gantry axes

a) Axial interface signals from PLC to axis (PLC → NCK)

The axial interface signals from the PLC to the axis are always referred to all gantry axes in the grouping. In this case, all gantry axes (leading and synchronized axis) have equal priority.

For example, all axes in the gantry groupings will be simultaneously shut down when the following interface signal is set to "0" from the leading axis:

V380x 0002.1 (servo enable)

The following table shows the effect of individual interface signals (from PLC to axis) on gantry axes:

Table 6- 3 Effect of interface signals from PLC to axis on leading and synchronized axes

| PLC interface signal | Address | Effect on | |
|-------------------------------|--------------|--------------------------------|-------------------|
| | | Leading axis | Synchronized axis |
| Axis/spindle disable | V380x 0001.3 | On all axes in gantry grouping | No effect |
| Position measuring system 1 | V380x 0001.4 | Axial | Axial |
| Controller enable | V380x 0002.1 | On all axes in gantry grouping | |
| Delete distance to go (axial) | V380x 0002.2 | Axial | No effect |
| Clamping in progress | V380x 0002.3 | Axial | Axial |
| Feed stop | V380x 0004.4 | On all axes in gantry grouping | |

| PLC interface signal | Address | Effect on | |
|-------------------------------------|-------------------|---|-------------------|
| | | Leading axis | Synchronized axis |
| Hardware limit switch minus/plus | V380x 1000.0/.1 | Axial alarm: Brake request on all axes in gantry grouping | |
| 2. Hardware limit switch minus/plus | V380x 1000.2/.3 | Axial | Axial |
| Select drive parameter set | V380x 4001.0 - .2 | Axial | Axial |
| Enable Pulses | V380x 4001.7 | Axial | Axial |

Either the "Follow-up" state (IS of one gantry axis = 1) or the "Stop" state (IS of all gantry axes = 0) is activated for all gantry axes, depending on interface signal:

V380x 0001.4 (follow-up mode)

b) Axial interface signals from axis to PLC (NCK → PLC)

Each of the axial, axis-to-PLC interface signals for the synchronized axis and the leading axis is always set on an axis-specific basis and output to the PLC.

Exception:

When the leading axis is being traversed, the interface signals are also set for the synchronized axis:

V390x 0004.6/.7 (traverse command minus/plus)

6.6 Miscellaneous points regarding gantry axes

Manual travel

It is not possible to traverse a synchronized axis directly by hand in JOG mode. Traverse commands entered via the traversing keys of the synchronized axis are ignored internally in the control. Rotation of the handwheel for the synchronized axis has no effect either.

Handwheel override

An overriding motion by means of the handwheel can only be applied to the leading axis in coupled axis mode. In this case, the synchronized axes traverse in synchronism with the leading axis.

Programming in part program

Only the leading axis of a gantry axis grouping may be programmed in the part program. An alarm is generated while programming a synchronized axis, even when a gantry axis grouping is separated.

PRESET

The PRESET function can only be applied to the leading axis. All axes in the gantry grouping are reevaluated internally in the control when PRESET is activated. The gantry axis then lose their reference and synchronization:

V390x 5005.5 = 0 (gantry grouping is synchronized)

Default for RESET

In an active gantry grouping, the following MD parameterization is ignored for the synchronized axes:

MD30450 IS_CONCURRENT_POS_AX=1
(reset default: neutral axis/channel axis)

The state of the leading axis is assumed. The user is informed about the inappropriate configuration with display alarm 4300.

Position display

The position actual value display shows the actual values of both the leading axis and the synchronized axes. The same applies to the service display values in the "System" operating area.

Software limit switch

The SW limit switch monitor is processed for the leading axis only. If the leading axis crosses the limit switch, all axes in the gantry grouping are braked to a standstill.

6.7 Example

6.7.1 Creating a gantry grouping

Introduction

The individual steps involved in the process are explained below using an example constellation:

- Setting up a gantry grouping
- Referencing its axes
- Aligning any offsets
- Synchronizing the axes involved

Constellation

Machine axis 1 = gantry leading axis, incremental measuring system

Machine axis 3 = gantry synchronized axis, incremental measuring system

The following MD describes the output values. Individual settings must be corrected or added later according to the information below.

Gantry machine data

Axis 1:

MD37100 GANTRY_AXIS_TYPE = 1

MD37110 GANTRY_POS_TOL_WARNING = 0

MD37120 GANTRY_POS_TOL_ERROR = 1 mm

MD37130 GANTRY_POS_TOL_REF = 100 mm (max. misalignment)

MD37140 GANTRY_BREAK_UP = 0

Axis 3:

MD37100 GANTRY_AXIS_TYPE = 11

MD37110 GANTRY_POS_TOL_WARNING = 0

MD37120 GANTRY_POS_TOL_ERROR = 1 mm

MD37130 GANTRY_POS_TOL_REF = 100 mm (max. misalignment)

MD37140 GANTRY_BREAK_UP = 0

Reference point machine data

The MD values specified apply for the first encoder in both axis 1 and axis 3.

MD34000 REFP_CAM_IS_ACTIVE = TRUE

MD34010 REFP_CAM_DIR_IS_MINUS = e.g. FALSE

MD34020 REFP_VELO_SEARCH_CAM =

MD34030 REFP_MAX_CAM_DIST = corresponds to max. distance traversed

MD34040 REFP_VELO_SEARCH_MARKER =

MD34050 REFP_SEARCH_MARKER_REVERSE = e.g. FALSE

MD34060 REFP_MAX_MARKER_DIST = difference btw. cam edge and 0 mark

MD34070 REFP_VELO_POS =

MD34080 REFP_MOVE_DIST = 0

MD34090 REFP_MOVE_DIST_CORR = 0

MD34092 REFP_CAM_SHIFT = 0

MD34100 REFP_SET_POS = 0

MD34200 ENC_REFP_MODE = 1

6.7.2 Setting of NCK PLC interface

Introduction

An automatic synchronization process during axis referencing must first be disabled in order to prevent any damage to grouping axes that are misaligned.

Disabling of automatic synchronization

The PLC user program sets the following IS:

- For the leading axis (axis 1):

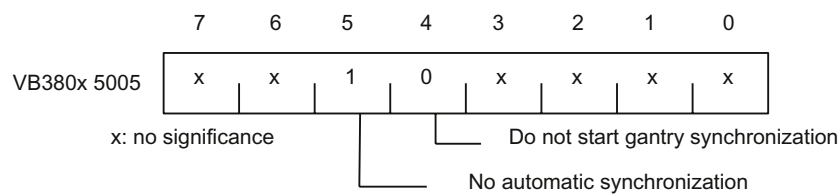


Figure 6-3 NCK-PLC interface VB380x 5005 relative to leading axis

- For the synchronized axis (axis 3):

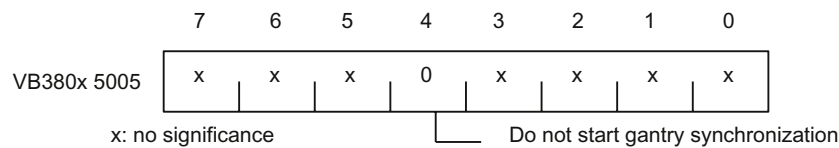


Figure 6-4 NCK-PLC interface VB380x 5005 relative to synchronized axis

The NCK sets the following IS as a confirmation:

- For the leading axis (axis 1):

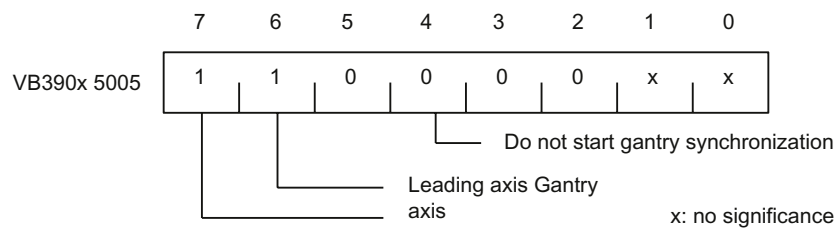


Figure 6-5 NCK-PLC interface VB390x 5005 relative to leading axis

- For the synchronized axis (axis 3):

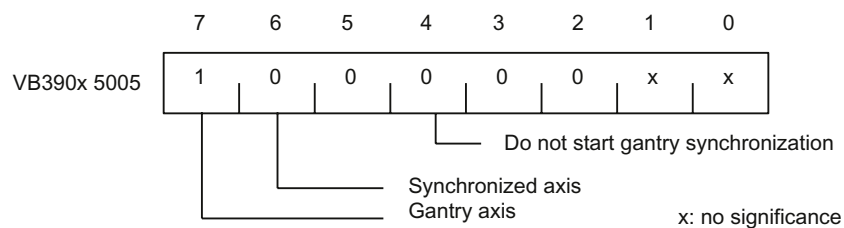


Figure 6-6 NCK-PLC interface VB390x 5005 relative to synchronized axis

6.7.3 Commencing start-up

Referencing

The following steps must be taken:

- Select "REF" operating mode
- Start referencing for the leading axis (axis 1)
- Wait until message "10654 Channel 1 Waiting for synchronization start" appears.

At this point, the NCK has prepared the leading axis for synchronization and signals this using the following IS:

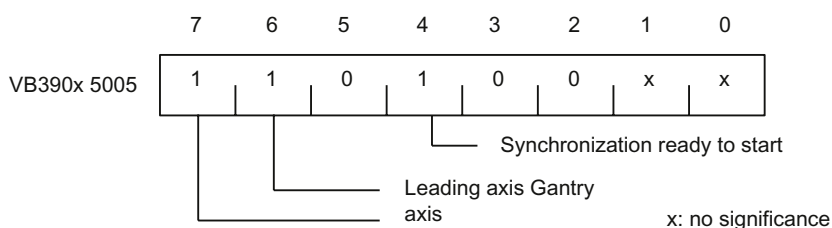


Figure 6-7 NCK-PLC interface VB390x5005: Leading axis ready for synchronization

In addition, the following steps must be taken:

- RESET
- Read off values in machine coordinate system:
 e.g.
 X = 0.941
 Y = 0.000
 XF = 0.000
- Enter the X value of the leading axis (axis 1) with inverted sign in the machine data of the synchronized axis (axis 3):
 MD34090 REFP_MOVE_DIST_CORR = - 0.941

Note

This MD is effective after POWER ON. To avoid having to perform a POWER ON in advance, this value can also be entered in the following machine data:

MD34080 REFP_MOVE_DIST (reference point distance)

The MD is then valid after a RESET.

- Start referencing again for axis 1 with the modified machine data
- Wait until message "10654 Channel 1 Waiting for synchronization start" appears

- At this point, the NCK has prepared axis 1 for synchronization and sends the same interface signal as shown in the image above:
- Examine actual positions of machine. Case A or B might apply:

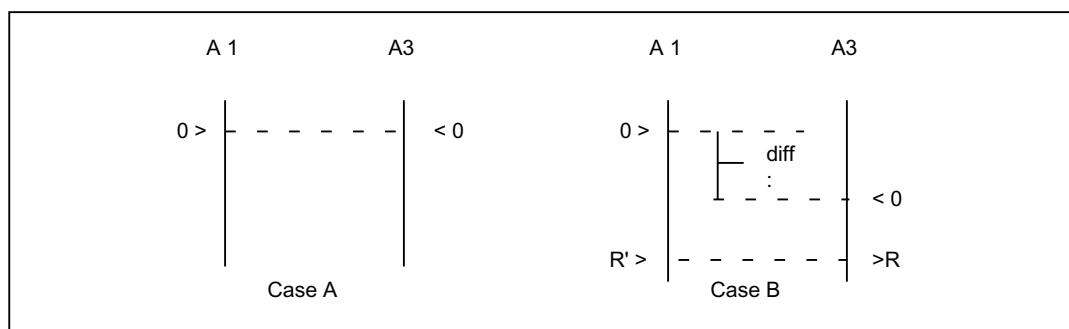


Figure 6-8 Possible results of referencing the leading axis

If Case A applies, the synchronization process can be started immediately. See step "Start synchronization". If Case B applies, the offset "diff" must be calculated and taken into account:

- Measuring of diff
- By using two appropriate, right-angled reference points R and R" in the machine bed (at the right of the image), the difference in position can be traversed in JOG. The diff offset can then be read as the difference in the position display. The diff offset must be entered in the machine data of axis 3 (synchronized axis):

MD34100 REFP_SET_POS

Continue with Step 1 (see above).

- Start gantry synchronization. PLC sets:
V380x 5005.4 = 1 (start gantry synchronization)

6.7.4 Setting warning and trip limits

As soon as the gantry grouping is set and synchronized, the following machine data must still be set to correspond:

MD37110 GANTRY_POS_TOL_WARNING (gantry warning limit)

MD37120 GANTRY_POS_TOL_ERROR (gantry trip limit)

Proceed as follows

- Set the machine data for all axes with a large value to begin with:
MD37120 GANTRY_POS_TOL_ERROR (gantry trip limit)
- Set a very small value in the machine data:
MD37110 GANTRY_POS_TOL_WARNING (gantry warning limit)
When you put a heavy, dynamic strain on the axes, always be careful to re-enter the self-canceling alarm "10652 channel %1 axis %2 gantry warning limit exceeded".
- Now increase MD37110.
Do this until the alarm no longer appears. The interface indicates the status specified below. (That must occur in the appropriate window, according to production.)
- Enter the value calculated for the warning limit + a small buffer value for safety purposes in machine data MD37120:

Error limit values

Values are entered in the following machine data:

MD37110 GANTRY_POS_TOL_WARNING (gantry warning limit)

MD37120 GANTRY_POS_TOL_ERROR (gantry trip limit)

MD37130 GANTRY_POS_TOL_REF (gantry trip limit for referencing)

These should have the following scales of magnitude at the end of the customizing process:

MD37110 < MD37120 < MD37130

Note

The same procedure must be followed when commissioning a gantry grouping in which the axes are operated by **linear motors** and associated measuring systems.

The error limits entered into machine data MD37110 and MD37120 are considered to be additional tolerance values for the actual-value difference of the leading axis and following axis if the IS "Gantry is synchronous" is not present (e.g. to be resynchronized after canceling alarms without the gantry grouping).

6.8 Data lists

6.8.1 Machine data

| Number | Identifier | Name |
|---------------|-------------------------------|--|
| Axis-specific | | |
| 30300 | IS_ROT_AX | Rotary axis |
| 32200 | POSCTRL_GAIN[0]...[5] | Kv factor |
| 32400 | AX_JERK_ENABLE | Axial jerk limitation |
| 32410 | AX_JERK_TIME | Time constant for axis jerk filter |
| 32420 | JOG_AND_POS_JERK_ENABLE | Initial setting for axial jerk limitation |
| 32430 | JOG_AND_POS_MAX_JERK | Axial jerk |
| 32610 | VELO_FFW_WEIGHT[0]...[5] | Feedforward control factor for speed feedforward control |
| 32620 | FFW_MODE | Feedforward control mode |
| 32810 | EQUIV_SPEEDCTRL_TIME[0]...[5] | Equivalent time constant speed control loop for precontrol |
| 32900 | DYN_MATCH_ENABLE | Dynamic response adaptation |
| 32910 | DYN_MATCH_TIME[0]...[5] | Time constant for dynamic response adaptation |
| 34040 | REFP_VELO_SEARCH_MARKER[0] | Creep velocity |
| 34070 | REFP_VELO_POS | Reference point start velocity |
| 34080 | REFP_MOVE_DIST[0] | Reference point approach distance |
| 34090 | REFP_MOVE_DIST_CORR[0] | Home position offset |
| 34100 | REFP_SET_POS[0]...[3] | Reference point value |
| 34110 | REFP_CYCLE_NR | Axis sequence for channel-specific referencing |
| 34330 | REFP_STOP_AT_ABS_MARKER[0] | Distance-coded linear measuring system without destination point |
| 36012 | STOP_LIMIT_FACTOR[0]...[5] | Exact stop coarse/fine factor and zero speed |
| 36030 | STANDSTILL_POS_TOL | Zero speed tolerance |
| 37100 | GANTRY_AXIS_TYPE | Gantry axis definition |
| 37110 | GANTRY_POS_TOL_WARNING | Gantry warning limit |
| 37120 | GANTRY_POS_TOL_ERROR | Gantry trip limit |
| 37130 | GANTRY_POS_TOL_REF | Gantry trip limit for referencing |
| 37140 | GANTRY_BREAK_UP | Invalidate gantry axis grouping |

6.8.2 Interface signals

| Number | Bit | Name | Leading axis | Synchronized axis |
|------------------|-----|---------------------------------------|--------------|-------------------|
| Mode-specific | | | | |
| V3100 0001 | .2 | Active machine function REF | - | - |
| Channel-specific | | | | |
| V3300 0001 | .0 | Referencing active | - | - |
| Axis-specific | | | | |
| V380x 5005 | .4 | Start gantry synchronization | x | - |
| V380x 5005 | .5 | Automatic synchronization locking | x | - |
| V390x 0000 | .4 | Referenced/synchronized 1 | - | - |
| V390x 5005 | .2 | Gantry trip limit exceeded | - | x |
| V390x 5005 | .3 | Gantry warning limit exceeded | - | x |
| V390x 5005 | .4 | Gantry synchronization ready to start | x | - |
| V390x 5005 | .5 | Gantry grouping is synchronized | x | - |
| V390x 5005 | .6 | Gantry leading axis | 1 | 0 |
| V390x 5005 | .7 | Gantry axis | 1 | 1 |

x ≙ relevant for ...

Velocities, Setpoint/Actual Value Systems, Closed-Loop Control (G2)

7

7.1 Velocities, traversing ranges, accuracies

7.1.1 Velocities

The maximum path/axis velocities and spindle speed are influenced by the machine design, the dynamic response of the drive and the limit frequency of the actual-value acquisition system (encoder limit frequency).

The maximum axis velocity is defined in MD32000 MAX_AX_VELO (maximum axis velocity).

The maximum permissible spindle speed is defined in MD35100 SPIND_VELO_LIMIT (maximum spindle speed).

As well as limiting using MD32000 MAX_AX_VELO, the control limits the maximum path velocity in relation to the situation and according to the following formula:

$$V_{\max} \leq \frac{\text{progr. path length in part program block [mm or degrees]}}{\text{Interpolation cycle [s]}} * 0.9$$

With a higher feedrate (resulting from program and affected by feed override), the maximum path velocity is limited to V_{\max} .

This automatic feedrate limiting can lead to a drop in velocity over several blocks with programs generated by CAD systems with extremely short blocks.

Example:

Interpolation cycle = 12 ms

N10 G0 X0 Y0; [mm]

N20 G0 X100 Y100; [mm]

⇒ Path length programmed in block = 141.42 mm

⇒ $V_{\max} = (141.42 \text{ mm}/12 \text{ ms}) * 0.9 = 10,606.6 \text{ mm/s} = 636.39 \text{ m/min}$

The following restriction applies to the minimum path or axis velocity:

$$V_{\min} \geq \frac{10^{-9}}{\text{comput. resolution} \left[\frac{\text{incr.}}{\text{mm or degrees}} \right] * \text{IPO cycle [s]}}$$

The computational resolution is defined via MD10200 INT_INCR_PER_MM (computational resolution for linear positions) or MD10210 INT_INCR_PER_DEG (computational resolution for angular positions). It is described in more detail on the pages that follow.

If V_{\min} is not reached, traversing is not carried out.

7.1 Velocities, traversing ranges, accuracies

Example: MD10200 INT_INCR_PER_MM = 1,000 [incr./mm]
 Interpolation cycle = 12 ms
 $\Rightarrow V_{\min} = 10^{-3} / (1,000 \text{ incr./mm} \times 12 \text{ ms}) = 0.005 \text{ mm/min}$

The value range of the feedrates depends on the computational resolution selected. If MD10200 INT_INCR_PER_MM (computational resolution for linear positions) (1,000 incr./mm) or MD10210 INT_INCR_PER_DEG (computational resolution for angular positions) (1,000 incr./degree) are assigned their default values, the following range of values can be programmed with the specified resolution:

Value range for path feed F: Metric system:
 0.001 v F v 999,999.999 [mm/min, mm/rev, degree/min, degree/rev].

Value range for spindle speed S:
 0.001 v S v 999,999.999 [rev/min]

If the computational resolution is increased/decreased by a factor of 10, the value ranges change accordingly (maximum number of decimal points is constant).

7.1.2 Traversing ranges

The range of values of the traversing range depends on the computational resolution selected.

If MD10200 INT_INCR_PER_MM (computational resolution for linear positions) (1000 incr./mm) or MD10210 INT_INCR_PER_DEG (computational resolution for angular positions) (1000 incr./degree) are assigned their default values, the following range of values can be programmed with the specified resolution:

Table 7- 1 Traversing ranges of axes

| | G71 [mm, degrees] | G70 [inch, degrees] |
|----------------------------------|-------------------|---------------------|
| Linear axes X, Y, Z, etc. | ± 999,999.999 | ± 399,999.999 |
| Rotary axes A, B, C, etc. | ± 999,999.999 | ± 999,999.999 |
| Interpolation parameters I, J, K | ± 999,999.999 | ± 399,999.999 |

The unit of measurement of rotary axes is always degrees.

If the calculation resolution is increased/decreased by a factor of 10, the value ranges change accordingly (maximum number of decimal points is constant).

The traversing range can be limited by software limit switches and working areas, see Chapter "Axis Monitoring (A3)".

The traversing range for rotary axes can be limited via machine data, see Chapter "Rotary Axes (R2)".

7.1.3 Input/display resolution, computational resolution

The following types of resolution, e.g. resolution of linear and angular positions, velocities, accelerations, and jerk, must be differentiated as follows:

- Input resolution - data is input via the control panel or part programs.
- Display resolution - data is displayed via the control panel.
- Computational resolution - data input via the control panel or part program is displayed internally.

The input and display resolution are determined by the control panel used (display machine data), whereby the display resolution for position values/spindle speed can be modified using MD203 DISPLAY_RESOLUTION (display resolution, metric linear position, general angular position) or MD205 DISPLAY_RESOLUTION_SPINDLE (display resolution for spindle speed).

MD204 DISPLAY_RESOLUTION_INCH can be used to configure the display resolution for linear position values in inches.

For the programming of part programs, the input resolutions listed in the Programming Guide apply.

The desired computational resolution is defined via MD10200 INT_INCR_PER_MM (computational resolution for linear positions) or MD10210 INT_INCR_PER_DEG (computational resolution for angular positions). It is independent of the input/display resolution but should have at least the same resolution.

The max. number of places after the decimal point for position values, velocities etc. in the part program and the number of places after the decimal point for tool offsets, zero offsets etc. (and therefore also for the maximum possible accuracy) are defined with the calculation resolution.

The accuracy of angle and linear positions is limited to the computational resolution by rounding the product of the programmed value with the computational resolution to an integer number. To make the rounding clear, powers of 10 should be used for the calculation resolution.

Example of rounding:

Computational resolution: 1000 increments/mm

Programmed path: 97.3786 mm

⇒ effective value = 97.379 mm

7.1.4 Standardizing physical quantities of machine and setting data

Machine and setting data that possess a physical quantity are interpreted in the input/output units below depending on whether the metric or inch system is selected:

| Physical quantity: | Input/output units for standard basic system: | |
|--------------------|---|-------------|
| | Metric | Inch |
| Linear position | 1 mm | 1 inch |
| Angular position | 1 degrees | 1 degrees |
| Linear velocity | 1 mm/min. | 1 inch/min. |

7.2 Metric/inch measuring system

| Physical quantity: | Input/output units for standard basic system: | |
|-------------------------------------|---|-----------------------|
| | Metric | Inch |
| Angular velocity | 1 rpm | 1 rpm |
| Linear acceleration | 1 m/s ² | 1 inch/s ² |
| Angular acceleration | 1 rev/s ² | 1 rev/s ² |
| Linear jerk | 1 m/s ³ | 1 inch/s ³ |
| Angular jerk | 1 rev/s ³ | 1 rev/s ³ |
| Time | 1 s | 1 s |
| Position controller servo gain | 1/s | 1/s |
| Rev. feedrate | 1 mm/rev | 1 inch/rev |
| Compensation value linear position | 1 mm | 1 inch |
| Compensation value angular position | 1 degree | 1 degree |

7.2 Metric/inch measuring system

The control system can operate with the inch or the metric measurement system. The initial state is defined via MD10240 SCALING_SYSTEM_IS_METRIC (metric basic system). Depending on the setting in the MD, all geometric values are interpreted either as metric or inch values. All manual settings also refer to this basic setting (e.g. handwheel, INC, JOG feedrate), as do zero offsets, tool offsets, etc. and the associated displays.

Switching over the measuring system is simplified considerably by setting MD10260 CONVERT_SCALING_SYSTEM=1:

- Availability of an MMC softkey to change the dimension system.
- Automatic conversion of NC active data when the dimension system is changed-over.
- Data backup is performed with the actual measuring system identifier.

7.2.1 Conversion of basic system by part program

When programming, it is possible to change over between dimension systems for some workpiece-related specifications with G70/G71 and with G700/G710. Data influenced by G70/G71/G700/G710 is described in the Programming Guide.

When changing the dimension system via HMI softkey, these initial settings are **automatically** initialized for the dimension system with G700 or G710.

Application

With this function it is possible, for example, with a metric basic system, to machine an inch thread in a metric part program. Tool offsets, zero offsets and feedrates remain metric.

Machine data is displayed on the screen using the basic system selected in MD10240 SCALING_SYSTEM_IS_METRIC (metric basic system).

Machine coordinates, tool data and zero offsets are always displayed in the initial state, workpiece coordinate displays are in the current setting.

Note

If programs including data sets (zero offset, tool offset) programmed in the measuring system, which differs from the basic system, are read in from an external source, the initial state must **first** be changed via machine data MD10240 SCALING_SYSTEM_IS_METRIC.

Data exchange with the PLC of interface signals containing dimension information, e.g. feedrate for path and positioning axes, is always carried out in the selected basic system.

G700/G710 extends the functionality of G70/G71 as follows:

1. The feedrate is interpreted in the programmed measuring system:

- G700: length parameters [inch]; feedrates [inch/min]
- G710: length parameters [mm]; feedrates [mm/min]

The programmed feed is modal and therefore remains active after subsequent G70/G71/G700/G710 commands. If the feedrate is to apply in the new G70/G71/G700/G710 context, it must be re-programmed.

2. System variables and machine data specifying lengths in the part program are read and written in the programmed measuring system.

This allows you to implement part programs that are independent of the current default measuring system.

Comparison of the effect of G70 and G700 on machine data and system variables in the part program:

- with G70: reading/writing in the basic system
- with G700: reading/writing in the programmed measuring system

Comparison: G70/G71-G700/G710

Where:

P: Data is read/written in the programmed measuring system.

G: Data is read/written in the basic system (MD10240)

R/W: Read/Write

Table 7- 2 Comparison

| Area | G70/G71 | G700/G710 |
|----------------------------------|------------------|------------|
| | R/W part program | |
| Display, decimal places (WCS) | P/P | P/P |
| Display, decimal places (MCS) | G/G | G/G |
| Feed rates | G/G | P/P |
| Positional data X, Y, Z | P/P | P/P |
| Interpolation parameters I, J, K | P/P | P/P |

| Area | G70/G71 | G700/G710 |
|------------------------------------|---------|------------|
| Circle radius (CR) | P/P | P/P |
| Polar radius (RP) | P/P | P/P |
| Pitch | P/P | P/P |
| Programmable offset, rotation,... | P/P | P/P |
| Programmable offset G54, G55, etc. | G/G | P/P |
| Working area limits (G25/G26) | G/G | P/P |
| Tool offsets | G/G | P/P |
| Length-related machine data | G/G | P/P |
| Length-related setting data | G/G | P/P |
| Length-related system variables | G/G | P/P |
| R parameters | G/G | G/G |
| Siemens cycles | P/P | P/P |
| Jog/handwheel increment factor | G/G | G/G |

7.2.2 Manual switchover of the basic system

General

The dimension system for the whole machine is changed using an HMI softkey "Switch to mm > inch" or "Switch to inch > mm" located in the "POSITION" → "Settings" area. The softkey can only be accessed in JOG or MDA mode. The switchover is only accepted if:

- The channel is in reset status
- The axes are not being traversed.

Actions such as NC start or mode change are disabled for the duration of the changeover.

If the switchover cannot be performed, this is indicated by a message in the user interface. These measures ensure that a consistent set of data is always used for a running program with reference to the system of measurement.

The actual switchover of the system of measurement is performed internally by writing all the necessary machine data and subsequently activating them with a Reset.

MD10240 SCALING_SYSTEM_IS_METRIC and the corresponding G70/G71/G700/G710 settings in MD20150 GCODE_RESET_VALUES are switched over automatically and consistently. The value of machine data: MD20150 GCODE_RESET_VALUES[12] (initial setting of the G groups) varies between G700 and G710.

This process takes place independently of the protection level currently set.

Note

MD20150 can only be read/written in expert mode (protection level 1).

System data

When changing over the measuring system, from the view of the user, all length-related specifications are converted to the new measuring system automatically. This includes:

- Positions
- Feed rates
- Acceleration rates
- Jerk
- Tool offsets
- Compensation values
- Machine data
- Jog and hand wheel factors

After the changeover, all of the above data is available in the physical quantities described in Section "Standardizing physical quantities of machine and setting data".

Data, for which no unique physical units are defined, such as R parameters, is not converted automatically. The user is prompted to take the current valid measuring system MD10240 SCALING_SYSTEM_IS_METRIC into consideration.

The current dimension system setting can be read at the PLC interface via the "inch system" signal (V2700 0001.7).

Reference point

The reference point is retained. It is not necessary to repeat referencing.

Input resolution and computational resolution

The input/computational resolution is set in the control via MD10200 INT_INCR_PER_MM. The default setting for a metric system is 1000 (0.001 mm). inches is required for an inch system.

Examples:

$$1 \text{ inch} = 25.4 \text{ mm} \Rightarrow 0.0001 \text{ inch} = 0.00254 \text{ mm} = 2.54 \text{ } \mu\text{m}$$

To be able to program and display the last 40 mm, MD10200 must be assigned a value of 100000.

Only with this identical setting for both measuring systems is it possible to change the measuring system without a significant loss of accuracy. Once MD10200 has been set to this value, it will not need to be changed each time the measuring system is switched over.

Jog and increment weighting

MD31090 JOG_INCR_WEIGHT consists of two values containing axis-specific increment weighting factors for each of the two dimension systems. Depending on the current setting in MD10240 SCALING_SYSTEM_IS_METRIC, the control automatically selects the appropriate value.

Note

MD31090 JOG_INCR_WEIGHT can only be read/written in "expert mode" access level (protection level 1)!

Default setting for each axis:

- Metric: MD31090 JOG_INCR_WEIGHT[0]=0.001 mm
- Inch: MD31090 JOG_INCR_WEIGHT[1]=0.00254 mm => 0.0001 inch

Data backup

Data sets which can be read out separately from the control and which contain data relating to the dimension system are given an INCH or METRIC identifier corresponding to MD10240 SCALING_SYSTEM_IS_METRIC. This specifies the dimension system in which the data were originally read out.

This information is intended to prevent data sets from being read into the control system with a measuring system, which is different from the active system. In this case, alarm 15030 is triggered and the write process is interrupted.

Since the language instruction is also evaluated in part programs, these can also be "protected" against operator errors as described above. You can therefore prevent part programs containing, e.g. only metric data, from running on an inch dimension system.

7.3 Setpoint/actual-value system

7.3.1 General

Block diagram

A control loop with the following structure can be configured for every closed-loop controlled axis/spindle:

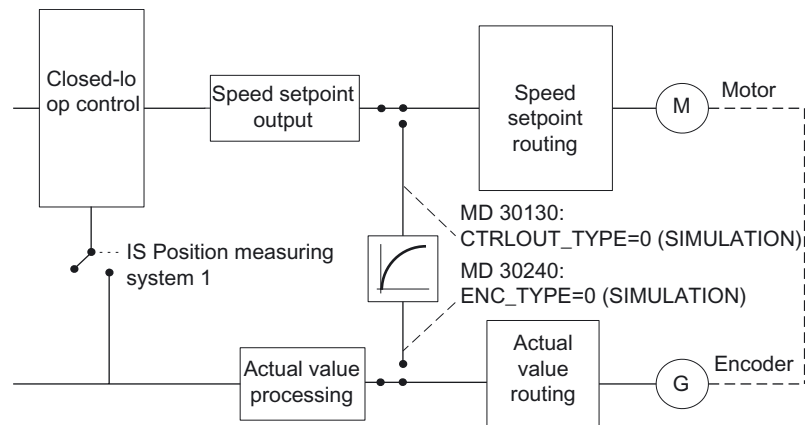


Figure 7-1 Block diagram of a control loop

Setpoint output

A setpoint can be output for each axis/spindle. The setpoint output at the setting device is done digitally or, in the case of analog spindles 10 V unidirectionally or bidirectionally.

Simulation axes

The speed control loop of an axis can be simulated for test purposes. The axis "traverses" with a following error, similar to a real axis.

A simulation axis is defined by setting MD30130 CTRLOUT_TYPE[n] (setpoint output type) and MD30240 ENC_TYPE[n] (actual-value acquisition type) to "0".

As soon as the standard machine data has been loaded, the axes become simulation axes.

The setpoint and actual value can be set to the reference point value with reference point approach.

MD30350 SIMU_AX_VDI_OUTPUT (output of axis signals for simulation axes) can be set to define whether axis-specific interface signals are output to the PLC during the simulation process.

7.3.2 Drives with DRIVE-CLiQ

General

With the 802D sl system, drive function of the Control Unit of SINAMICS S120 (SINAMICS_I) are integrated in the control system.

From this integrated controller, the connection to further SINAMICS modules is realized via DRIVE-CLiQ. This connection can extend up to motors with SMI interface. The topology is identified automatically when starting up the SINAMICS drive

Reference:

/BE/ Operating Instructions

Configuring drives with SINUMERIK 802 D sl using SINAMICS components

The control system provides preconfigured system data blocks for various drive configurations. These configurations are set via MD11240 PROFIBUS_SBD_NUMBER[2].

| |
|--|
| NOTICE |
| Do not change MD11240 [1], [3]. They are reserved for Siemens. |

See the comprehensive machine data description in the Chapter "Data description" for the full range of selection options.

The drive is assigned a drive number via the slot in the DRIVE-CLiQ system. Please note that the ALM (Active Line Module) is also assigned a number in this system. These numbers are displayed in the screen of the drive machine data after starting up the SINAMICS drives.

This drive number is used for assignment of the NC axes to the drives. The relevant values are entered in the machine data 30110 and 30220 (see also next Section "Speed setpoint and actual value assignment").

Example

The following modules are used for a 3-axis milling machine with spindle:

- One active line module (ALM)
- Two one-axis modules (for spindle and Z axis)
- One double-axis module (for X and Y axes)

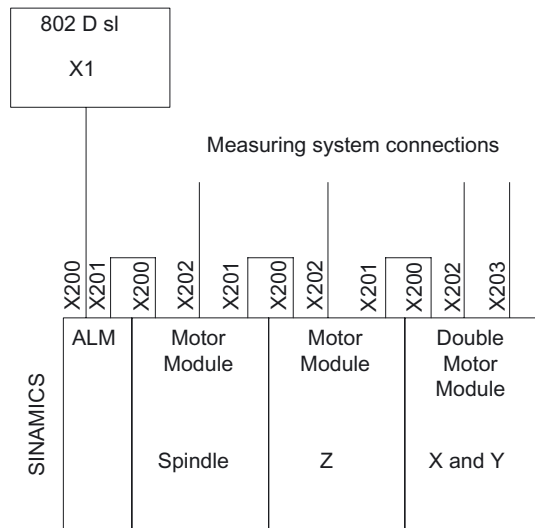


Figure 7-2 Examples: Milling machine with three axes and spindle

The configuration is performed via machine data.

Table 7-3 Configuration

| FD | X | Y | Z | SP |
|-------|---|---|---|----|
| 30100 | 5 | 5 | 5 | 5 |
| 30110 | 3 | 4 | 2 | 1 |
| 30120 | 1 | 1 | 1 | 1 |
| 30130 | 1 | 1 | 1 | 1 |
| 30134 | 0 | 0 | 0 | 0 |
| 30200 | 1 | 1 | 1 | 1 |
| 30230 | 1 | 1 | 1 | 1 |
| 30220 | 3 | 4 | 2 | 1 |

Note

In the case of dual-axis power modules, both drives (A and B) are each assigned to an axis.

The commissioning of the drives is performed via the drive machine data. Thus, the topology can be identified and the motor data set up.

MD13060 DRIVE_TELEGRAM_TYPE[n] is used to set the default message frame type for communication with each drive. When using the SINAMICS drive, the required value = 116 is already set by default. The message frame type 116 must also be set in SINAMICS. For the relevant procedure, please refer to the Start-up Guide.

The SINAMICS drive is thus ready for traversing. Further fine optimization can be performed later via the V24 connection using the Starter tool. The V24 connection must be activated for the connection using the "SYSTEM" > "PLC" > "STEP 7 connection" menu item.

7.3.3 Speed setpoint and actual-value routing

Prerequisite for routing

All NC machine axes must be uniquely defined in MD10000 AXCONF_MACHAX_NAME_TAB[n] (machine axis name). This name must be unique across the whole system.

Note

To connect an analog (instead of a digital) spindle, install the MCPA module. When doing so, various additional notes must be observed. These can be found in:

Reference:

/BE/ Operating Instructions, Section "Connecting an analog spindle"

Speed setpoint routing

The speed setpoint assignment is illustrated in the diagram below.

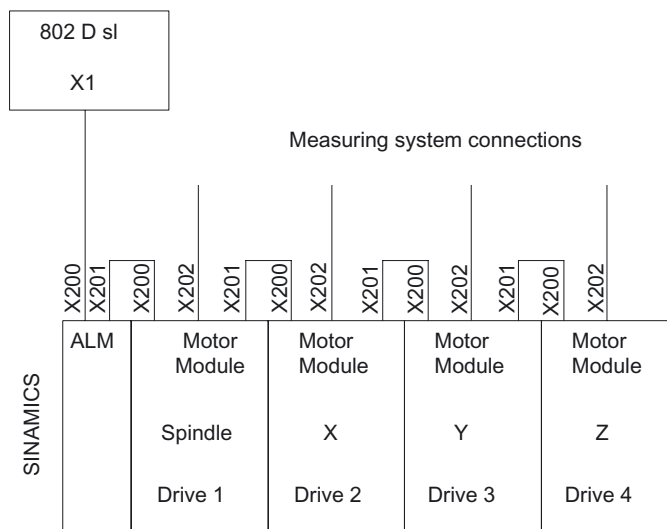


Figure 7-3 Speed setpoint routing, example

The following machine data of each machine axis must be parameterized:

- MD30110 CTRLOUT_MODULE_NR[n] Assignment of the drive number
 - [n] = 1 (drive number 1) → SP spindle = machine axis 4
 - [n] = 2 (drive number 2) → X1 axis = machine axis 1
 - [n] = 3 (drive number 3) → Y1 axis = machine axis 2
 - [n] = 4 (drive number 4) → Z1 axis = machine axis 3
 - [n] = 5 (drive number 5) → A1 axis = machine axis 5
- MD30130 CTRLOUT_TYPE[n] Setpoint output type. The speed setpoint output type is entered here

Actual-value routing

Due to the fixed assignment of the encoder (measuring system) to the drive, the actual value assignment must be performed identically to the speed setpoint assignment (the same drive number). For the actual-value assignment, please refer to the diagram below. The parameterization is described in the associated machine data:

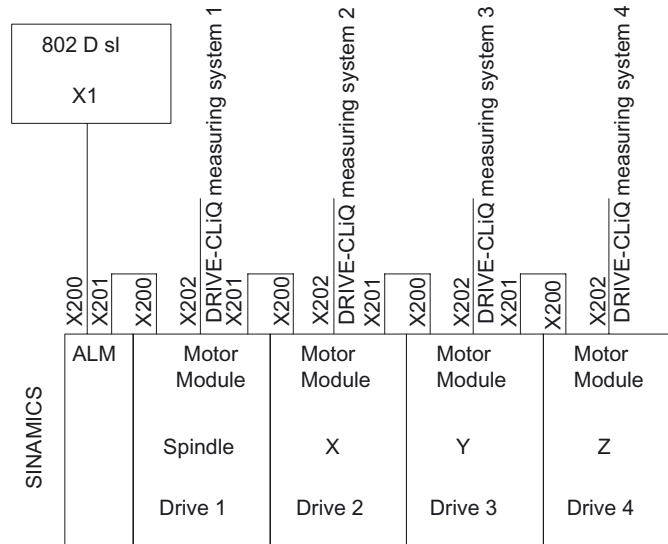


Figure 7-4 Actual-value assignment, example

The following machine data of each machine axis must be parameterized:

- MD30220 ENC_MODULE_NR[n] Assignment of the drive number
 - [n] = 1 (drive number 1) → SP spindle = machine axis 4
 - [n] = 2 (drive number 2) → X1 axis = machine axis 1
 - [n] = 3 (drive number 3) → Y1 axis = machine axis 2
 - [n] = 4 (drive number 4) → Z1 axis = machine axis 3
 - [n] = 5 (drive number 5) → A1 axis = machine axis 5
- MD30240 ENC_TYPE[n] Type of actual-value assignment; enter the encoder type you are using here.

Special features

MD30110 CTRLOUT_MODULE_NR[n] and MD30220 ENC_MODULE_NR[n] of one machine axis must have the same drive number.

For operation of a digital spindle with a second direct position encoder, the following MD must be set for the actual-value assignment. A reloadable SDB must be loaded from the toolbox, granting access to the message frame type = 116 for the relevant drives. The message frame type 116 must also be set in SINAMICS.

MD13060 DRIVE_TELEGRAM_TYPE[0-5] = 116

MD30220 ENC_MODULE_NR[n] = 3 (module no. for input 2nd spindle measuring system)

MD30230 ENC_INPUT_NR[n] = 2 (measuring system slot on the module)

MD32110 ENC_FEEDBACK_POL[n] = -1 (correct the direction)

7.3.4 Speed setpoint output

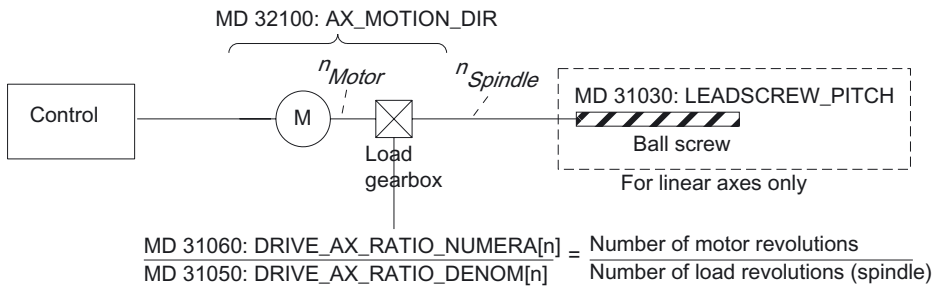


Figure 7-5 Speed setpoint output

Traversing direction

MD32100 AX_MOTION_DIR (traversing direction) can be used to reverse the direction of motion of the axis (without affecting the control sense of the position control).

Maximum speed setpoint

The maximum speed setpoint is defined with MD36210 CTRLOUT_LIMIT. The percent value refers to the speed (100%) at which the axis velocity of MD32000 MAX_AX_VELO is reached. A value greater than 100 % contains the required control margin for the position control of axes.

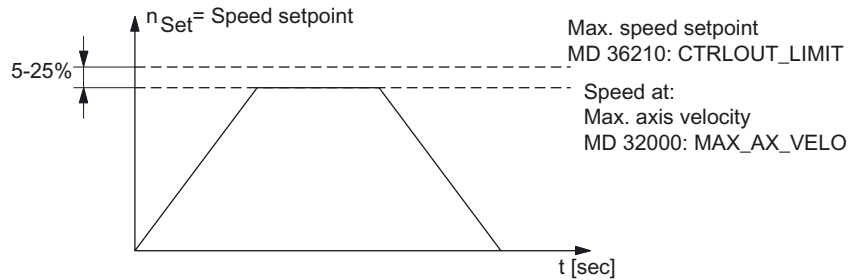


Figure 7-6 Speed setpoint output

If settings beyond the limit are made, the value in MD36210 is used as the limiter, an alarm is output and the axes brought to a halt. See Chapter "Axis monitoring functions" for more details.

With an analog spindle the maximum speed that can be output is limited by the maximum setpoint output voltage of 10 V. The value in MD36210 CTRLOUT_LIMIT should not be greater than the speed value that is reached at this voltage (100%).

Note

For special considerations for controlling a spindle, see Chapter "Spindles (S1)".

7.3.5 Actual-value processing

Actual-value resolution

In order to be able to create a correctly closed position closed-loop control, the control system must be informed of the valid actual-value resolution.

Depending on the type of axis (linear axis, rotary axis/spindle, analog spindle) and the type of actual-value resolution (direct, indirect), the following machine data must be parameterized to calculate the actual-value acquisition for the main applications:

| Machine data | Linear axis | Rotary axis | Spindle | | |
|---|------------------|------------------|------------------|--------------------|--------------------------|
| | Encoder on motor | Encoder on motor | Encoder on motor | Encoder on machine | Without measuring system |
| MD30200 NUM_ENC (number of encoders) | 1 | 1 | 1 | 1 | 0 |
| MD30300 IS_ROT_AX (rotary axis) | 0 | 1 | 1 | 1 | 1 |

7.3 Setpoint/actual-value system

| Machine data | Linear axis | Rotary axis | Spindle | | |
|--|------------------|------------------|------------------|--------------------|--------------------------|
| | Encoder on motor | Encoder on motor | Encoder on motor | Encoder on machine | Without measuring system |
| MD31040 ENC_IS_DIRECT[0] (encoder is connected directly at the machine) | 0 | 0 | 0 | 1 | 1 |
| MD31020 ENC_RESOL[0] (encoder pulses per revolution) | Pulses/rev | Pulses/rev | Pulses/rev | Pulses/rev | - |
| MD31030 LEADSCREW_PITCH (leadscrew pitch) | mm/rev. | - | - | - | - |
| MD31080 DRIVE_ENC_RATIO_NUMERA[n] (measuring gearbox numerator) | 1 | 1 | 1 | load revs | - |
| MD31070 DRIVE_ENC_RATIO_DENOM[n] (measuring gearbox denominator) | 1 | 1 | 1 | encoder revs | - |
| MD31060 DRIVE_AX_RATIO_NUMERA[n] (load gearbox numerator) | motor revs | motor revs | motor revs | See note *) | - |
| MD31050 DRIVE_AX_RATIO_DENOM[n] (load gearbox denominator) | Leadscrew revs | load revs | load revs | See note *) | - |

"-" Does not apply to this combination.
The coding of the machine data MD... DRIVE_AX_... index [n] is: [closed-loop control parameter set no.]: 0-5

Note

*) These MD are not required for encoder matching (path evaluation). However, they must be entered correctly for the setpoint calculation! Otherwise the required servo gain factor will not be set.

Variants of actual-value acquisition

The associated machine data for different variants of actual-value acquisition is described below.

Linear axis with rotary encoder on motor

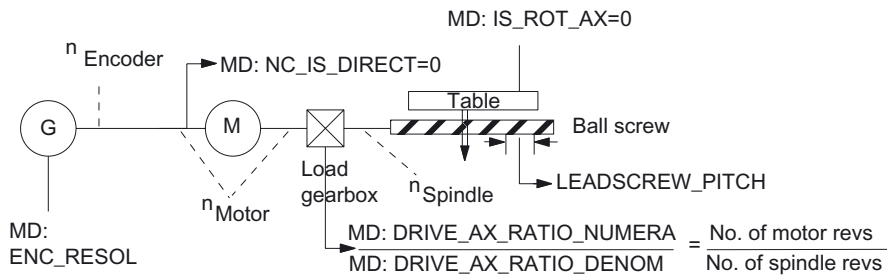


Figure 7-7 Linear axis with rotary encoder on motor

Rotary axis with rotary encoder on motor

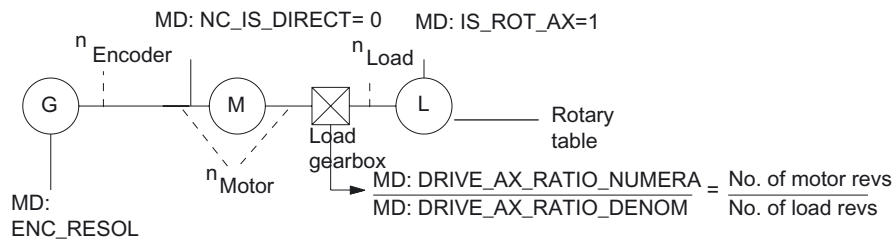


Figure 7-8 Rotary axis with rotary encoder on motor

Spindle with rotary encoder on the machine

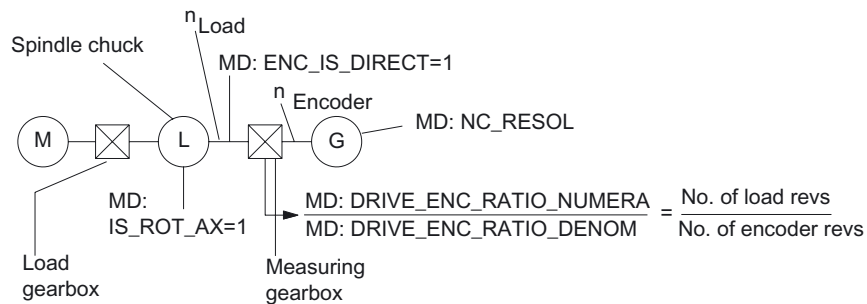


Figure 7-9 Spindle with rotary encoder on the machine

Note

With MD32110 ENC_FEEDBACK_POL (sign of actual value), the leading sign of the actual-value acquisition and also the control sense of the position control can be changed.

7.3.6 Evaluation of internal drive variables

Note

This function is under license and is not available with version T/M value.

If the corresponding option is activated without a valid license, alarm 8081 "%1 option(s) that has (have) not been licensed using a license key was (were) set" is output. It will not be possible to operate the machine as normal.

For information on operations relating to "Setting (an) option(s)", please refer to the section titled "Licensing in SINUMERIK 802D sl" in the "Turning, Milling, Grinding, Nibbling" Operating Instructions.

Function

To ensure that the internal drive variables are available for evaluation in the NC, these are transferred from the drive as additional process data (PDA) during cyclic PROFIBUS communication and saved by the operating system in system variables.

The additional PDA is part of standard telegram 116.

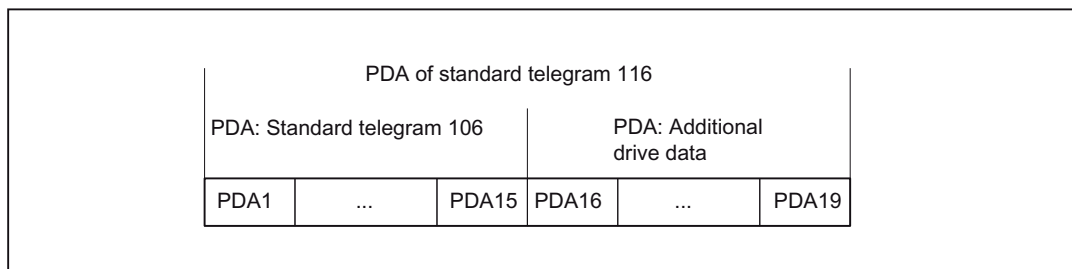


Figure 7-10 Standard telegram 116 with process data

Activation

To transfer drive variables to individual system variables you must set the following machine data:

MD36730 DRIVE_SIGNAL_TRACKING = 1

Table 7-4 Specific drive variables

| PDA | Drive variable | System variable | Drive MD |
|-----|--|-----------------|----------|
| 16 | Torque utilization | \$VA_LOAD[X] | r0081[0] |
| 17 | Smoothed torque setpoint | \$VA_TORQUE[X] | r0079[1] |
| 18 | Smoothed actual active power value | \$VA_POWER[X] | r0082[1] |
| 19 | Smoothed, torque-generating actual current value | \$VA_CURR[X] | r0078[1] |

The system variables can be smoothed with a PT1 filter. The filter time constant is defined using the following MD:

MD32920 AC_FILTER_TIME (smoothing filter time constant for adaptive control)

Specifying 0 seconds for the smoothing time deactivates the filter.

Data formats

The user must take the following points into account with regard to the data formats of the process data stored in the system variables:

- The PDA is transferred to the telegram in the **16-bit integer without leading sign (UINT16)** format. They are stored in the system variables in the **32-bit integer with leading sign (INT32)** format. During the necessary format conversion, bit 15 of the 16-bit integer PDA value without leading sign is transferred to bits 16 to 31 of the signed 32-bit integer value with leading sign in the system variable.
- For the physical unit as well as the drive-end weighting of the drive actual values transferred in the additional PDA, please refer to the data description of the specific drive documentation.

Consistency check

While the SINUMERIK 802D sl is powering up, a consistency check is performed on the parameters for cyclic PROFIBUS communication that are relevant for process data configuration:

- NC: MD13060 DRIVE_TELEGRAM_TYPE[n] (drive telegram type)
- Drive: p0922 (PROFIdrive telegram selection)

If the drive telegram type in MD13060 is different from the one defined in p0922, alarm 26015 "Axis %1 machine data %2[%3] value not permissible" is output (in this case, %2 corresponds to the identifier for MD13060).

Application examples

- System variables during synchronization with a modal effect:

```
ID=1 WHENEVER TRUE DO $R0=$VA_LOAD[X]
```

- System variables in a part program:

```
...
WHENEVER $VA_LOAD[X]>80 DO $AC_OVR=80
G1 G90 X200 F10000
...
```

7.4 Closed-loop control

General

The closed-loop control of an axis consists of the current and speed control loop of the drive plus a higher-level position control loop in the NC.

The speed and current control are described in:

Reference:

/FH1/ Function Manual SINAMICS S120, Section "Operating modes"

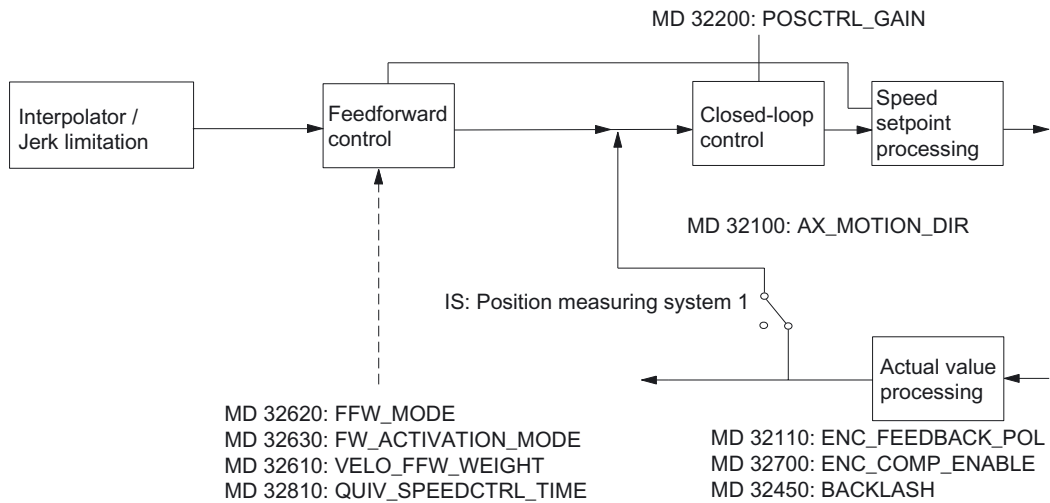


Figure 7-11 Principle of position control for an axis/spindle

For a description of jerk limiting, see the section titled "Acceleration"

For a description of precontrol, backlash compensation, and leadscrew error compensation, refer to the section titled "Compensation (K3)".

Servo gain factor

To ensure that only low contour deviations occur in continuous-path mode, a high servo gain factor MD32200 POSCTRL_GAIN[n] (position controller servo gain) is required.

The index[n] of the machine data has the following code:
 [closed-loop control parameter set no.]: 0 - 5

However, if the servo gain factor is too high, this will result in instability, overshoots, and possibly impermissibly high loads on the machine.

The maximum permissible servo gain factor depends on the following:

- Design and dynamic response of the drive
 (rise time, acceleration and braking capacity)
- Quality of the machine (elasticity, vibration damping)
- Position-control cycle

$$K_v = \frac{\text{Velocity} \quad [\text{m/min}]}{\text{Following error} \quad [\text{mm}]} \quad \text{Unit of servo gain factor acc. to VDI standard}$$

Parameter sets of the position controller

Position control can operate with 6 different parameter sets. This information is used for

- Fast adaptation of position control to altered machine characteristics during operation, e.g. a spindle gear change.
- Matching the dynamic response of an axis to another axis, e.g. during tapping.

The following machine data can be changed as related units of data by switching over the parameter set during operation:

- MD31050 DRIVE_AX_RATIO_DENOM[n] (load gearbox denominator)
- MD31060DRIVE_AX_RATIO_NUMERA[n] (load gearbox numerator)
- MD32200 POSCTRL_GAIN[n] (servo gain factor).
- MD32810 EQUIV_SPEEDCTRL_TIME[n] (equivalent time constant speed control loop for precontrol)
- MD36200 AX_VELO_LIMIT[n] (threshold value for velocity monitoring)

Index[n] of the machine data is coded as follows: [closed-loop control parameter set no.]: 0-5

Spindle parameter sets:

In the case of spindles, each gear stage is assigned its own parameter set. The relevant parameter set is activated in accordance with the "Actual gear stage" IS (V380x 2000.0 to .2). The machine data that is changed over with the spindle gear stage is listed in the section titled "Spindles".

| IS "Actual gear stage" | Active parameter set |
|------------------------|----------------------|
| 000 | 2 (index=1) |
| 001 | 2 (index=1) |
| 010 | 3 (index=2) |
| 011 | 4 (index=3) |
| 100 | 5 (index=4) |
| 101 | 6 (index=5) |

Axis parameter sets

- Parameter set 1 (index=0) is always activated for axes not involved in tapping or thread cutting.
- For axes involved in tapping or thread cutting, the same parameter set number as for the current gear stage of the spindle is activated.

Note

If, for example, a load gearbox is active for an axis, this ratio (numerator, denominator) must be entered in all other parameter sets used for threads in addition to the parameter set with index=0.

The current parameter set is displayed in the "System" → "Service Display" operating area on the "Service Axes" screen.

7.5 Data lists

7.5.1 Machine data

| Number | Identifier | Name |
|---------------------------|---------------------------------|--|
| Specific to control panel | | |
| 203 | DISPLAY_RESOLUTION | Display resolution |
| 204 | DISPLAY_RESOLUTION_INCH | Display resolution for INCH system of measurement |
| 205 | DISPLAY_RESOLUTION_SPINDLE | Display resolution for spindle |
| General information | | |
| 10000 | AXCONF_MACHAX_NAME_TAB[0]...[5] | Machine axis name |
| 10200 | INT_INCR_PER_MM | Computational resolution for linear positions |
| 10210 | INT_INCR_PER_DEG | Computational resolution for angular positions |
| 10240 | SCALING_SYSTEM_IS_METRIC | Basic system metric |
| 10260 | CONVERT_SCALING_SYSTEM | Basic system switchover active |
| 11240 | PROFIBUS_SDB_NUMBER[0]...[3] | SDB number |
| 13060 | DRIVE_TELEGRAM_TYPE[0]...[30] | Standard telegram type for PROFIdrive |
| Channel-specific | | |
| 20150 | GCODE_RESET_VALUES[n] | Reset G groups |
| Axis-specific | | |
| 30110 | CTRLOUT_MODULE_NR[0] | Setpoint assignment: Drive number/module number (analog spindle) |
| 30120 | CTRLOUT_NR[0] | Setpoint: Output to module |
| 30130 | CTRLOUT_TYPE[0] | Output type of setpoint |
| 30134 | IS_UNIPOLAR_OUTPUT[0] | Setpoint output is unipolar (analog spindle) |
| 30200 | NUM_ENCS | Number of encoder = 1 (spindle without encoder value = 0) |
| 30220 | ENC_MODULE_NR[0] | Actual value: Drive module number/measuring circuit number |
| 30230 | ENC_INPUT_NR[0] | Actual value: Input number on module/measuring circuit board |
| 30240 | ENC_TYPE[0] | Type of actual-value acquisition (position actual value) |
| 30300 | IS_ROT_AX | Rotary axis |
| 30350 | SIMU_AX_VDI_OUTPUT | Output of axis signals for simulation axes |
| 31000 | ENC_IS_LINEAR[0] | Direct measuring system (linear scale) |
| 31010 | ENC_GRID_POINT_DIST[0] | Distance between reference marks on linear scales |
| 31020 | ENC_RESOL[0] | Encoder pulses per revolution |
| 31030 | LEADSCREW_PITCH | Leadscrew pitch |
| 31040 | ENC_IS_DIRECT[0] | Encoder is connected directly to the machine |
| 31050 * | DRIVE_AX_RATIO_DENOM[0]...[5] | Denominator load gearbox |
| 31060 * | DRIVE_AX_RATIO_NUMERA[0]...[5] | Numerator load gearbox |
| 31070 | DRIVE_ENC_RATIO_DENOM[0] | Measuring gearbox denominator |
| 31080 | DRIVE_ENC_RATIO_NUMERA[0] | Measuring gearbox numerator |
| 32000 | MAX_AX_VELO | Maximum axis velocity |
| 32100 | AX_MOTION_DIR | Travel direction |

| Number | Identifier | Name |
|---------|-------------------------------|---|
| 32110 | ENC_FEEDBACK_POL[0] | Sign actual value (feedback polarity) |
| 32200 * | POSCTRL_GAIN[0]...[5] | Servo gain factor Kv |
| 32450 | BACKLASH[0] | Backlash |
| 32700 | ENC_COMP_ENABLE[0] | Interpolatory compensation |
| 32810 * | EQUIV_SPEEDCTRL_TIME[0]...[5] | Equivalent time constant speed control loop for feedforward control |
| 32920 | AC_FILTER_TIME | Smoothing filter time constant for adaptive control |
| 35100 | SPIND_VELO_LIMIT[0]...[5] | Maximum spindle speed |
| 36200 * | AX_VELO_LIMIT[0]...[5] | Threshold value for velocity monitoring |
| 36210 | CTRL_OUT_LIMIT[0] | Maximum speed setpoint |
| 36730 | DRIVE_SIGNAL_TRACKING | Detection of additional drive actual values |

The machine data marked with * is contained in a parameter set of the position controller.

7.5.2 Interface signals

| Number | .Bit | Name |
|---------------------|----------|------------------------------|
| General information | | |
| V2700 0001 | .7 | INCH dimension system |
| Axis-specific | | |
| V380x 2000 | .0 to .2 | Actual gear stage of spindle |

Manual and Handwheel Travel (H1)

8.1 General characteristics of traversing in JOG

JOG mode

Axes/Spindles can be traversed manually in JOG mode. The active mode is transmitted to the PLC via the IS "Active mode: JOG" (V3100 0000.2) and is visible in the display, see also Chapter "Operating Modes, Program Operation (K1)".

Traversing possibilities

Traversing the axes can be done via the traverse keys of a connected machine control panel (manual travel) or via connected handwheels (handwheel jogging).

All machine axes can be traversed simultaneously using keys (with an appropriate version of a user-specific machine control panel) or via handwheel, depending on the number of handwheels connected. If several machine axes are moved simultaneously, there is no interpolatory relation.

Coordinate systems

The user has the option of traversing axes in the coordinate systems:

- Machine coordinate system (MCS); machine axes manually traversable
- Workpiece coordinate system (WCS); geometry axes manually traversable

Machine functions

Variants exist for **manual traverse** (the so-called machine functions):

- Continuous traversal
- Incremental traversing (INC, preset number of traversing increments). An increment is evaluated with 0.001 mm if the basic system setting is metric.

The PLC user program transfers a user-specific machine function queued at the machine control interface to the relevant PLC/NCK interface. Here the axis-specific NCK/PLC interface should be used for a machine axis/spindle, and the channel-specific NCK/PLC interface should be used for a geometry axis or valid for all axes/spindles and geometry axes: Signals in the operating mode range (see also following section).

Handwheel jogging

The axes can also be traversed via the handwheel in MCS or WCS. Incremental traversing (INC...) must be set to evaluate the handwheel pulses (see Section "Handwheel traversal in JOG").

Traversing the geometry axes

If workpieces whose workpiece coordinate system is not parallel to the machine coordinate system are being machined (inclined clamping, programmed rotation active in the contour), traversing can be done along the axes of the workpiece coordinate system via the traverse keys or handwheel. In the stopped state, switch from the operating mode AUTO to JOG and traverse a geometry axis instead of a machine axis. Depending on the active rotation of the workpiece coordinate system, between one and three machine axes move.

If a machine axis is traversed, this cannot also be moved via the traverse keys of a geometry axis. The traversing motion of the machine axis must first have been completed - otherwise alarm 20062 "Axis already active" is output. Two geometry axes can be traversed simultaneously with the handwheels 1 and 2.

Note

A separate, channel-specific PLC interface supplies geometry axes.

Transverse axis in "turning" technology

A geometry axis is defined as a transverse axis. If radius programming (DIAMOF) is selected instead of diameter programming (DIAMON), the following must be noted when traversing in JOG:

- Continuous traversing: There are no differences when a transverse axis is traversed continuously.
- Incremental traversing: Only half the distance of the selected increment size is traversed.
- Traversing with the handwheel: As for incremental travel, with the handwheel only half the distance is traversed per handwheel pulse.

Spindle manual travel

The spindle can also be traversed manually in the JOG mode. Essentially the same conditions apply as for manual traverse of machine axes. With JOG, the spindle can be traversed via the traverse keys/ IS "continuous" or "INC...". The mode is selected and activated via the axis-/spindle-specific PLC interface as for the axes.

Spindle manual travel is possible in positioning mode (spindle in position control) or in open-loop control mode. The parameter set (machine data) of the current gear stage applies.

Velocity

The velocity of the axes/spindle during manual traverse in JOG is defined by the following default values:

- For linear axes with the general SD41110 JOG_SET_VELO (JOG velocity with G94) or for rotary axes with SD41130 JOG_ROT_AX_SET_VELO (JOG velocity for rotary axes) or SD41200 JOG_SPIND_SET_VELO (JOG velocity for the spindle).
- If the corresponding SD is zero, the appropriate axis-specific MD32020 JOG_VELO (conventional axis velocity) applies. In this case, the value of the assigned machine axis is used for geometry axes: X->X1, Y->Y1, Z->Z1 (for default setting).

Rapid traverse override

If in the case of machine axes the rapid traverse override key is pressed at the same time as the traversing keys, then the movement is executed at the rapid traverse velocity set in axis-specific MD32010 JOG_VELO_RAPID (axis velocity in JOG mode with rapid traverse override).

The value of the assigned machine axis is used for geometry axes: X->X1, Y->Y1, Z->Z1 (for default setting). The separate PLC interface area of the geometry axes must be used for control.

Velocity override

The velocity at which axes traverse in JOG can also be influenced by the axis-specific feedrate override switch for machine axes, provided that axis-specific IS "Override active" (V380x 0001.7) is set. If the switch is set at 0%, the axis is not traversed - even if IS "Override active" is not set.

The channel-specific feedrate override switch applies to geometry axes, or, in the case of rapid traverse override, the rapid traverse override switch.

The activated spindle override switch applies to the spindle.

Acceleration

The maximum axis acceleration is defined with the axis-specific MD32300 MAX_AX_ACCEL. The acceleration can also be set via a preset characteristic curve in JOG mode. The possible settings are described in Chapter "Acceleration (B2)".

PLC interface

A separate PLC interface (VB 3200 1000, ff or VB 3300 1000, ff) exists for **geometry axes** (axes in WCS) that contains the same signals as the axis-specific PLC interface.

When the **spindle** is traversed manually, the PLC interface signals between the NCK and PLC have the same effect as for machine axes. Interface signals "Position reached with fine or coarse exact stop" are only set if the spindle is in position control.

8.1 General characteristics of traversing in JOG

In the case of interface signals that are only spindle-specific, while the spindle is traversing in JOG, the following should be noted:

- The following PLC interface signals to the spindle have no effect:
 - IS "Invert M3/M4" (V380x 2001.6)
 - IS "Set direction of rotation ccw" or "Set direction of rotation cw" (V380x 2002.7 or .6)
 - IS "Oscillation speed" (V380x 2001.5)
- The following PLC interface signals from the spindle are not set:
 - IS "Actual speed cw" (V390x 2001.7)
 - IS "Spindle in setpoint range" (V390x 2001.5)

| |
|--|
| NOTICE |
| A reset causes the manual traverse motion (axis/spindle) to be terminated with brake ramp. |

Limitations


The following limitations are active for manual travel:

- Working-area limitation (axis must be referenced)
- Software limit switches 1 or 2 (axis must be referenced)
- Hardware limit switches

The control ensures that the traversing movement is aborted as soon as the first valid limitation has been reached. Velocity control ensures that deceleration is initiated early enough for the axis to stop exactly at the limit position (e.g. software limit switch). Only when the hardware limit switch is triggered does the axis stop abruptly with "rapid stop".

An alarm is output when the corresponding limit is reached. The control automatically prevents further movement in this direction. The traversing keys and the handwheel have no effect in this direction.

| |
|---|
| NOTICE |
| The software limit switches and working area limits are only active if the axis has previously been referenced. |

| |
|---|
| NOTICE |
|  The function for retracting an axis that has approached the limit position depends on the machine manufacturer. Please refer to the machine manufacturer's documentation! |

For further information on working area limits and hardware and software limit switches see Chapter "Axis Monitoring (A3)".

8.2 Continuous travel

Selection

When JOG mode is selected, the active machine function "continuous" interface signal is set automatically:

- For geometry axes: V3300 1001.6, V3300 1005.6, V3300 1009.6
- For machine axes/spindle: V390x 0005.6

Continuous mode in JOG mode can also be selected via the PLC interface (IS "Machine function: continuous"). The PLC defines via the "INC inputs in mode group range active" interface signal (V2600 0001.0) the signal range within which INC/continuous signals are delivered to the NCK:

V2600 0001.0 = 1 → in the operating mode range: VB3000 0002,
valid for all axes

V2600 0001.0 = 0 → in the geometry axis / axis range:
VB3200 1001, VB3200 1005,
VB3200 1009, VB380x 0005

Traversing keys +/-

The plus and minus traversing keys are selected to move the relevant axis in the appropriate direction.

Traverse key signals PLC to NCK IS:

- For geometry axes (traverse in WCS):
V3200 1000.7/.6, V3200 1004.7/.6, V3200 1008.7/.6
- For machine axes / spindle (traverse in MCS): V380x 004.7/.6

If both traversing keys of an axis are pressed simultaneously, there is no traversing movement, or, if an axis is in motion, it is stopped.

Motion command +/-

As soon as a traverse request for an axis/spindle is active (e.g. after selection of a traverse key), the IS "Travel command+" or "Travel command-" is sent to the PLC (depending on selected traverse direction):

- For geometry axes: V3300 1000.7/.6, V3300 1004.7/.6, V3300 1008.7/.6
- For machine axes / spindle: V390x 004.7 / .6

Continuous travel in jog mode

The axis traverses for as long as the traverse key is held down if no axis limit is reached first. When the traversing key is released, the axis is decelerated to standstill and the movement comes to an end.

8.3 Incremental travel (INC)

Programming increments

The path to be traversed by the axis is defined by so-called increments (also called "incremental dimensions"). The required increment must be set by the machine user before the axis is traversed.

The setting is made on the machine control panel, for example. After the corresponding logic operation, the IS "Machine function: INC1 to INCvar" associated with the required increment must be set by the PLC user program after it has been correctly linked. The PLC defines via the "INC inputs in mode group range active" interface signal (V2600 0001.0) the signal range within which INC signals are delivered to the NCK:

V2600 0001.0 = 1 → in the operating mode range: VB3000 0002,
valid for all axes

V2600 0001.0 = 0 → in the geometry axis / axis range:
VB3200 1001, VB3200 1005,
VB3200 1009, VB380x 0005

The active machine function IS "INC..." is signaled by the NCK to the PLC:

- For geometry axes: V3300 1001.0, V3300 1005.0, V3300 1009.0 to .5
- For machine axes / spindle: V390x 0005.0 to .5

Settable increments

The operator can set different increment sizes:

- **Fixed increments** whose increment sizes are common to all axes: INC1, INC10, INC100, INC1000 (only via IS: INC10000).
- **A variable increment (INCvar)**. The increment setting for the variable increment can also be made for all axes using general SD41010 JOG_VAR_INCR_SIZE (size of the variable increment for INC/handwheel).

Incremental travel in inching mode

If the traversing key for the required direction (e.g. +) is pressed, the axis begins to traverse the increment that has been set. If the traversing key is released before the increment has been fully traversed, the movement is interrupted and the axis stops. If the same key is pressed again, the axis moves the remaining distance. Up to this point, the movement can still be interrupted by releasing the traversing key. Pressing the traversing key for the opposite direction does not have any effect, unless the increment has been fully traversed or the movement has been aborted.

Traverse keys and travel command

As for continuous traversing (see Section "Continuous travel")

Abort traversing movement

If you do not want to traverse the whole increment, the traverse movement can be aborted with RESET or "Delete distance-to-go" interface signal (V380x 0002.2).

8.4 Handwheel traversal in JOG

Selection

JOG mode must be active. The user must also set the increment INC1, INC10, etc., which applies to handwheel travel.

Up to 2 handwheels can be connected. This means that up to 2 axes can be traversed by handwheel simultaneously and independently.

A handwheel is assigned to the geometry or machine axes (WCS or MCS) via interface signals.

The axis to be moved as a result of rotating handwheel 1 to 2 can be set:

- Via the PLC user interface with IS "Activate handwheel 1 to 2"
 - For machine axis (traverse in MCS): V380x 0004.0 to .2
 - For geometry axis (traverse in WCS): V3200 0000.0 to .2, V3200 0004.0 to .2, V3200 0008.0 to .2.

The assignment is linked to the PLC interface through the PLC user program. Only here can several machine axes be assigned to one handwheel simultaneously.

- Using menu-assisted operation (HMI). Pressing the "Handwheel" softkey in the JOG-mode basic menu displays the "Handwheel" window. This enables an axis (WCS or MCS) to be assigned to each handwheel.

A separate user interface between the HMI and PLC is provided to allow activation of the handwheel from the operator panel (HMI). This interface that the basic PLC program supplies for handwheels 1 to 2 contains the following information:

- The axis numbers assigned to the handwheel IS "Axis number handwheel n" (VB1900 1003, ff)
- Additional information on the machine or geometry axis
IS "Machine axis" (VB1900 1003.7, ff)

The "Activate handwheel" interface signal is either set to "0" (disable) or to "1" (enable) by the PLC user program for the defined axis.

Settings as path or velocity

When the electronic handwheel is turned, the assigned axis is traversed either in the positive or negative direction depending on the direction of rotation.

The general MD11346 HANDWH_TRUE_DISTANCE (handwheel path or velocity specification) can be used to set the setting type of the handwheel motion and thus adapted to the intended use.

- MD value = 0 (default):

The settings from the handwheel are velocity specifications. When the handwheel is stationary, braking is realized along the shortest path.

- MD value = 1:

The settings from the handwheel are path specifications. No pulses are lost. Limiting the velocity to the maximum permissible value can cause the axes to overtravel. Particular care should be taken in the case of a high weighting of the handwheel pulses. Further variants of the path or speed setting are possible with the value = 2 or 3.

Evaluation

The traversing path/velocity produced by rotation of the handwheel is dependent on the following factors:

- Number of handwheel pulses received at the interface
- Active increment (machine function INC1, INC10, INC100, ...)
An increment is evaluated with 0.001 mm if the basic system setting is metric.
- Pulse weighting of the handwheel using general MD11320 HANDWH_IMP_PER_LATCH (handwheel pulses per locking position)

Motion command +/-

While the axis is moving, the "Travel command+" or "Travel command-" interface signal is transmitted to the PLC depending on the direction of motion.

- For geometry axes: V3300 1000.7/.6, V3300 1004.7/.6, V3300 1008.7/.6
- For machine axes / spindle: V390x 004.7/.6.

If the axis is already being moved using the traversing keys, the handwheel cannot be used. Alarm 20051 "Jogging with the handwheel not possible" is output.

Velocity

The velocity results from the pulses generated by the handwheel and the pulse evaluation: Traverse path per time unit. This velocity is limited by the value in the axis-specific MD32000 MAX_AX_VELO.

Abortion/interruption of traversing movement

The traversing movement is aborted as the result of a RESET or the axis-specific IS "Deletion of distance-to-go" (V380x 0002.2). The setpoint/actual-value difference is deleted.

NC STOP only interrupts the traversing movement. NC START releases the handwheel motion again.

Movement in the opposite direction

Depending on MD11310 HANDWH_REVERSE, the behavior when the traversing direction is reversed is as follows:

- MD value = 0:

If the handwheel is moved in the opposite direction, the resulting distance is computed and the calculated end point is approached as fast as possible: If this end point is located before the point where the moving axis can decelerate in the current direction of travel, the unit is decelerated and the end point is approached by moving in the opposite direction. If this is not the case, the newly calculated end point is approached immediately.

- MD value > 0:

If the handwheel is moved in the opposite direction by at least the number of pulses indicated in the machine data, the axis is decelerated as fast as possible and all pulses received until the end of interpolation are ignored. That means, another movement takes place only after standstill (setpoint side) of the axis (new function).

Response at software limit switches, working-area limitation

When axes are traversed in JOG mode, they can traverse only up to the first active limitation before the corresponding alarm is output.

Depending on the machine data MD11310 HANDWH_REVERSE, the behavior is as follows (as long as the axis on the setpoint side has not yet reached the end point):

- MD value = 0:

The distance resulting from the handwheel pulses forms a fictitious end point which is used for the subsequent calculations: If this fictitious end point is positioned, for example, 10 mm behind the limitation, these 10 mm must be traversed in the opposite direction before the axis traverses again. If a movement in the opposite direction is to be performed immediately after a limit, the fictitious distance-to-go can be deleted via IS "Delete distance-to-go" (V380x 0002.2) or by deselecting of the handwheel assignment.

- MD value > 0:

All handwheel pulses leading to an end point behind the limitation are ignored. Any movement of the handwheel in the opposite direction leads to an immediate movement in the opposite direction, i.e. away from the limitation.

8.5 Fixed-point approach in JOG

8.5.1 Introduction

Function

The machine user can use the "Approach fixed point in JOG" function to approach axis positions defined using machine data by actuating the traversing keys of the Machine Control Panel or by using the handwheel. The traveling axis comes to a standstill automatically on reaching the defined fixed point.

Applications

Typical applications are, for example:

- Approaching a basic position before starting an NC program.
- Travel towards tool change points, loading points and pallet change points.

Requirements

- The "Approaching fixed point in JOG" can be activated only in the JOG mode.
The function cannot be enabled in the JOG-REPOS and JOG-REF sub-modes and in JOG in the AUTOMATIC mode.
- The axis to be traversed must be referenced.
- A kinematic transformation may not be active.
- The axis to be traversed must not be a synchronized axis of an active coupling.
- No ASUPs are executed.

Approaching a fixed point with G75

The process for approaching defined fixed points can be activated from the part program too using the *G75* command.

For more information on approaching fixed points with *G75*, please refer to:

References:

Programming and Operating Manuals; Section: "Programming" -> "Axis movements" > "Fixed-point approach (G75)"

8.5.2 Functionality

Procedure

Procedure in "Approaching fixed point in JOG"

- Selection of JOG mode
- Enabling the "Approach fixed point in JOG" function
- Traversing of the machine axis with traverse keys or handwheel

Activation

The PLC sets the interface signal after the "Approach fixed point in JOG" function has been selected:

"JOG - Approach fixed point" (V380x 1001.0-2)

The number of the fixed point to be approached is output using bit 0 - 2 in binary code. The NC confirms activation with the following interface signal as soon as the function takes effect:

"JOG - Approaching fixed point active" (V390x 1001.0-2)

Sequence

The actual traversing is started with the traverse keys or the handwheel in the direction of the approaching fixed point.

The selected machine axis traverses till it comes to an automatic standstill at the fixed point.

The corresponding NC/PLC interface signal is sent on reaching the fixed point with "Exact stop fine":

"JOG - Approaching fixed point reached" (V390x 1001.3-5)

This display signal is also reported if the axis reaches the fixed point position in the machine coordinate system using other methods (e.g. NC program, synchronized action) at the setpoint end, and comes to a standstill at the actual-value end within the "Exact stop fine" tolerance window (MD36010 STOP_LIMIT_FINE)

Movement in the opposite direction

The response while traversing in the opposite direction (i.e. in the opposite direction to the one used when approaching the fixed point) depends on the setting of bit 2 in the following machine data:

MD10735 JOG_MODE_MASK (settings for JOG mode)

Traversing in the opposite direction is only possible if bit 2 is set.

Traversing in the opposite direction is blocked if bit 2 is not set, and the following channel status message is output if an attempt is made (using the traversing keys or the handwheel) to traverse in the opposite direction to the one used when approaching the fixed point:

"JOG: <Axis> direction blocked"

Approaching other fixed point

If a different fixed point is set during the fixed-point approach, the axis motion is stopped and the following alarm is signaled:

Alarm 17812 "Channel %1 axis %2 fixed-point approach in JOG: Fixed point changed"

The message signal "JOG - Approaching fixed point active" displays the number of the newly selected fixed point. The JOG traverse must be triggered again to continue traversing.



Tip

To avoid the alarm message, the machine user should proceed as follows:

1. Cancel the current traverse movement with residual distance deletion.
 2. Activate fixed point approach for another fixed point and start the operation after the axis comes to a standstill.
-

Withdrawal from fixed point / deactivation

To withdraw from a fixed position, you must deactivate the "Approaching fixed point in JOG" function. This is done by resetting the activation signal to "0".

V380x 1001.0-2 = 0

The message signals "JOG - Approaching fixed point active" and "JOG - Approaching fixed point reached" are canceled on leaving the fixed-point position.

Special case: Axis is already on fixed point

The axis cannot be moved if, while starting the fixed point traverse, the axis is already at the position of the fixed point to be approached. This is displayed by means of the following channel status message:

"JOG: <Axis> position reached"

To withdraw from the fixed position, you must deactivate the "Approaching fixed point in JOG" function.

Special features of incremental travel

If, during incremental travel, the fixed point is reached before the increment is completed, then the increment is considered to have been completed fully. This is the case even when only whole increments are traveled.

MD11346 HANDWH_TRUE_DISTANCE = 2 or 3

Features of modulo rotary axes

Modulo rotary axes can approach the fixed point in both directions (bit 2 of MD10735 has no significance for them). No attempt is made to follow the shortest path (DC) during the approach.

Features of spindles

A spindle changes to the positioning mode on actuating the "Approaching fixed point in JOG" function. The closed loop position control is active and the axis can traverse to the fixed point.

If a zero mark has not been detected, the following alarm message is output (as with axis operation):

Alarm 17810 "Channel %1 axis %2 not referenced"

As a spindle must also be a modulo rotary axis at all times, the same conditions apply for direction observation as for modulo rotary axes (refer to the paragraph "Features of modulo rotary axes")

8.5.3 Parameter setting

Movement in the opposite direction

The response while traversing in the opposite direction, i.e., against the direction of the approaching fixed point depends on the setting of Bit 2 in the machine data:

MD10735 JOG_MODE_MASK (settings for JOG mode)

| Bit | Value | Description |
|-----|-------|--|
| 2 | 0 | Travel in the opposite direction is not possible (default setting). |
| | 1 | Movement in the opposite direction is possible. |

Fixed point positions

A maximum of 4 fixed point positions can be defined for each axis via the following machine data:

MD30600 FIX_POINT_POS[n]

Number of valid fixed point positions

The number of valid fixed point positions of an axis is defined via the machine data:

MD30610 NUM_FIX_POINT_POS

Note

"Approaching fixed point with G75" constitutes an exception here. In this case, it is also possible to approach two fixed-point positions with one setting (MD30610 = 0).

8.5.4 Programming

System variables

The following system variables that can be read in the part program and in the synchronous actions for the "Approach fixed point" function.

| System variable | Description |
|----------------------------------|--|
| \$AA_FIX_POINT_SELECTED [<Axis>] | Number of fixed point to be approached |
| \$AA_FIX_POINT_ACT [<Axis>] | Number of the fixed point on which the axis is currently located |

8.5.5 Supplementary Conditions

Axis is indexing axis

The axis is not traversed and an alarm is output if the axis to be traversed is an indexing axis and the fixed point position to be approached does not match an indexing position.

Frames active

All active frames are ignored. Traversing is performed in the machine coordinate system.

Offset values active

Active compensation values (external work offset, synchronized action offset \$AA_OFF, online tool offset) are also applied. The fixed point is a position in the machine coordinates system.

An alarm is signaled if an offset movement (external work offset, synchronized action offset \$AA_OFF, online tool offset) is made during a fixed-point approach in JOG. The position of the fixed point to be approached in the machine coordinates system is not reached; instead a position that would have been reached without active offset movement is reached. The NC/PLC interface signal "JOG - Approaching fixed point reached" (V390x 1001.3-5) is not signaled.

Working-area limitations

Working-area limitations (in BCS and WCS) are considered and the axis motion is stopped on reaching the limits.

8.5.6 Application example

Target

A rotary axis (machine axis 4 [AX4]) is to be moved to Fixed Point 2 (90 degrees) with the "Approaching fixed point in JOG" function.

Parameter setting

The machine data for the "Approaching fixed point" function of machine axis 4 are parameterized as follows:

| | |
|------------------------------------|--|
| MD30610 NUM_FIX_POINT_POS[AX4] = 4 | 4 fixed points are defined for machine axis 4. |
| MD30600 FIX_POINT_POS[0,AX4] = 0 | 1st Fixed point of AX4 = 0 degree |
| MD30600 FIX_POINT_POS[1,AX4] = 90 | 2nd Fixed point of AX4 = 90 degree |
| MD30600 FIX_POINT_POS[2,AX4] = 180 | 3rd Fixed point of AX4 = 180 degree |
| MD30600 FIX_POINT_POS[3,AX4] = 270 | 4th Fixed point of AX4 = 270 degree |

Initial situation

Machine axis 4 is referred and is in Position 0 degree. This corresponds to the 1st fixed point and is output via the following NC/PLC interface signal:
V390x 1001.0 = 1 (bit 0 - 2 = 1)

Approaching fixed point 2

The control is switched in the JOG mode.

The "Approaching fixed point" procedure for fixed point 2 is activated via the following NC/PLC interface signal:
V380x 1002.1 = 1 (bit 0 - 2 = 2)

Activation is confirmed by the following NC/PLC interface signal:
V390x 1001.1 = 1 (bit 0 - 2 = 2)

The Plus traverse key in the machine control table is used to traverse continuously to approach Fixed Point 2.

The machine axis 4 stops at the 90 degree position. This is signaled via the following NC/PLC interface signal:
V390x 1001.4 = 1 (bit 3 - 5 = 2)

8.6 Data lists

8.6.1 Machine data

| Number | Identifier | Name |
|-----------------------|-----------------------------------|--|
| General information | | |
| 10000 | AXCONF_MACHAX_NAME_TAB[n] | Machine axis name [n = axis index] |
| 10735 | JOG_MODE_MASK | Settings for JOG mode |
| 11310 | HANDWH_REVERSE | Defines movement in the opposite direction |
| 11320 | HANDWH_IMP_PER_LATCH[0]...[2] | Handwheel pulses per locking position |
| 11346 | HANDWH_TRUE_DISTANCE | Handwheel path or velocity specification |
| Channel-specific | | |
| 20060 | AXCONF_GEOAX_NAME_TAB[n] | Geometry axis in channel [n = geometry axis index] |
| 20100 | DIAMETER_AX_DEF | Geometry axes with transverse axis functions |
| Axis/spindle-specific | | |
| 30600 | FIX_POINT_POS[n] | Fixed-point positions for the axis |
| 30610 | NUM_FIX_POINT_POS | Number of fixed-point positions for an axis |
| 32000 | MAX_AX_VELO | Maximum axis velocity |
| 32010 | JOG_VELO_RAPID | Rapid traverse in JOG mode |
| 32020 | JOG_VELO | JOG axis velocity |
| 32300 | MAX_AX_ACCEL | Axis acceleration |
| 32420 | JOG_AND_POS_JERK_ENABLE | Enable for axis-spec. jerk limitation |
| 32430 | JOG_AND_POS_MAX_JERK | Axis-specific jerk |
| 35130 | GEAR_STEP_MAX_VELO_LIMIT[0]...[5] | Maximum velocity for gear stage/spindle |

8.6.2 Setting data

| Number | Identifier | Name |
|---------------------|---------------------|--|
| General information | | |
| 41010 | JOG_VAR_INCR_SIZE | Size of variable increment for INC/handwheel |
| 41110 | JOG_SET_VELO | JOG velocity for linear axes |
| 41130 | JOG_ROT_AX_SET_VELO | JOG speed for rotary axes |
| 41200 | JOG_SPIND_SET_VELO | JOG velocity for the spindle |

8.6.3 Interface signals

| Number | Bit | Name |
|--|----------------------------------|---|
| Signals from HMI to PLC | | |
| V1900 1003 | .0 to .2 | Axis number for handwheel 1 |
| V1900 1004 | .0 to .2 | Axis number for handwheel 2 |
| NCK-specific | | |
| V2600 0001 | .0 | INC inputs in operating mode range active |
| Specific to operating mode | | |
| V3000 0000 | .2 | JOG mode |
| V3000 0002 | .0 to .6 | Machine function INC1 up to continuous in operating mode range |
| V3100 0000 | .2 | Active JOG mode |
| Channel-specific | | |
| V3200 1000 V3200 1004 V3200 1008 | .1, .0 .1, .0 .1, .0 | Activate handwheel (2, 1) for geometry axis 1 for geometry axis 2 for geometry axis 3 |
| V3200 1000 V3200 1004 V3200 1008 | .4 .4 .4 | Traversing-key lock for geometry axis 1 for geometry axis 2 for geometry axis 3 |
| V3200 1000 V3200 1004 V3200 1008 | .5 .5 .5 | Rapid traverse override for geometry axis 1 for geometry axis 2 for geometry axis 3 |
| V3200 1000 V3200 1004 V3200 1008 | .7 or .6 .7 or .6 .7 or .6 | Traversing keys plus or minus for geometry axis 1 for geometry axis 2 for geometry axis 3 |
| V3200 1000 V3200 1004 V3200 1008 | .0 to .6 .0 to .6 .0 to .6 | Machine function INC1 to continuous for geometry axis 1 for geometry axis 2 for geometry axis 3 |
| V3300 1000 V3300 1004 V3300 1008 | .1, .0 .1, .0 .1, .0 | Handwheel active (2, 1) for geometry axis 1 for geometry axis 2 for geometry axis 3 |
| V3300 1000 V3300 1004 V3300 1008 | .7 or .6 .7 or .6 .7 or .6 | Traverse command plus or minus for geometry axis 1 for geometry axis 2 for geometry axis 3 |
| V3300 1001 V3300 1005 V3300 1009 | .0 to .6 .0 to .6 .0 to .6 | Active machine function INC1 to continuous for geometry axis 1 for geometry axis 2 for geometry axis 3 |
| Axis/spindle-specific | | |
| VB380x 0000 | - | Feed override |
| V380x 0000 | .7 | Override active |
| V380x 0002 | .2 | Delete distance-to-go |
| V380x 0004 | .1, .0 | Activate handwheel (2, 1) |
| V380x 0004 | .4 | Traversing-key lock |
| V380x 0004 | .5 | Rapid traverse override |
| V380x 0004 | .7 or .6 | Traversing keys plus or minus |
| V380x 0005 | .0 to .6 | Machine function INC1 up to continuous in axis range |

8.6 Data lists

| Number | Bit | Name |
|------------|----------|--|
| V380x 1002 | .0 to .2 | Activated fixed-point approach in JOG (binary coded: fixed point 1 to 4) |
| V390x 0000 | .7/.6 | Position reached with coarse/fine exact stop |
| V390x 0004 | .1, .0 | Handwheel active (2, 1) |
| V390x 0004 | .7 or .6 | Traverse command plus or minus |
| V390x 0005 | .0 to .6 | Active machine function INC1 to continuous |
| V390x 1001 | .0 to .2 | Fixed-point approach in JOG active (binary coded) |
| V390x 1001 | .3 to .5 | Fixed point reached (binary coded) |

Auxiliary Function Outputs to PLC (H2)

9.1 Brief description

Auxiliary functions

For the purpose of workpiece machining operations, it is possible to program process-related functions (feedrate, spindle speed, gear stages) and functions for controlling additional devices on the machine tool (sleeve forward, gripper open, clamp chuck) in the part program in addition to axis positions and interpolation methods. This is performed with the "auxiliary functions" as collective term for various types.

The following types of auxiliary function are available:

- Miscellaneous function M
- Spindle function (S)
- Auxiliary function (H)
- Tool number T
- Tool offset D
- Feed F (for SINUMERIK 802D there is no output from F to PLC)

Output of auxiliary functions to PLC

The auxiliary function output sends information to the PLC indicating, for example, when the NC program needs the PLC to perform specific switching operations on the machine tool. The auxiliary functions are output, together with their parameters, to the PLC.

The values and signals must be processed by the PLC user program. The following section describes the various methods of configuring and programming auxiliary functions as well as their operating principles.

Auxiliary function groups

Auxiliary functions can be combined to form groups.

9.2 Programming of auxiliary functions

General structure of an auxiliary function

Letter[address extension]=Value

The letters which can be used for auxiliary functions are: **M, S, H, T, D, F.**

9.2 Programming of auxiliary functions

An address extension only exists for the H function. The address extension must be an integer. The square brackets can be omitted when an address extension is specified directly as a numeric value.

The value is defined differently for the individual auxiliary functions:

- INT= integer
- REAL= fractional decimal number (floating point)

Table 9- 1 Overview of auxiliary functions, programming

| Function | Address extension (integer) | | Value | | | Explanation | Number per block max |
|-----------------------|-----------------------------|--------|-------------------|------|-----------------------|--|-------------------------|
| | Meaning | Area | Area | Type | Meaning | | |
| M | Spindle no. | 1 - 2 | 0-99 | INT | Function | Specific numbers are assigned a fixed function. | 5 |
| Switchgear protection | Spindle no. | 1 - 2 | 0-±3.4028 ex 38 | REAL | Spindle speed | | 1 |
| H | Any | 0 - 99 | ±3.4028 ex 38 | REAL | Any | Functions have no effect in the NCK; only to be implemented on the PLC | 3 |
| D | - | - | 0-32000 | INT | Tool selection | | 1 |
| D | - | - | 0-9 | INT | Tool offset selection | D0 deselection, default D1 | 1 |
| F | - | - | 0,001-999 999,999 | REAL | Path feedrate | | 1 |

A maximum total of 10 auxiliary functions may be programmed in one block. Alarm 14770 "Auxiliary function incorrectly programmed" is output when the specified length for address extension of value is exceeded or when the wrong data type is used. The following table shows some programming examples for H functions.

If the admissible number of auxiliary functions per block is exceeded, alarm 12010 is issued.

Table 9- 2 Programming examples of H functions

| Programming | Output of H function to the PLC |
|-------------|---------------------------------|
| H5 | H0=5.0 |
| H=5.379 | H0=5.379 |
| H17=3.5 | H17=3.5 |
| H5.3=21 | Error, alarm 14770 |

Block change

A new auxiliary function output from the NCK to the PLC is only possible after the PLC has acknowledged all transferred auxiliary functions. Auxiliary functions are present in the user interface for at least one PLC cycle. A block is considered as completed when the programmed movement has been completed and the auxiliary function has been acknowledged. To do so, the NCK stops the part program processing if necessary to ensure that no auxiliary functions are lost from the PLC user program's point of view.

9.3 Transfer of values and signals to the PLC interface

Time of transfer

In the case of auxiliary functions which are output at the end of a block (e.g. M2), the output is only made after all axis movements and the SPOS movement of the spindle have been completed.

If several auxiliary functions with different output types (prior, during, at end of motion) are programmed in one motion block, then they are output individually according to their output type.

In a block without axis movements or SPOS movement of the spindle, the auxiliary functions are all output immediately in a block.

Continuous-path mode

A path movement can only remain continuous if auxiliary function output takes place **during the movement** and is acknowledged by the PLC before the path end is reached, see Chapter "Continuous Path Mode (B1)".

Interface signals

Transfer of the signals from NCK to the PLC:

Reference:

/LIS/ SINUMERIK 802D sl lists

9.4 Grouping of auxiliary functions

Functionality

The auxiliary functions of the types M, H, D, T, and S that are to be issued can be grouped to auxiliary function groups by means of machine data.

An auxiliary function can only be assigned to one group.

Only one auxiliary function of a group can be programmed per block. Otherwise, alarm 14760 is issued.

Configuration

You can define a maximum of 64 auxiliary function groups. A maximum of 64 auxiliary functions can be assigned to these 64 auxiliary function groups. This number does not include auxiliary functions (group 1 to 3) that are pre-assigned as standard.

The actual number of auxiliary functions that are to be assigned must be entered in the NCK-specific MD11100 AUXFU_MAXNUM_GROUP_ASSIGN (number of the auxiliary functions distributed to the AUXFU groups). To do so, the password for protection level 2 must be set. Then, the control must be turned off and on again. Now, the subsequent machine data with an index n greater than zero are available and additional values can be entered.

An allocated auxiliary function is defined in the following machine data:

- MD22000 AUXFU_ASSIGN_GROUP[n] (auxiliary function group)
- MD22010 AUXFU_ASSIGN_TYPE[n] (auxiliary function type)
- MD22020 AUXFU_ASSIGN_EXTENSION[n] (auxiliary function extension)
- MD22030 AUXFU_ASSIGN_VALUE[n] (auxiliary function value)

Predefined auxiliary function groups

Group 1:

The auxiliary functions M0, M1, and M2 (M17, M30) are, by default, allocated to group 1. The output is always made at the end of the block.

Group 2:

The M functions M3, M4, and M5 (M70) are, by default, allocated to group 2. The output is always made before the movement.

Group 3:

The S function is, by default, contained in group 3. The output is made with the movement.

User-defined groups

The other (user-defined) groups are issued with the movement. A different setting of the outputs can only be set with the "Expert mode" access (protection level 1).

Ungrouped auxiliary functions

The output of auxiliary functions that are not assigned to groups is made with the movement.

Configuring example:

Distribute 8 auxiliary functions to 7 groups:

Group 1: M0, M1, M2 (M17, M30) - by default, should be kept

Group 2: M3, M4, M5 (M70) - by default, should be kept

Group 3: S functions - by default, should be kept

Group 4: M78, M79

Group 5: M80, M81

Group 6: H1=10, H1=11, H1=12

Group 7: all T functions

Password for protection level 2 is set.

Make entry in MD11100 AUXFU_MAXNUM_GROUP_ASSIGN=8.

Then, turn off the control and turn it on again or perform the control start-up through the softkey and define the remaining machine data with a subsequent restart of the control.

Table 9- 3 Entries into the machine data for the example

| Index n | MD22000 (GROUP) | MD22010 (TYPE) | MD22020 (EXTENSION) | MD22030 (VALUE) |
|---------|--------------------|-------------------|------------------------|--------------------|
| 0 | 4 | M | 0 | 78 |
| 1 | 4 | M | 0 | 79 |
| 2 | 5 | M | 0 | 80 |
| 3 | 5 | M | 0 | 81 |
| 4 | 6 | H | 1 | 10 |
| 5 | 6 | H | 1 | 11 |
| 6 | 6 | H | 1 | 12 |
| 7 | 7 | D | 0 | -1 |

9.5 Block-search response

Block search with calculation

For the block search with calculation all auxiliary functions that are assigned to a group are collected and are issued at the end of the block search before the actual re-entry block (except for group 1: M0, M1,...). The last auxiliary function of a group is issued.

All collected auxiliary functions are issued in a separate block as regular auxiliary functions and before the movement.

NOTICE

If the auxiliary functions are to be collected during the block search, they must be assigned to an auxiliary function group!

9.6 Description of the auxiliary functions

9.6.1 M function

Application

You can use the M functions to enable the various switching operations on the machine per part program.

Scope of functions

- Five M functions per part program block are possible.
- Value range of M functions: 0 to 99; integer number
- Permanent functions have already been assigned to some of the M functions by the control manufacturer (see "Operation and Programming" in the user manual). The functions not yet assigned fixed functions are reserved for free use of the machine manufacturer.

9.6.2 T function

Application

The T function can be used to make the tool required for a machining operation available through the PLC. Whether a tool change is to be performed directly with the T command or with a subsequent M6 command can be set in MD22550 TOOL_CHANGE_MODE.

The programmed T function can be interpreted as tool number or as location number.

Scope of functions

One T function per part program block is possible.

Peculiarity

T0 is reserved for the following function: remove the current tool from the tool holder without loading a new tool.

9.6.3 D function

The D function is used to select the tool offset for the active tool. Tool offsets are described in detail under:

Reference:

/BP_/ Operation and Programming

9.6.4 H function

Application

The H functions can be used to transfer different values from the part program to the PLC. The meaning can be chosen by the user.

Scope of functions

- Three H functions per part program block are possible.
- Value range of the H functions: Floating data (as calculating parameter R)
- Address extension 0 to 99 (H0=... to H99=...) possible

9.6.5 S function

The S function is used to determine the speed for the spindle with M3 or M4. For turning machines with G96 (constant cutting speed) the cutting value is specified.

Reference:

/BPD/ Operation and Programming

9.7 Data lists

9.7.1 Machine data

| Number | Identifier | Name |
|------------------|---------------------------|--|
| General | | |
| 11100 | AUXFU_MAXNUM_GROUP_ASSIGN | Number of auxiliary functions distributed among the AUXFU groups |
| Channel-specific | | |
| 22000 | AUXFU_ASSIGN_GROUP[n] | Auxiliary function group |
| 22010 | AUXFU_ASSIGN_TYPE[n] | Auxiliary function type |
| 22020 | AUXFU_ASSIGN_EXTENSION[n] | Auxiliary function extension |
| 22030 | AUXFU_ASSIGN_VALUE[n] | Auxiliary function value |

9.7.2 Interface signals

| Number | Bit | Name |
|------------------|----------|--|
| Channel-specific | | |
| V2500 0000 | .0 to .4 | M function 1 change to M function 5 change |
| V2500 0006 | .0 | S function 1 change |
| V2500 0008 | .0 | T function 1 change |
| V2500 0010 | .0 | D function 1 change |
| V2500 0012 | .0 to .2 | H function 1 change to H function 3 change |
| VD2500 2000 | | T function 1 (DINT) |
| VD2500 3000 | | M function 1 (DINT) |
| VB2500 3004 | | Extended address of M function 1 (BYTE) |
| VD2500 3008 | | M function 2 (DINT) |
| VB2500 3012 | | Extended address of M function 2 (BYTE) |
| VD2500 3016 | | M function 3 (DINT) |
| VB2500 3020 | | Extended address of M function 3 (BYTE) |
| VD2500 3024 | | M function 4 (DINT) |
| VB2500 3028 | | Extended address of M function 4 (BYTE) |
| VD2500 3032 | | M function 5 (DINT) |
| VB2500 3036 | | Extended address of M function 5 (BYTE) |
| VD2500 4000 | | S function 1 (REAL format) |
| VB2500 4004 | | Extended address of S function 1 (BYTE) |
| VD2500 4008 | | S function 2 (REAL format) |
| VB2500 4012 | | Extended address of S function 2 (BYTE) |
| VD2500 5000 | | D function 1 (DINT) |
| VW2500 6004 | | Extended address of H function 1 (Word) |
| VD2500 6000 | | H function 1 (REAL format) |
| VW2500 6012 | | Extended address of H function 2 (Word) |
| VD2500 6008 | | H function 2 (REAL format) |
| VW2500 6020 | | Extended address of H function 3 (Word) |
| VD2500 6016 | | H function 3 (REAL format) |
| V2500 1000 | .0 - .7 | Decoded M signals: M00 - M07 |
| V2500 1001 | .0 - .7 | Decoded M signals: M08 - M15 |
| to | | |
| V2500 1012 | .0 - .7 | Decoded M signals: M96 - M99 |
| VD370x 0000 | - | M function for the spindle (DINT), axis-specific |
| VD370x 0004 | - | S function for the spindle (REAL), axis-specific |

Operating Modes, Program Operation (K1)

10.1 Brief description

Program operation

The execution of part programs or part program blocks in AUTOMATIC or MDA modes is referred to as program operation. During execution, the program sequence can be controlled by PLC interface signals and commands.

Channel

A channel constitutes a unit in which a part program can be executed.

A channel is assigned an interpolator with program processing by the system. A certain mode is valid for it.

The SINUMERIK 802D sl control has one channel.

10.2 Operating modes

Activating

The required operating mode is activated by the interface signals in the VB 3000 0000. If several modes are selected at the same time the priority of the operating modes is as follows:

- **JOG** (high priority): The axes can be traversed manually with the handwheel or the traversing keys. Channel-specific signals and interlocks are not observed.
- **MDA**: Program blocks can be processed
- **AUTOMATIC** (lower priority): Automatic processing of part programs

Feedback signal

The active operating mode is displayed by the interface signals in the VB3100 0000.

Possible machine functions in JOG

The following machine function can be selected in the JOG operating mode:
REF (reference point approach)

The required machine function is activated with IS "REF" (V3000 0001.2). The display is visible in the IS "active machine function REF" (V3100 0001.2).

Possible machine functions in MDA

The following machine function can be selected in the MDA operating mode:
TEACH IN (insert program blocks)

The required machine function is activated with IS "TEACH IN" (V3000 0001.0). The display is visible in the IS "Active machine function TEACH IN" (V3100 0001.0).

Stop

A stop signal can be issued with the following interface signals

- IS "NC stop" (V3200 0007.3)
- IS "NC stop axes plus spindles" (V3200 0007.4)
- IS "NC stop at block limit" (V3200 0007.2)

Depending on the interface signal used, either only the axes or in addition the spindles of the channels are stopped or the axes at block end.

RESET

The active part program is aborted by the IS "Reset" (V3000 0000.7).

The following actions are executed when the IS "Reset" is triggered:

- Part program preparation is stopped immediately.
- Axes and spindles are stopped.
- Any auxiliary functions of the current block not yet output, are no longer output.
- The block indicator is reset to the beginning of the relevant part program.
- All Reset alarms are deleted from the display.
- The reset is complete as soon as IS "Channel status Reset" (V3300 0003.7) is set.

Ready

Ready to run is displayed by IS "802 Ready" (V3100 0000.3).

10.2.1 Mode change

General

A changeover to another operating mode is requested and activated via the interface.

Note

The mode is not changed internally until the IS "Channel status active" (V3300 0003.5) is **no longer** present.

In the "Channel status Reset" (IS: V3300 0003.7, e.g. after pressing the "Reset key") one can switch from any operating mode into another.

In the "Channel status interrupted" (IS: V3300 0003.6) only a conditional changeover is possible (see following table).

If one leaves AUTO to change to JOG, one must return to AUTO again or press "Reset". Thus a change AUTO-JOG-MDA is made impossible. The same applies for MDA from which one may change neither directly nor indirectly to AUTO, provided the Reset state is present.

The table shows the possible operating mode changes depending on the current operating mode and the channel state ("Channel in reset" or "Channel interrupted").

Table 10- 1 Operating mode change depending on channel state

| | From | AUTOMATIC | | JOG | | | MDI | |
|------------------|------|-----------|-----------|-------|-----------------|----------------|-------|-----------|
| | | | | | AUTO previously | MDI previously | | |
| To | | Reset | Interrupt | Reset | Interrupt | Interrupt | Reset | Interrupt |
| AUTOMATIC | | | | X | X | | X | |
| JOG | | X | X | | | | X | X |
| MDI | | X | | X | | X | | |

Possible mode changes are shown by an "X".

Error on operating mode changeover

A corresponding error message is output if a mode change request is rejected by the system. This error message can be cleared without changing the channel status.

Mode change disable

Changeover between operating modes can be inhibited by means of IS "Mode group changeover disable" (V3000 000.4). This suppresses the mode change request.

10.2.2 Functional possibilities in the individual modes

Overview of the functions

You see from the following table which function can be selected in which operating mode and in which operating state.

Table 10- 2 Functional possibilities in the individual modes

| Mode of operation | AUTOMATIC | | | JOG | | | | | MDI | | | | | |
|---|-----------|----|----|-----|---|----|---|----|-----|----|----|----|---|---|
| | 1 | 2 | 3 | 1 | 3 | 4 | 3 | 5 | 3 | 1 | 2 | 3 | 6 | 7 |
| Loading a part program from outside through "Services" | sb | sb | | sb | | sb | | sb | sb | sb | sb | | | |
| Processing a part program/block | s | s | b | | | | | | | s | s | b | | |
| Block search | s | s | b | | | | | | | | | | | |
| Reference point approach via part program command (G74) | | | sb | | | | | | | | | sb | | |

s: Function can be started in this status
 b: Function can be processed in this status
 1: Channel in reset
 2: Channel interrupted
 3: Channel active
 4: Channel interrupted JOG during AUTOMATIC interruption
 5: Channel interrupted JOG during MDA interruption
 6: Channel active JOG in MDA during MDA interruption
 7: Channel active JOG in MDA

10.2.3 Monitoring functions in the individual modes

Overview of monitoring functions

Different monitoring functions are active in individual operating modes.

Table 10- 3 Monitoring functions and interlocks

| Mode of operation | AUTO | | | JOG | | | | | MDI | | | | | |
|--|------|---|---|-----|---|---|---|---|-----|---|---|---|---|---|
| | 1 | 2 | 3 | 1 | 3 | 4 | 3 | 5 | 3 | 1 | 2 | 3 | 6 | 7 |
| Axis-specific monitoring functions or when positioning the spindle | | | | | | | | | | | | | | |
| SW limit switch + | | | x | | x | | x | | x | | | x | x | x |
| SW limit switch – | | | x | | x | | x | | x | | | x | x | x |
| HW limit switch + | x | x | x | x | x | x | x | x | x | x | x | x | x | x |
| HW limit switch – | x | x | x | x | x | x | x | x | x | x | x | x | x | x |
| Exact stop coarse/fine | x | x | x | x | x | x | x | x | x | x | x | x | x | x |
| Clamping tolerance | x | x | x | x | x | x | x | x | x | x | x | x | x | x |
| DAC limit (analog spindle) | x | x | x | x | x | x | x | x | x | x | x | x | x | x |
| Contour monitoring | | | x | | x | | x | | x | | | x | x | x |

| Spindle-specific monitoring functions | | | | | | | | | | | | | | |
|---------------------------------------|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Speed limit exceeded | | | x | | x | | x | | x | | | x | | x |
| Spindle is stationary | x | x | x | x | x | x | x | x | x | x | x | x | x | x |
| Spindle synchronized | | | x | | x | | x | | x | | | x | | x |
| Speed in setpoint range | | | x | | | | | | | | | | | |
| Maximum permissible speed | | | x | | x | | x | | x | | | x | | x |
| Encoder frequency limit | | | x | | x | | x | | x | | | x | | x |

x: Monitoring is active in this status

- 1: Channel in reset
- 2: Channel interrupted
- 3: Channel active
- 4: Channel interrupted JOG during AUTOMATIC interruption
- 5: Channel interrupted JOG during MDA interruption
- 6: Channel active JOG in MDA during MDA interruption
- 7: Channel active JOG in MDA

10.2.4 Interlocks in the individual modes

Overview of interlocks

Different interlocks can be active in the different operating modes.

The following table shows which interlocks can be activated in which operating mode and in which operating state.

| Mode of operation | AUTO | | | JOG | | | | | MDI | | | | | |
|--|------|---|---|-----|---|---|---|---|-----|---|---|---|---|---|
| Functions | 1 | 2 | 3 | 1 | 3 | 4 | 3 | 5 | 3 | 1 | 2 | 3 | 6 | 7 |
| General interlocks | | | | | | | | | | | | | | |
| 802 Ready | x | x | x | x | x | x | x | x | x | x | x | x | x | x |
| Mode change disable | x | x | x | x | x | x | x | x | x | x | x | x | x | x |
| Channel-specific interlocks | | | | | | | | | | | | | | |
| Feed stop | | | x | | x | | x | | x | | | x | x | x |
| NC Start disable | x | x | x | x | x | x | x | x | x | x | x | x | x | x |
| Read-in disable | x | x | x | x | x | x | x | x | x | x | x | x | x | x |
| Axis-specific interlocks | | | | | | | | | | | | | | |
| Spindle disable | x | x | x | x | x | x | x | x | x | x | x | x | x | x |
| Controller disable | x | x | x | x | x | x | x | x | x | x | x | x | x | x |
| Axis disable | x | x | x | x | x | x | x | x | x | x | x | x | x | x |
| Spindle-specific interlocks | | | | | | | | | | | | | | |
| Controller disable | x | x | x | x | x | x | x | x | x | x | x | x | x | x |
| Spindle disable | x | x | x | x | x | x | x | x | x | x | x | x | x | x |
| x: Interlock can be activated in this status 1: Channel in reset 2: Channel interrupted 3: Channel active 4: Channel interrupted JOG during AUTOMATIC interruption 5: Channel interrupted JOG during MDA interruption 6: Channel active JOG in MDA during MDA interruption 7: Channel active JOG in MDA | | | | | | | | | | | | | | |

10.3 Processing a part program

10.3.1 Program mode and part program selection

Definition

Program mode applies if a part program is processed in the AUTOMATIC mode or program blocks are processed in the MDA mode.

Channel control

The Program mode can be controlled even while being executed via interface signals from the PLC. These can be either mode groupspecific or channelspecific interface signals.

The channel reports its current program operation status to the PLC with interface signals.

Selection

A part program can be selected only if the relevant channel is in the Reset state.

The part program can be selected via:

- operator input (Machine operating area) / Program manager
- the PLC
 - Selection of a program via the program number in "Program list" (see Manual "Operation and Programming...", Chapter "System")
 - Reselection of an active program via the PLC-HMI interface (see Section "Signals from HMI to PLC (A2)")

10.3.2 Start of part program or part program block

START command, channel status

The channel-specific IS "NC start" (V3200 0007.1), which is usually controlled from the machine control panel key "NC Start", starts program processing.

The START command can only be executed in AUTOMATIC and MDA modes. For this purpose, the channel must be in the "Channel status reset" (V3300 0003.7) or "Channel status interrupted" (V3300 0003.6).

Required signal states

The selected part program can now be enabled for processing with the START command. The following enable signals are relevant:

| | |
|---|--------------------|
| IS "802 Ready" (V3100 0000.3) | must be set |
| IS "Activate program test" (V3200 0001.7) | may not be set |
| IS "NC Start disable" (V3200 0007.0) | may not be set |
| IS "NC Stop at block limit" (V3200 0007.2) | may not be set |
| IS "NC stop" (V3200 0007.3) | may not be set |
| IS "NC Stop axes plus spindle" (V3200 0007.4) | may not be set |
| IS "EMERGENCY STOP" (V2700 0000.1) | may not be set |
| Axis or NCK alarm | may not be present |

Execution of command

The part program or part program block is automatically processed and IS "Channel status active" (V3300 0003.5) and IS "Program status running" (V3300 0003.0) are set.

The program is processed until the end of the program has been reached or the channel is interrupted or aborted by a STOP or RESET command.

Interrupts

The START command is not effective if the prerequisite is not fulfilled. Then one of the following interrupts occurs: 10200, 10202, 10203

10.3.3 Part program interruption

Channel status

The STOP command is executed only if the channel concerned is in the "Channel active" status (V3300 0003.5).

STOP commands

There are various commands which stop processing of the program and set the channel status to "interrupted":

- IS "NC Stop at block limit" (V3200 0007.2)
- IS "NC stop" (V3200 0007.3)
- IS "NC Stop axes plus spindle" (V3200 0007.4)
- IS "Single block" (V3200 0000.4)
- Programming command "M0" or "M1" and corresponding activation

Execution of command

After execution of the STOP command, IS "Program status stopped" (V3300 0003.2) and the IS "Channel status interrupted" (V3300 0003.6) are set. Processing of the interrupted part program can continue from the point of interruption with another START command.

The following actions are executed when the STOP command is triggered:

- Part program processing is stopped at the next block limit (with NC stop at block limit, M0/M1 or single block), processing is stopped immediately with the other STOP commands.
- Any auxiliary functions of the current block not yet output, are no longer output.
- The axes are stopped with subsequent stop of the part program processing.
- The block indicator stops at the point of interruption.

10.3.4 RESET command

Function

The RESET command (IS "Reset" (V3000 000.7)) can be executed in every channel state. This command is aborted by another command.

A RESET command can be used to interrupt an active part program or part program blocks. After execution of the Reset command, IS "Channel status reset" (V3300 0003.7) and the IS "Program status aborted" (V3300 0003.4) are set.

The part program cannot be continued at the point of interruption. All axes in the channel are at exact stop.

The following actions are executed when the RESET command is triggered:

- Part program preparation is stopped immediately.
- All axes and if appropriate spindles are braked.
- Any auxiliary functions of the current block not yet output, are no longer output.
- The block indicator is reset to the beginning of the part program.
- All alarms are cleared from the display if they are not POWER ON alarms.

10.3.5 Program control

Selection/activation

The user can control part program processing via the user interface. Under the "Program control" menu (operating mode AUTOMATIC, operating area "Position") certain functions can be selected, whereby some functions act on interface signals of the PLC. These signals are merely selection signals from the user interface. They do not activate the selected function.

These signal states must be transferred from the PLC user program to another area of the data block to activate the selected functions. With program control by the PLC the signals are to be set directly.

Table 10- 4 Program control

| Function | Selection signal | Activation signal | Checkback signal |
|---|-------------------------|------------------------|------------------|
| SKP skip block | V1700 0001.0 | V3200 0002.0 | |
| DRY dry run feedrate | V1700 0000.6 | V3200 0000.6 | |
| ROV rapid traverse override | V1700 0001.3 | V3200 0006.6 | |
| Preselection: SBL -single block coarse SBL -single block fine Single block | - - User-specific | - - V3200 0000.4 | |
| M1 programmed stop | V1700 0000.5 | V3200 0000.5 | V3300 0000.5 |
| PRT program test | V1700 0000.7 | V3200 0001.7 | V3300 0001.7 |

10.3.6 Program status

Program states

The status of the selected program is displayed in the interface in the AUTOMATIC and MDA operating modes. If the JOG operating mode is selected when the program is stopped, then the "interrupted" program status is displayed there or on reset also "aborted".

The following program states are available in SINUMERIK 802D:

- IS "Program status aborted" (V3300 0003.4)
- IS "Program status interrupted" (V3300 0003.3)
- IS "Program status stopped" (V3300 0003.2)
- IS "Program status running" (V3300 0003.0)

The effect of commands/signals

The program status can be controlled by activating different commands or interface signals. The following table shows the resulting program state when these signals are set (status before the signal is set -> Program status running).

Table 10- 5 Effect on program status

| Commands | Program execution states | | | |
|--|--------------------------|-------------|---------|---------|
| | Aborted | Interrupted | Stopped | Running |
| IS "Reset" | X | | | |
| IS "NC Stop" | | | X | |
| IS "NC stop at block limit" | | | X | |
| IS "NC stop axes and spindles" | | | X | |
| IS "Read-in disable" | | | | X |
| IS "Feed stop, channel-sp." | | | | X |
| IS "Feed stop, axis-sp." | | | | X |
| Feed override = 0% | | | | X |
| IS "Spindle stop" | | | | X |
| M2 in the block | X | | | |
| M0/M1 in the block | | | X | |
| IS "Single block" | | | X | |
| Auxiliary functions output to PLC but not yet acknowledged | | | X | |

10.3.7 Channel status

Channel states

The current channel status is signaled at the interface for the channel. The PLC can then trigger certain responses and interlocks configured by the manufacturer depending on the status at the interface. The channel status is displayed in all operating modes.

The following channel states are available:

- IS "Channel status reset" (V3300 0003.7)
- IS "Channel status interrupted" (V3300 0003.6)
- IS "Channel status active" (V3300 0003.5)

The effect of commands/signals

The channel status can be modified through the activation of various commands or interface signals. The following table shows the resulting channel status when these signals are set (assumed status before the signal is set - > Channel status active).

The "Channel status active" signal is obtained when a part program or part program block is being executed or when the axes are traversed in JOG mode.

Table 10- 6 Effect on channel status

| Commands | Resulting channel status | | |
|--|--------------------------|-------------|--------|
| | Reset | Interrupted | Active |
| IS "Reset" | X | | |
| IS "NC Stop" | | X | |
| IS "NC stop at block limit" | | X | |
| IS "NC stop axes and spindles" | | X | |
| IS "Read-in disable" | | | X |
| IS "Feed stop, channel-sp." | | | X |
| IS "Feed stop, axis-sp." | | | X |
| Feed override = 0 % | | | |
| IS "Spindle stop" | | | X |
| M2 in the block | X | | |
| M0/M1 in the block | | X | |
| IS "Single block" | | X | |
| Auxiliary functions output to PLC but not yet acknowledged | | | X |

10.3.8 Eventdriven program calls

Application

In the case of certain events, an implied user program is to start. This allows the user to activate the initial settings of functions or carry out initialization routines by part program command.

Event selection

MD20108 PROG_EVENT_MASK (event-driven program call) can be used to specify which of the following events is to enable the user program:

- Bit0 = 1: Part program start
- Bit1 = 1: Part program end
- Bit2 = 1: Operator panel reset
- Bit3 = 1: Power up (of the NC control)

The user program is stored per default under the path name `/_N_CMA_DIR/_N_PROG_EVENT_SPF`. Another program name can be specified in MD11620 PROG_EVENT_NAME.

Other program name

A name is specified in MD11620 PROG_EVENT_NAME (program name of PROG_EVENT). The following directories are searched for the user program in the specified sequence:

- `/_N_CUS_DIR/` for user cycles
- `/_N_CMA_DIR/` for manufacturer cycles

The first program found with the stored name is called when a configured event occurs.

The same protection mechanisms that can be activated for cycles (protection levels for writing, reading etc.) are activated.

MD20108 PROG_EVENT_MASK is ignored during the simulation.

Request which start event

In the user program, the system variable `$P_PROG_EVENT` can be used to request the event, which enabled the part program.

Event

Part program start

Table 10- 7 Sequence when starting a part program

| Sequence | Command | Boundary conditions (must be satisfied before the command) | Comments |
|----------|--|--|---|
| 1 | Channel selection: Reset status Operating mode selection: AUTO or AUTO and overstringing or MDA or TEACH IN | None | Select channel and mode |
| 2 | NC Start | None | NCK start |
| 3 | MD20112 START_MODE_MASK | Initialization sequence with evaluation | |
| 4 | /_N_CMA_DIR/_N_PROG_EVENT_SP F or name from MD11620 | as a subroutine | Implied call of the path name as a subroutine |
| 5 | | None | Processing of the data part of the main program |
| 6 | | None | Processing of the program part of the main program |

Event

Part program end

Table 10- 8 Sequence at part program end

| Sequence | Command | Boundary conditions (must be satisfied before the command) | Comments |
|----------|--|--|---|
| 1 | Channel selection: Reset status Operating mode selection: AUTO or AUTO and overstringing or MDA or TEACH IN | None | Select channel and mode |
| 2 | NC Start | Block with end of part program | Block is changed |
| 3 | MD20110 RESET_MODE_MASK, MD20150 GCODE_RESET_VALUES, MD20152 GCODE_RESET_MODE | Control activated: Reset sequence with evaluation | |
| 4 | /_N_CMA_DIR/_N_PROG_EVENT_SP F or name from MD11620 | as an ASUB | Implied call of the path name as an ASUB |
| 5 | MD20110 RESET_MODE_MASK, MD20150 GCODE_RESET_VALUES, MD20152 GCODE_RESET_MODE | Control activated: Reset sequence with evaluation | The G code reset position continues to be specified with machine data |

Event

Operator panel reset

Table 10- 9 Processing sequence in operator panel reset

| Sequence | Command | Boundary conditions (must be satisfied before the command) | Comments |
|----------|---|--|---|
| 1 | Selection of channel and mode: any | Initial state: Any mode, any channel status | Select mode / channel status from any state |
| 2 | Reset | | |
| 3 | MD20110 RESET_MODE_MASK, MD20150 GCODE_RESET_VALUES, MD20152 GCODE_RESET_MODE | Control activated: Reset sequence with evaluation | |
| 4 | /_N_CMA_DIR/_N_PROG_EVENT_SP F or name from MD11620 | as an ASUB | Implied call of the path name as an ASUB |
| 5 | MD20110 RESET_MODE_MASK, MD20150 GCODE_RESET_VALUES, MD20152 GCODE_RESET_MODE | Control activated: Reset sequence with evaluation | The G code reset position continues to be specified with machine data |

Event

Startup

Table 10- 10 Sequence with Powerup

| Sequence | Command | Boundary conditions (must be satisfied before the command) | Comments |
|----------|---|---|---|
| 1 | Reset | after power up | |
| 2 | MD20110 RESET_MODE_MASK, MD20150 GCODE_RESET_VALUES, MD20152 GCODE_RESET_MODE | Control activated after ramp up: Reset sequence with evaluation | |
| 3 | /_N_CMA_DIR/_N_PROG_EVENT_SP F or name from MD11620 | as an ASUB | Implied call of the path name as an ASUB |
| 4 | MD20110 RESET_MODE_MASK, MD20150 GCODE_RESET_VALUES, MD20152 GCODE_RESET_MODE | Control activated: Reset sequence with evaluation | The G code reset position continues to be specified with machine data |

Chronological sequences

For part program start and part program end:

Time sequence of VDI signals VB3300 0003 ("Program status" and "Channel status") when processing a part program with an event-driven program call for part program start and part program end:

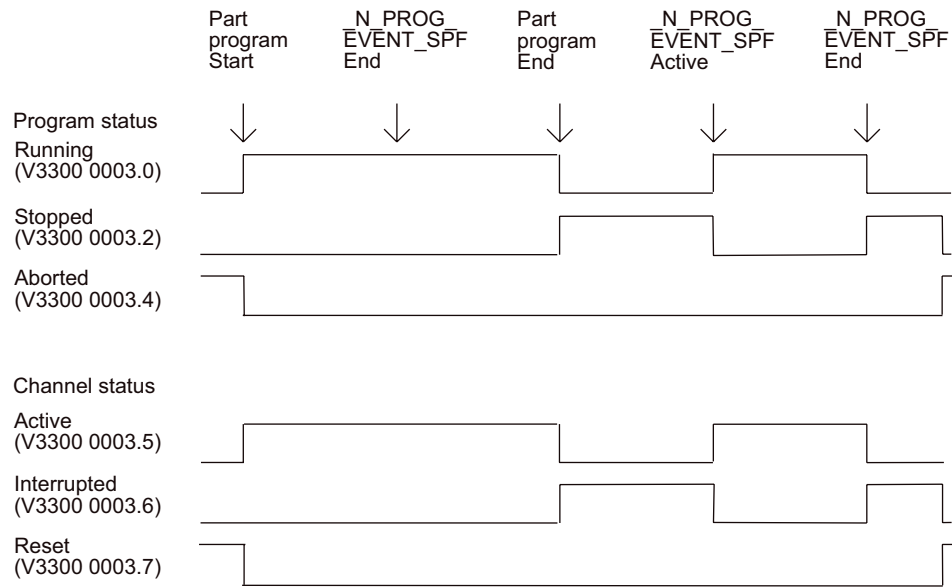


Figure 10-1 Time sequence of the interface signals for program status and channel status (1)

With operator panel reset:

Time sequence of VDI signals VB3300 0003 ("Program status" and "Channel status") when processing with an event-driven program call:

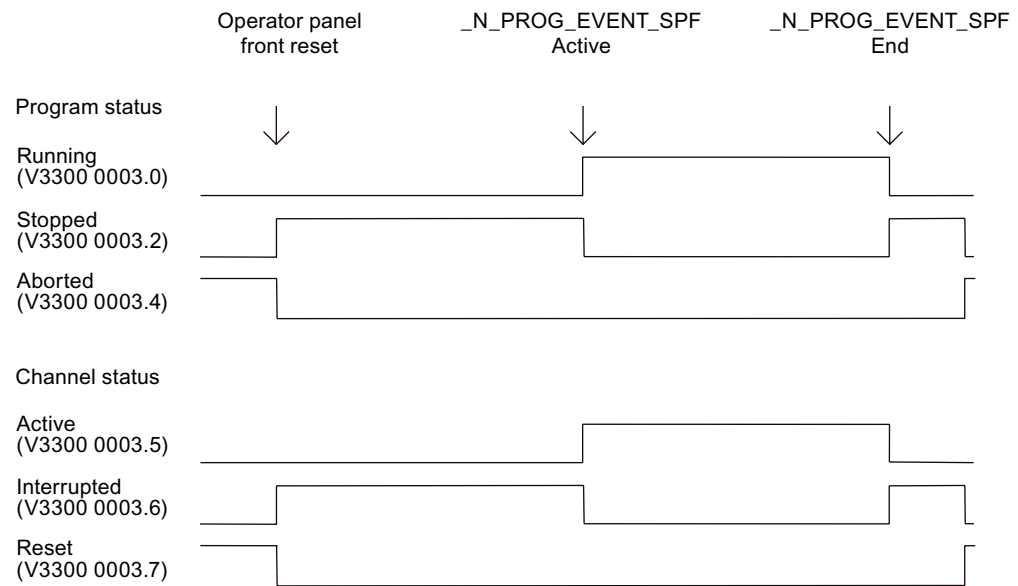


Figure 10-2 Time sequence of the interface signals for program status and channel status (2)

Note

IS V3300 0003.4 ("Program status aborted") and V3300 0003.7 ("Channel status reset") are only received if `_N_PROG_EVENT_SPF` has been completed.

Neither IS V3300 0003.4 ("Program status aborted") nor V3300 0003.7 ("Channel status reset") are received between the program end and the start of the program event.

This is also the case between an operator panel reset and the start of the program event.

Special points to be noted

The following must be noted for user program `_N_PROG_EVENT_SPF`:

- It is run with the lowest priority and can, therefore, be interrupted by the user ASUB.
- The PLC can be advised of the processing status of `_N_PROG_EVENT_SPF` via user M functions.
- The triggering event can be defined at the **interface** via the PLC program:

VB3300 4004 offers the following information:

0 No active event

Bit 0 = 1 Part program start from channel status RESET

Bit 1 = 1 Part program end

Bit 2 = 1 Operator panel reset

Bit 3 = 1 Ramp-up

Bit 4 = 1 First start after the search run

Bit 5-7 reserved, currently always 0

With the general request to 0, it is possible to determine whether an event is present. If a running event disappears upon RESET, the associated display bit in the interface extinguishes. For very brief events, the corresponding bit remains for at least the duration of a complete PLC cycle.

- Each time MD20108 `PROG_EVENT_MASK` is reconfigured, `/_N_CMA_DIR/_N_PROG_EVENT_SPF` must be loaded or enabled. Otherwise, the alarm 14011 "Program `_N_PROG_EVENT_SPF` does not exist or not enabled for execution" is output.
- The display can be suppressed in the current block display using the `DISPLOF` attribute in the `PROC` statement.
- A single block stop can be disabled with `SBLOF` attribute command or via MD10702 `IGNORE_SINGLEBLOCK_MASK` (prevent single block stop) with Bit 0.

The response to read-in disable and single-block processing can be controlled separately through the machine data MD20106 `PROG_EVENT_IGN_SINGLEBLOCK` (Prog events ignore the single block) and MD20107 `PROG_EVENT_IGN_INHIBIT` (Prog events ignore the read-in disable).

MD20106 PROG_EVENT_IGN_SINGLEBLOCK:

_N_PROG_EVENT_SPF causes a block change despite single block without a further start when

Bit 0 = 1 is set, after Part program start event

Bit 1 = 1 is set, after Part program end event

Bit 2 = 1 is set, after Operator panel reset event

Bit 3 = 1 is set, after Ramp-up event

Bit 4 = 1 is set, after First start after search run event

MD 20107: PROG_EVENT_IGN_INHIBIT:

_N_PROG_EVENT_SPF causes a block change despite read-in disable when

Bit 0 = 1 is set, after Part program start event

Bit 1 = 1 is set, after Part program end event

Bit 2 = 1 is set, after Operator panel reset event

Bit 3 = 1 is set, after Ramp-up event

Bit 4 = 1 is set, after First start after search run event

The following constraint applies for Bit 0 == 1 (program event after part program start):

If the program event ends with the part program command "RET", then RET always leads to an executable block (analogous to M17).

There is no new behavior for Bit 0 == 0, i.e. RET is interpreted in the interpreter and does not lead to an "executable block".

No sequences for **start/end of part program** are passed:

- If a user ASUB is started from the reset status, the described sequences for the event for start/end of part program are not passed.

- **Settable Prog-Event properties**

Machine data MD20109 PROG_EVENT_MASK_PROPERTIES can be used to define further properties of "event-driven program calls" for specific channels:

- Bit0 = 0: An ASUB started from the RESET channel state is followed by an "event-driven program call" as in earlier versions
- Bit0 = 1: An ASUB started from the RESET channel state is not followed by an "event-driven program call"

With the **Part program start**:

/_N_CMA_DIR/_N_PROG_EVENT_SPF is executed as a subroutine.

_N_PROG_EVENT_SPF must be ended with M17 or RET. A return by means of REPOS command is not permitted and triggers alarm 16020 "Repositioning not possible".

Error **with operator panel reset** or **after ramp-up**:

If EMERGENCY STOP or an operating mode / NCK error is still present when the operator panel is reset or after rampup, then _N_PROG_EVENT_SPF will only be processed after EMERGENCY STOP has been acknowledged or the error has been acknowledged in the channel.

Assignment example

MD20106 PROG_EVENT_IGN_SINGLEBLOCK = 'H1F'

MD20107 PROG_EVENT_IGN_INHIBIT = 'HC'

MD20109 PROG_EVENT_MASK_PROPERTIES = 'H1'

Event programs

Example for call by all events

MD20108 PROG_EVENT_MASK = 'H0F' (event-driven program call),
i.e. call of _N_PROG_EVENT_SPF during part program start, part program end, operator
panel reset and ramp-up:

PROC PROG_EVENT DISPLOF

Sequence for **part program start**

```
IF ($P_PROG_EVENT == 1)
N 10 R100 = 0 ; Transfer parameters for machining cycles
N 20 M17
ENDIF
```

Sequence for **part program end and operator panel reset**

```
IF ($P_PROG_EVENT == 2) OR ($P_PROG_EVENT == 3)
N10 R20 = 5
N20 ENDIF
N30 M17
ENDIF
```

Sequence for **powerup**

```
IF ($P_PROG_EVENT == 4)
N10 $SA_SPIND_S[Ax4] = 0 ; Speed for spindle start through virtual
interface
N20 ENDIF
N30 M17
ENDIF
M17
```


Start with RESET key

One of the following part programs is automatically started with the RESET key:

- Whose name is in MD11620 PROG_EVENT_NAME (program name for Prog Event) and that has been stored in one of the /_N_CUS_DIR/ or /_N_CMA_DIR/ directories
- _N_PROG_EVENT_SPF (default).

Control via MD20107 PROG_EVENT_IGN_INHIBIT

If the following machine data settings are present:
MD20107 PROG_EVENT_IGN_INHIBIT= 'H04F'
MD20108 PROG_EVENT_MASK= 'H04F'

The program started with the RESET key is executed right up to the end independently of a possibly set read-in disable.

Note

Recommendation for MD11450 with block search:

MD11450 SEARCH_RUN_MODE = 'H7' (search parameterization)

Bit 0 = 1:

With the loading of the last action block after block search, the processing is stopped and the VDI signal "Last action block active" is set. Alarm 10208 is not output until the PLC requests this by setting the VDI signal "PLC action ended".

Application: PLC starts an ASUB after block search.

Bit 1 = 1

Automatic ASUB start after output of the action blocks (see also MD11620 PROG_EVENT_NAME). Alarm 10208 is not output until the ASUB is completed.

Bit 2 = 1:

Output of the auxiliary functions is suppressed in the action blocks. The spindle programming that accumulated during the block search can be output at a later point in time (e.g. in an ASUB).

The program data for this is stored in the following system variables:

- \$P_SEARCH_S
 - \$P_SEARCH_SDIR
 - \$P_SEARCH_SGEAR
 - \$P_SEARCH_SPOS
 - \$P_SEARCH_SPOSMODE
-

10.3.9 Asynchronous subroutines (ASUBs)

Function

It is possible to activate two different ASUBs (PLCASUP1_SPF and PLCASUP2_SPF) from the PLC via the ASUB interface area. Before an asynchronous subroutine (ASUB) can be started from the PLC, it must have been assigned to an interrupt number by an NC program or by the PI service ASUB (see also NC services VB1200 0000).

Once prepared in this way, it can be started at any time from the PLC. The NC program running is interrupted by the ASUB.

Only one ASUP can be started at one time. If the start signal for both ASUBs is to be set to logical 1 in a PLC cycle, the ASUBs are started in the sequence INT1 and then INT2.

The start signal must be set to logical 0 by the user once the ASUB has been completed or if an error has occurred.

Note

The call of the ASUB PI service must have been completed before an ASUB may be started.

Initialization

The initialization is performed via the ASUB PI service, see also Section "Starting PI services in the NCK area (A2)".

Starting an ASUB

The time sequence of an ASUB is shown in the following pulse diagram in the example of PLCASUP1_SPF. You can see from the table which interface signals are of relevance for PLCASUP2_SPF.

Table 10- 11 Assignment of the signals to the pulse diagram

| Signal | Address - PLCASUP1_SPF | Address - PLCASUP2_SPF |
|-----------------------------|------------------------|------------------------|
| Start | V3400 0000.0 | V3400 0001.0 |
| Being executed | V3400 1000.1 | V3400 1001.1 |
| Completed | V3400 1000.0 | V3400 1001.0 |
| Error | V3400 1000.3 | V3400 1001.3 |
| Interrupt no. not allocated | V3400 1000.2 | V3400 1001.2 |

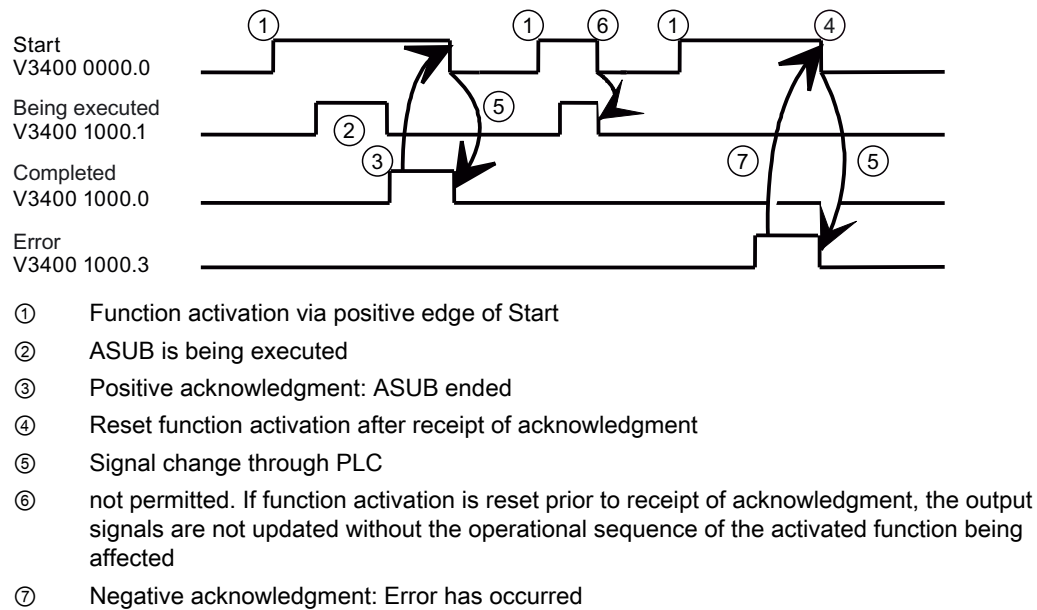


Figure 10-3 Pulse diagram for PLCASUP1_SPF

Configuration

The behavior of the ASUB can be influenced via the following standard machine data.

- MD11602 ASUP_START_MASK (ignore stop reasons for ASUB)
 The machine data specifies which stop reasons are to be ignored for an ASUB start.
 Recommended: MD11602 = 'H7'
- MD11604 ASUP_START_PRIO_LEVEL (priority, as of which MD11602 is effective)
 This machine data specifies the ASUB priority as of which machine data MD11602 ASUP_START_MASK is to be applied. MD11602 is applied from the level specified here up to the highest ASUB priority level 1.
 Recommended: MD11604 = 2

- MD20116 IGNORE_INHIBIT_ASUP (execute interrupt program in spite of read-in disable)

In spite of set read-in disable, an assigned user ASUB is processed completely for the interrupt channel with the set bit.

Bit 0 is assigned to interrupt channel 1 (PLCASUP1)

Bit 1 is assigned to interrupt channel 2 (PLCASUP2)

The machine data is effective only if MD11602 ASUP_START_MASK Bit2 = 0
- MD20117 IGNORE_SINGLEBLOCK_ASUP (execute interrupt program completely in spite of single block)

In spite of selected SBL processing mode, an assigned user ASUB is processed completely

for the interrupt channel with the set bit.

Bit 0 is assigned to interrupt channel 1 (PLCASUP1)

Bit 1 is assigned to interrupt channel 2 (PLCASUP2)

The machine data is effective only if

MD10702 IGNORE_SINGLE_BLOCK_MASK Bit1 = 0

10.3.10 Responses to operator or program actions

Responses

The following table shows the channel and program states that result after certain operator and program actions.

The left-hand side of the table shows the channel and program states and the mode groups from which the initial situation can be selected. Various operator/program actions are listed on the righthand side of the table, the number of the situation after the action has been carried out is shown in brackets after each action.

Table 10- 12 Responses to operator or program actions

| Situation | Channel status | | | Program status | | | Active mode | | | | Operator or program action (Situation after the action) |
|-----------|----------------|---|---|----------------|---|-----------------------|-------------|---|---|---|---|
| | R | U | A | N | U | Switchgear protection | A | A | M | J | |
| 1 | | x | | | | | x | x | | | RESET (4) |
| 2 | | x | | | | | x | | x | | RESET (5) |
| 3 | | x | | | | | x | | | x | RESET (6) |
| 4 | x | | | x | | | | x | | | NC Start (13); mode change (5 or 6) |
| 5 | x | | | x | | | | | x | | NC Start (14); mode change (4 or 6) |
| 6 | x | | | x | | | | | | x | Direction key (15); mode change (4 or 5) |
| 7 | | x | | x | | | | | x | | NC Start (14) |
| 8 | | x | | x | | | | | | x | NC Start (15) |

| Situation | Channel status | | | Program status | | | | Active mode | | | Operator or program action (Situation after the action) |
|-----------|----------------|---|---|----------------|---|-----------------------|---|-------------|---|---|--|
| | R | U | A | N | U | Switchgear protection | A | A | M | J | |
| 9 | | x | | | x | | | x | | | NC Start (13); mode change (10 or 11) |
| 10 | | x | | | x | | | | x | | NC Start (16); mode change (9 or 11) |
| 11 | | x | | | x | | | | | x | Direction key (17); mode change (9 or 10) |
| 12 | | x | | | | x | | x | | | NC Start (13); mode change (10 or 11) |
| 13 | | | x | | | | x | x | | | NC Stop (12) |
| 14 | | | x | x | | | | | x | | NC Stop (7); at block end (5) |
| 15 | | | x | x | | | | | | x | NC Stop (8); at JOG end (6) |
| 16 | | | x | | x | | | | x | | NC Stop (10); at block end (10) |
| 17 | | | x | | x | | | | | x | NC Stop (11); at JOG end (11) |

Description

Channel status:

R: aborted

U: interrupted

A: running

Program status:

N: aborted

U: interrupted

S: stopped

A: running

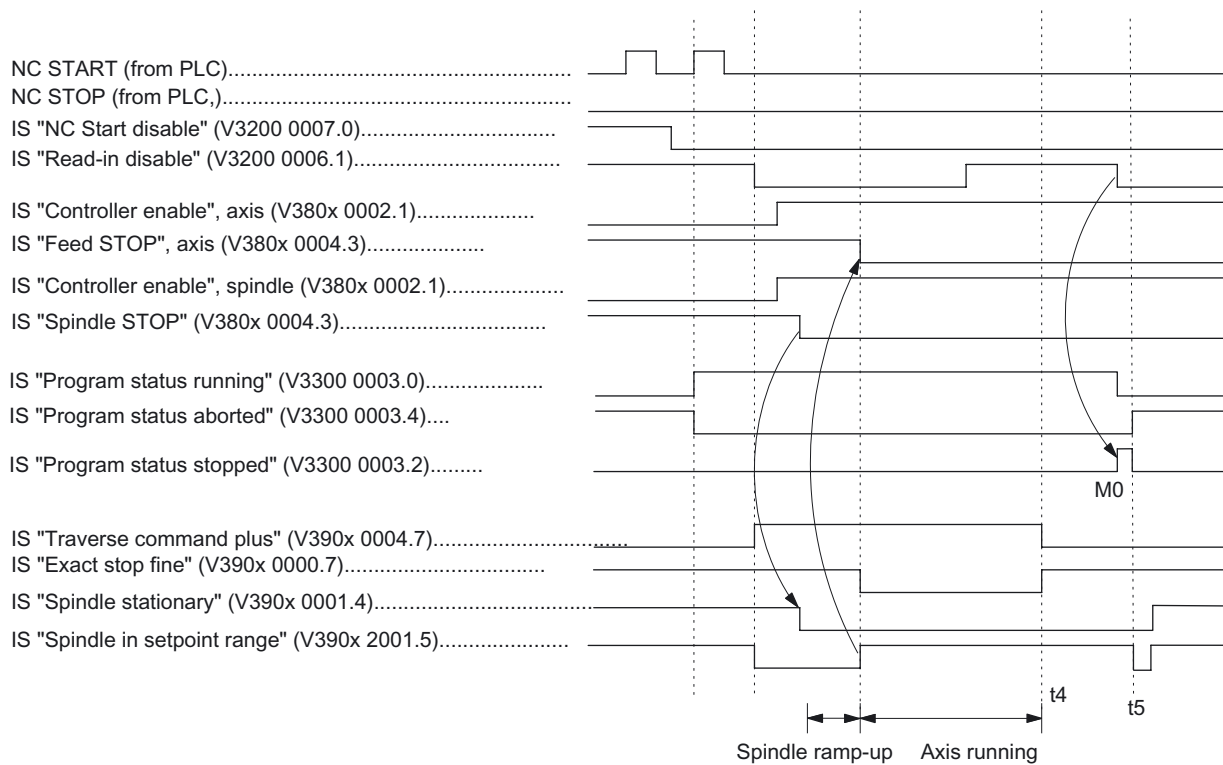
Operating modes:

A: AUTOMATIC

M: MDA

J: JOG

10.3.11 Example of a timing diagram for a program run



Explanation:

controlling input signals generated by PLC user program,
 t4: Block advance to N20 stopped with "Read-in disable",
 t5: Program aborted with RESET

Program:

N10 G01 G90 X100 M3 S1000 F1000 M88 N20 M0

Figure 10-4 Examples of signals during a program run

10.4 Program test

10.4.1 General information on the program test

Purpose

Several control functions are available for testing a new part program. These functions are provided to reduce danger at the machine and time required for the test phase. It is possible to activate several program test functions simultaneously.

The following test options are described here:

- Program processing without axis movements
- Program processing in singleblock mode
- Program processing with dry run feedrate

- Processing of certain program sections
- Skipping certain program parts
- Graphic simulation

10.4.2 Program processing without axis movements (PRT)

Functionality

The part program can be started and processed with active "Program test" function via the IS "NC Start" (V3200 0007.1), i.e. with auxiliary function outputs, dwell times. Only the axes/spindles are simulated. The software limit switch safety function continues to be valid.

The position control is not interrupted, so the axes do not have to be referenced when the function is switched off.

The user can check the programmed axis positions and auxiliary function outputs of a part program.

Note

Program processing without axis motion can also be activated with the function "Dry run feedrate".

Selection/activation

This function is selected via the user interface in the menu "Program control". IS "Program test selected" (V1700 0001.7) is set on selection of the function.

The PLC user program must activate the function via the IS "Activate program test" (V3200 0001.7).

Display

As a checkback for the active program test, "PRT" is displayed in the status line on the user interface and the IS "Program test active" (V3300 0001.7) is set in the PLC.

10.4.3 Program processing in single block mode (SBL)

Functionality

The user can execute a part program block-by-block to check the individual machining steps. Once the user decides that an executed part program block is functioning correctly, he/she can call the next block. The program is advanced to the next part program block via IS "NC Start" (V3200 0007.1).

When the function "single block" is activated, the part program stops after every program block during processing. In this case the activated single block type must be observed.

Single-block type

The following different types of single block are provided:

- Single block, coarse

With this type of single block, the blocks that initiate actions (traversing motions, auxiliary function outputs, etc.) are processed individually. If tool radius compensation is active (G41,G42), processing stops after every intermediate block inserted by the control. Processing is however not stopped at calculation blocks as these do not trigger actions.

- Single block, fine

With this type of single block, **all** blocks of the part program (even the pure computation blocks without traversing motions) are processed sequentially by "NC Start".

"Single block coarse" is the default setting after switching on.



In a series of G33 blocks single block is effective only if "dry run feedrate" is selected.

Selection/activation

The selection signal normally comes from a user machine control panel.

This function must be activated by the PLC user program via the IS "Activate single block" (V3200 0000.4).

The preselection whether "Single block coarse" or "Single block fine" type is made in the user interface in the "Program control" menu.

Display

The checkback signal that single block mode is active is displayed in the relevant "SBL" field on the operator interface.

Because of the single block mode, as soon as the part program processing has processed a part program block:

- The following interface signals are set:
 - IS "Channel status interrupted" (V3300 0003.6)
 - IS "Program status stopped" (V3300 0003.2)
- The following interface signals are reset:
 - IS "Channel status active" (V3300 0003.5)
 - IS "Program status running" (V3300 0003.0)

10.4.4 Program processing with dry run feedrate (DRY)

Functionality

The part program can be started via IS "NC Start" (V3200 0007.1). When the function is active, the traversing velocities programmed in conjunction with G1, G2, G3, CIP, and CT are replaced by the feed value stored in SD42100 DRY_RUN_FEED. The dry run feedrate also replaces the programmed revolutional feedrate in program blocks with G95. However, if the programmed feedrate is larger than the dry run feedrate, then the larger value is used.

| |
|--|
|  CAUTION |
|--|

| |
|--|
| Workpieces may not be machined when "dry run feedrate" is active because the altered feedrates might cause the permissible tool cutting rates to be exceeded and the workpiece or machine tool could be damaged. |
|--|

Selection/activation

Operation with dry run feedrate is selected in the "Position" operating area -> "Program control" softkey (AUTOMATIC mode). IS "Dry run feedrate" (V1700 0000.7) is set on selection of the function. In addition, the required dry run feedrate must be entered in the menu "Setting data". This does not activate the function.

This function is activated via the IS "Activate dry run feedrate" (V3200 0000.4) and is evaluated at NC start.

The dry run feedrate must be entered before program start in SD42100 DRY_RUN_FEED.

Display

The checkback signal that dry run feedrate is active is displayed in the relevant "DRY" status line on the user interface.

10.4.5 Block search: Processing of certain program sections

Functionality

To set the program run to a certain block (target block) of a part program, the block search function can be used. It can be selected whether or not the same calculations are to be performed during the block search up to the target block as would be performed during normal program operation.

After the target block is reached, the part program can be started via IS "NC Start" (give 2x) (V3200 0007.1). If necessary there is an automatic compensating movement of the axes to start or end positions of the target block. Execution of the remaining program then continues.

| |
|----------------|
| CAUTION |
|----------------|

| |
|--|
| Pay attention to a collision-free start position and appropriate active tools and other technological values! If necessary, a collisionfree start position must be approached manually with JOG. Select the target block considering the selected block search type. |
|--|

Selection/activation

The block search is selected in AUTOMATIC mode on the user interface.

The search run can be activated with corresponding softkey for the following functions:

- Block search with calculation to contour

Is used in any circumstances in order to approach the contour. On NC Start, the **start position of the target block** or the end position of the block before the target block is approached. This is traversed up to the end position. Processing is true to contour.

- Block search with calculation to block end point

Is used in any circumstances in order to approach a target position (e.g. tool change position). The **end position of the target block** or the next programmed position is approached using the type of interpolation valid in the target block. This is not true to contour. Only the axes programmed in the target block are moved.

- Block search without calculation.

Is used for a quick search in the main program. No calculations are performed. The internal controller values indicate the status valid before the search. Whether the program can be executed subsequently depends on the program and must be decided by the operator. This search run is suitable for a fast syntax check of a new program.

Interface signal

In the PLC, the following interface signals are set according to a time sequence (see figure):

- "Block search active" (V3300 0001.4)
- "Action block active" (V3300 0001.3)

- "Approach block active" (V3300 0000.4)

Note

The "Approach block active" is only enabled with "Block search with calculation on contour" because a separate approach block is not generated with "Block search with calculation at block end point" (the approach block is the same as the target block).

- "Last action block active" (V3300 0000.6)

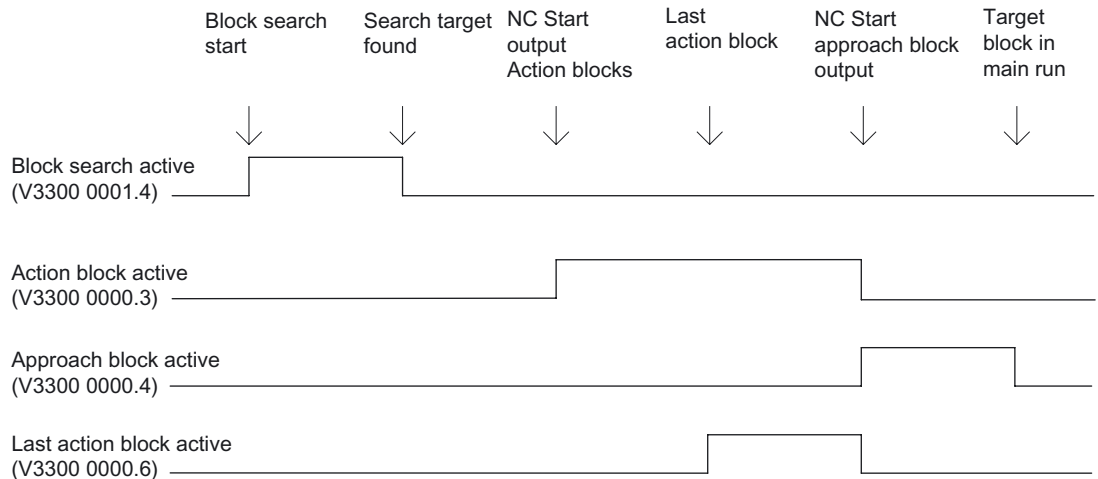


Figure 10-5 Chronological order of interface signals

After "Block search with calculation at block end point", automatic repositioning is not performed between "Last action block active" and continuation of part program processing by NC Start. The start point of the approach movement is the current axis position on NC Start; the end point results from the processing of the part program.

Action blocks

Action blocks contain the actions accumulated during "block search with calculation", e.g. auxiliary function outputs, and tool (T, D), spindle (S) and feed programming commands. During "block search with calculation" (contour or block end point), actions such as M function outputs are accumulated in so-called "action blocks". These blocks are output on an NC Start after "Search target found".

Note

The action blocks also activate the accumulated spindle programming (S value, M3/M4/M5, SPOS). The PLC user program must ensure that the tool can be operated and, if necessary, the spindle programming is reset via the IS "Spindle reset" (V380x 0002.2).

PLC actions after block search

There is the IS "Last action block active" to enable activation of PLC actions after block search. The signal indicates that all action blocks have been executed and it is now possible to perform PLC actions or operator actions (e.g. mode change). This allows the PLC to perform another tool change, for example, before the start of the movement.

The alarm 10208 is also output per default at this time. It should indicate to the operator that an NC start is still necessary to continue program processing.

Supplementary condition

The approach movement "Search with calculation to block end point" is performed using the type of interpolation valid in the target block. This should be G0 or G1, as appropriate. With other types of interpolation, the approach movement can be aborted with an alarm (e.g. circle end point error on G2/G3).

Note

For further information about the block search function, please see:

Reference:

/BP_/ Operation and Programming

10.4.6 Skip part program blocks (SKP)

Functionality

When testing or breaking in new programs, it is useful to be able to disable or skip certain part program blocks during program execution.

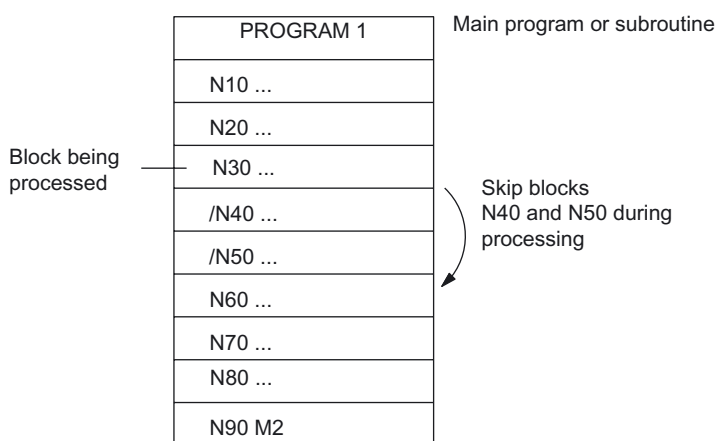


Figure 10-6 Skipping part program blocks

Selection/activation

The skip function is selected via the user interface in the menu "Program control". IS "Skip block selected" (V1700 0002.0) is set when the function is selected. In addition, a slash "/" must be written before the blocks to be skipped (see figure). This does not activate the function.

This function is activated via IS "Activate skip block" (V3200 0002.0).

Display

The checkback signal that the "Skip block" function is active is displayed in the relevant "SKP" status line on the user interface.

10.4.7 Graphic simulation

Function

In the AUTOMATIC operating mode a selected and opened program can be simulated graphically on the screen of the control unit. The movements of the programmed axes are recorded as line diagram after an "NC Start".

Selection/deselection

The graphic simulation can be reached for the selected program through the "Program" operating area, open program and "Simulation" softkey. Here the IS "Simulation active" (V1900 0000.6) is set and reset again on leaving the "Program" operating area or changing to "Edit".

Display

Due to numerous operating possibilities a complete workpiece, or else only enlarged details of it, can be displayed on the screen.

References:

/BP_/ Operation and Programming

PLC user program

The PLC user program must itself influence the required behavior of the control system in simulation, for example:

- Stop axes/spindle by transition into the program test: Set IS "Activate program test" (V3200 0001.7)
- Abort the running program if "Simulation" is exited by setting IS "Reset" (V3000 000.7), etc.

Display machine data

A number of display machine data (MD283 to MD292) is available for the user-specific configuration of the graphic simulation.

References:

/LIS/ SINUMERIK 802D sl lists

10.5 Timers for program execution time

Function

Timers are provided under the "Program execution time" function and these can be used for monitoring technological processes in the program or only in the display. These timers are read-only.

There are timers that are always active. Others can be deactivated via machine data.

Timers - always active

- Time since the last "Control powerup with default values" (in minutes):

\$AN_SETUP_TIME

The timer is automatically reset to zero in the case of a "Control power-up with default values".

- Time since the last control powerup (in minutes):

\$AN_POWERON_TIME

It is reset to zero automatically with each power-up of the control system.

Timers that can be deactivated

The following timers are activated via machine data (default setting). The start is timer-specific. Each active run-time measurement is automatically interrupted in the stopped program state or for feedrate-override = zero.

The behavior of the activated timers for active dry run feedrate and program testing can be specified using machine data.

- Total execution time in seconds of NC programs in the automatic mode (in seconds):
\$AC_OPERATING_TIME
In the AUTOMATIC mode, the runtimes of all programs between NC start and end of program / RESET are summed up. The timer is zeroed with each power-up of the control system.
- Runtime of the selected NC program (in seconds):
\$AC_CYCLE_TIME
The runtime between NC Start and End of program / Reset is measured in the selected NC program. The timer is reset with the start of a new NC program.
- Tool action time (in seconds):
\$AC_CUTTING_TIME
The runtime of the path axes is measured in all NC programs between NC START and end of program / RESET without rapid traverse active and with the tool active. The measurement is interrupted when a dwell time is active. The timer is automatically reset to zero in the case of a "Control power-up with default values".

Display

The contents of the timers are visible on the screen in the "OFFSET/PARAM" operating area -> "Setting data" softkey -> "Times/counters" softkey:

- **Run time** = \$AC_OPERATING_TIME
- **Cycle time** = \$AC_CYCLE_TIME
- **Cutting time** = \$AC_CUTTING_TIME
- **Setup time** = \$AN_SETUP_TIME
- **Power on time** = \$AN_POWERON_TIME

"Cycle time" is also visible in the AUTOMATIC mode in the "Position" operating area in the information line.

References:

/BP_/ Operation and Programming

10.6 Workpiece counter

Function

The "Workpiece counter" function provides counters for counting workpieces. These counters can be read and written by the program or by operation (note protection level for writing).

Range of values: 0 to 999 999 999.

The following channel-specific machine data can be used to control counter activation, counter reset timing and the counting algorithm.

- MD27880 PART_COUNTER (activation of workpiece counters)
- MD27882 PART_COUNTER_MCODE (workpiece counting with user-defined M command)

Counter

- Number of workpieces required (workpiece target):

\$AC_REQUIRED_PARTS

In this counter you can define the number of workpieces at which the actual workpiece counter \$AC_ACTUAL_PARTS is reset to zero.

MD27880 PART_COUNTER (Bit 0) can be used to generate the display alarm 21800 "Required number of workpieces reached" and to output the IS "Required number of workpieces reached" (V3300 40001.1).

- Total number of workpieces produced (total actual):

\$AC_TOTAL_PARTS

The counter specifies the total number of all workpieces produced since the start time.

- Number of actual workpieces (current actual):

\$AC_ACTUAL_PARTS

This counter registers the number of all workpieces produced since the starting time. The counter is automatically reset to zero (on condition that \$AC_REQUIRED_PARTS is not equal to 0) when the required number of workpieces (\$AC_REQUIRED_PARTS) has been reached.

- Number of workpieces specified by the user:

\$AC_SPECIAL_PARTS

This counter allows users to make a workpiece counting in accordance with their own definition. Alarm output can be defined for the case of identity with \$AC_REQUIRED_PARTS (workpiece target). Users must reset the counter themselves.

The first output of the M command for counting after resetting the counter applies as start point. This M command is set in MD27880 PART_COUNTER or MD27882 PART_COUNTER_MCODE for the relevant counter.

Display

The contents of the counters are visible on the screen in the "OFFSET/PARAM" operating area -> "Setting data" softkey -> "page down (2nd page):

- **Part total** = \$AC_TOTAL_PARTS
- **Part required**= \$AC_REQUIRED_PARTS
- **Part count** = \$AC_ACTUAL_PARTS
(\$AC_SPECIAL_PARTS not available for display)

"Part count" is visible in addition in the AUTOMATIC mode in the "Position" operating area in the information line.

References:

/BP_/ Operation and Programming

10.7 Data lists

10.7.1 Machine data

NC-specific machine data

| Number | Identifier | Name |
|---------|-------------------------|---------------------------------|
| General | | |
| 10702 | IGNORE_SINGLEBLOCK_MASK | Prevent single-block stop |
| 11450 | SEARCH_RUN_MODE | Block search parameter settings |
| 11602 | ASUP_START_MASK | Ignore stop conditions for ASUB |
| 11604 | ASUP_START_PRIO_LEVEL | Priorities for ASUP_START_MASK |
| 11620 | PROG_EVENT_NAME | Program name for program event |

Basic machine data of channel

| Number | Identifier | Name |
|------------------|----------------------------|--|
| Channel-specific | | |
| 20050 | AXCONF_GEOAX_ASSIGN_TAB[n] | Assignment between geometry axis and channel axis [GEOaxis no.]: 0...2 |
| 20060 | AXCONF_GEOAX_NAME_TAB[n] | Geometry axis name in channel [GEOaxis no.]: 0...2 |
| 20070 | AXCONF_MACHAX_USED[n] | Machine axis number valid in channel [channel axis no.]: 0...4 |
| 20080 | AXCONF_CHANAX_NAME_TAB[n] | Channel axis name in channel [channel axis no.]: 0...4 |
| 20100 | DIAMETER_AX_DEF | Geometry axis with transverse axis function |
| 20106 | PROG_EVENT_IGN_SINGLEBLOCK | Prog events ignore the single block |
| 20107 | PROG_EVENT_IGN_INHIBIT | Prog events ignore the read-in disable |
| 20108 | PROG_EVENT_MASK | Eventdriven program calls |
| 20109 | PROG_EVENT_MASK_PROPERTIES | Prog event properties |
| 20110 | RESET_MODE_MASK | Initial setting at RESET |
| 20112 | START_MODE_MASK | Initial setting at special NC Start after power-up and at RESET |
| 20116 | IGNORE_INHIBIT_ASUP | Execute user ASUBs completely in spite of readin disable |
| 20117 | IGNORE_SINGLEBLOCK_ASUP | Process user ASUBs completely in spite of single-block processing |
| 20700 | REFP_NC_START_LOCK | NC-Start disable without reference point |
| 21000 | CIRCLE_ERROR_CONST | Circle end point monitoring constant |
| 20150 | GCODE_RESET_VALUES | Reset G groups |
| 20152 | GCODE_RESET_MODE | G code basic setting at RESET |

Auxiliary function settings of the channel

| Number | Identifier | Name |
|------------------|---------------------------|--|
| Channel-specific | | |
| 22000 | AUXFU_ASSIGN_GROUP[n] | Auxiliary function group [aux. func. no. in channel]: 0...63 |
| 22010 | AUXFU_ASSIGN_TYPE[n] | Auxiliary function type [aux. func. no. in channel]: 0...63 |
| 22020 | AUXFU_ASSIGN_EXTENSION[n] | Auxiliary function extension [aux. func. no. in channel]: 0...63 |
| 22030 | AUXFU_ASSIGN_VALUE[n] | Auxiliary function value [aux. func. no. in channel]: 0...63 |
| 22550 | TOOL_CHANGE_MODE | New tool offset for M function |

Timers and counters of the channel

| Number | Identifier | Name |
|------------------|-----------------------|---|
| Channel-specific | | |
| 27860 | PROCESSTIMER_MODE | Activation of the program runtime measurement |
| 27880 | PART_COUNTER | Activation of the workpiece counters |
| 27882 | PART_COUNTER_MCODE[n] | Workpiece counting via M command, n = 0 ... 2 |

Display machine data

| Number | Identifier | Name |
|-------------|------------|---|
| 283 ... 292 | | Setting of the display for the graphic simulation |

10.7.2 Setting data

| Number | Identifier | Name |
|------------------|--------------------|---|
| Channel-specific | | |
| 42000 | THREAD_START_ANGLE | Start angle for thread |
| 42010 | THREAD_RAMP_DISP | Starting and deceleration distance of feed axis in thread cutting G33 |
| 42100 | DRY_RUN_FEED | Dry run feedrate |

10.7.3 Interface signals

Operating mode signals

| Number | Bit | Name |
|------------|-----|-----------------------------|
| PLC to NCK | | |
| V3000 0000 | .0 | AUTOMATIC mode |
| V3000 0000 | .1 | MDA mode |
| V3000 0000 | .2 | JOG mode |
| V3000 0000 | .4 | Mode change disable |
| V3000 0000 | .7 | RESET |
| V3000 0001 | .2 | Machine function REF |
| NCK to PLC | | |
| V3100 0000 | .0 | Active mode AUTOMATIC |
| V3100 0000 | .1 | Active mode MDA |
| V3100 0000 | .2 | Active JOG mode |
| V3100 0000 | .3 | 802 READY |
| V3100 0001 | .2 | Active machine function REF |

Channel signals

| Number | Bit | Name |
|------------|-----|--------------------------------|
| PLC to NCK | | |
| V3200 0000 | .4 | Activate single block |
| V3200 0000 | .5 | Activate M01 |
| V3200 0000 | .6 | Activate dry run feed |
| V3200 0001 | .0 | Activate referencing |
| V3200 0001 | .7 | Activate program test |
| V3200 0002 | .0 | Block skip |
| V3200 0006 | .0 | Feed disable |
| V3200 0006 | .1 | Read-in disable |
| V3200 0006 | .2 | Delete distance-to-go |
| V3200 0006 | .3 | Delete UP number of passes |
| V3200 0006 | .4 | Program level abort |
| V3200 0006 | .6 | Rapid traverse override active |
| V3200 0006 | .7 | Feed rate override active |
| V3200 0007 | .0 | NC Start disable |
| V3200 0007 | .1 | NC Start |
| V3200 0007 | .2 | NC Stop at block limit |
| V3200 0007 | .3 | NC stop |
| V3200 0007 | .4 | NC Stop axes plus spindles |
| V3200 0007 | .7 | Reset |

10.7 Data lists

| Number | Bit | Name |
|------------|-----|--|
| NCK to PLC | | |
| V3300 0000 | .3 | Action block active |
| V3300 0000 | .4 | Approach block active |
| V3300 0000 | .5 | M00/M01 active |
| V3300 0000 | .6 | Last action block active |
| V3300 0001 | .0 | Referencing active |
| V3300 0001 | .4 | Block search active |
| V3300 0001 | .5 | M2 / M30 active |
| V3300 0001 | .7 | Program test active |
| V3300 0003 | .0 | Program status: Running |
| V3300 0003 | .2 | Program status: Stopped |
| V3300 0003 | .3 | Program status: Interrupted |
| V3300 0003 | .4 | Program status: Aborted |
| V3300 0003 | .5 | Channel status: Active |
| V3300 0003 | .6 | Channel status: Interrupted |
| V3300 0003 | .7 | Channel status: Reset |
| V3300 4001 | .1 | Workpiece target reached |
| HMI to PLC | | |
| V1700 0000 | .5 | M01 selected |
| V1700 0000 | .6 | Dry run feed rate selected |
| V1700 0001 | .3 | Feed rate override selected for rapid traverse |
| V1700 0001 | .7 | Program test selected |
| V1700 0002 | .0 | Skip-block selected |
| V1900 0000 | .6 | Simulation active |

ASUB signals

| Number | Bit | Name | |
|------------|-----|-----------------------------|------|
| PLC to NCK | | | |
| V3400 0000 | .0 | INT1 Start | |
| V3400 0001 | .0 | INT2 Start | |
| V3400 1000 | .0 | ASUB ended | INT1 |
| V3400 1000 | .1 | ASUB is being executed | |
| V3400 1000 | .2 | Interrupt no. not allocated | |
| V3400 1000 | .3 | ASUB version not possible | |
| V3400 1001 | .0 | ASUB ended | INT2 |
| V3400 1001 | .1 | ASUB is being executed | |
| V3400 1001 | .2 | Interrupt no. not allocated | |
| V3400 1001 | .3 | ASUB version not possible | |

Compensation (K3)

11.1 Brief description

Compensations

For SINUMERIK 802D sl, the following axis-specific compensation functions can be activated:

- Backlash compensation
- Interpolatory compensation
 - LEC (leadscrew error and measuring system error compensation)
 - Sag compensation and angularity error compensation
- Following error compensation (speed feedforward control)

These compensation functions can be set for each machine individually with axis-specific machine data.

Position display

The normal actual-value and setpoint position displays ignore the compensation values and show the position values of an ideal machine. The compensation values are displayed in the "System" -> "Service Display" -> "Service Axes" operating area, under "Abs. compens. value meas. system 1".

11.2 Backlash compensation

Effect

In the case of axes/spindle with indirect measuring systems, mechanical backlash results in corruption of the traverse path, causing an axis, for example, to travel too much or too little by the amount of the backlash when the direction of movement is reversed (see following figure).

Compensation

To compensate for backlash, the axis-specific actual value is corrected by the amount of backlash every time the axis/spindle changes direction.

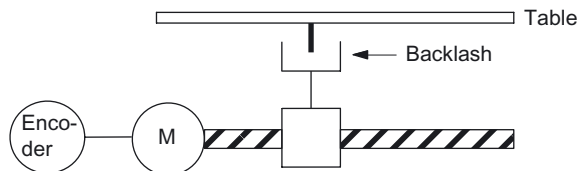
This quantity can be entered for each axis/spindle at the commissioning phase in MD32450 BACKLASH (backlash compensation)

Effectiveness

Backlash compensation is always active in all operating modes after reference point approach.

Positive backlash

The encoder leads the machine part (e.g. table). Since the actual position acquired by the encoder also leads the real actual position of the table, the table travels too short a distance (see diagram below). The backlash compensation value must be entered as a **positive** value here (= normal case).



Encoder actual value leads the real actual value (table):
The table does not traverse far enough

Figure 11-1 Positive backlash (normal case)

Negative backlash

The encoder lags behind the machine part (e.g. table); the table then travels too far. The correction value entered is **negative**.

High backlash compensation values

The user has the option of applying the backlash compensation value gradually in several increments when the relevant axis reverses direction. This prevents an excessive setpoint step change from causing specific axis errors.

The contents of the axis-specific MD36500 ENC_CHANGE_TOL determine the increment with which the backlash compensation value (MD32450 BACKLASH) is applied. Please note that the backlash compensation is fully calculated only after n servo cycles ($n = MD32450 / MD36500$). An excessive time span can cause the triggering of standstill monitoring alarms. If MD36500 is greater than MD32450, the compensation is performed in a servo cycle.

11.3 Interpolatory compensation

11.3.1 General

Terminology

Compensation value: The difference between the axis position measured by the position actual-value encoder and the required programmed axis position (= axis position of the ideal machine). The compensation value is often also referred to as the correction value.

Interpolation point: A position of the axis and the corresponding offset value.

Offset table: Table containing interpolation points

Compensation table

Because dimensional deviations between the leadscrew pitch and the measuring system directly affect the accuracy of workpiece machining, they must be compensated for by the relevant position-dependent compensation values. The compensation values are derived from measured error curves and entered in the control in the form of compensation tables during installation. A separate table must be created for each compensation relation.

The compensation values and additional table parameters are entered in the compensation tables using special system variables.

Entry of compensation table

Compensation tables can be loaded to the backed up NC user memory by two different methods.

- The compensation values are loaded when an NC program with the compensation tables is started.
- The compensation values can also be loaded by transferring the tables from a PC via the serial interface on the HMI.

Note

The compensation tables can be output via the serial interface on the HMI from operating area "System" -> "Data I/O" -> "Data selection" / Data ... / Compensation: Leadscrew error and loaded back following editing.

Linear interpolation between interpolation points

The traversing path to be compensated - defined using the start and end positions - is divided up into several (number depends on error curve shape) path segments of equal size (see figure). The actual positions that limit these sub-paths are designated "interpolation points". A compensation value must be entered for each interpolation point (actual position) during commissioning. The compensation value applied between two interpolation points is generated on the basis of **linear interpolation** using the compensation values for the adjacent interpolation points (i.e. adjacent interpolation points are linked along a line).

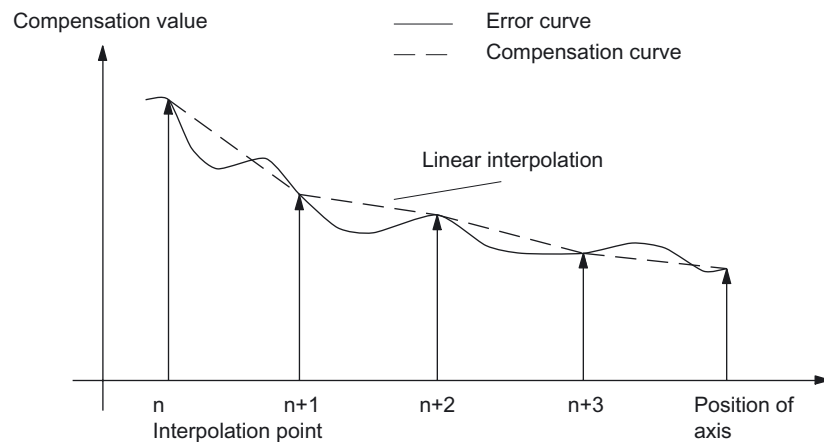


Figure 11-2 Linear interpolation between the interpolation points

Compensation value at reference point

The compensation table should be structured such that the compensation value at the reference point is "zero". This prevents position jumps occurring when the LEC is activated (after reference point approach).

11.3.2 LEC

Function

The leadscrew error compensation / measuring system error compensation (LEC) is an axis-specific compensation.

The principle of the LEC is to modify the axis-specific position actual value by the assigned compensation value in the interpolation cycle and to apply this value to the machine axis for immediate traversal. A positive compensation value causes the corresponding machine axis to move in the negative direction.

The magnitude of the compensation value is not limited and is not monitored. In order to avoid impermissibly high velocities and accelerations caused by compensation, small compensation values must be selected. Large compensation values can cause other axis monitoring functions to output alarms (e.g. contour monitoring, velocity setpoint limitation).

Effectiveness

- The compensation values are stored in the NC user memory and active (after POWER ON).
- The function has been activated for the relevant machine axis (MD32700 ENC_COMP_ENABLE [0] = 1).
- The axis has been referenced (IS "Referenced/synchronized 1" V390x 0000.4 set).

As soon as these conditions have been fulfilled, the axis-specific actual value is altered by the compensation value in all modes and traversed by the machine axis immediately.

If the reference is then lost, e.g. because the encoder frequency has been exceeded (IS "Referenced/synchronized 1" =0), compensation processing is de-activated.

Compensation table

The position-related compensation values are stored in the form of system variables for the relevant axis in the compensation table. 125 interpolation points (N = 0...124) are possible.

The following measuring-system-specific parameters must be set for the table (see Fig. "Compensation table parameters (system variables for LEC)"):

- **Compensation value for interpolation point N in compensation table:**

$\$AA_ENC_COMP [0,N,AXi]= \dots$

where: AXi = machine axis name, e.g. X1, Y1, Z1; N = interpolation point index

For every individual interpolation point (axis position) the compensation value must be entered in the table. The magnitude of the compensation value is not limited.

Note

The first and last compensation values remain active over the entire traversing range; i.e. these values should be set to "0" if the compensation table does not cover the entire traversing range.

- **Distance between interpolation points:** $\$AA_ENC_COMP_STEP[0,AXi]= \dots$

The distance between interpolation points defines the distance between the compensation values in the relevant compensation table (see above for AXi).

- **Starting position:** $\$AA_ENC_COMP_MIN[0,AXi]= \dots$

The starting position is the axis position at which the compensation table for the relevant axis begins (interpolation point 0).

The compensation value for the starting position is $\$AA_ENC_COMP[0,0,AXi]$.

The compensation value of interpolation point 0 is used for all positions smaller than the starting position (exception: table with modulo function).

- End position:** $\$AA_ENC_COMP_MAX[0,AXi]= \dots$


The end position is the axis position at which the compensation table for the relevant axis ends (interpolation point $k < 125$).

The compensation value for the end position is $\$AA_ENC_COMP[0,k,AXi]$

The compensation value of interpolation point k is used for all positions larger than the end position (exception: table with modulo function). Compensation values which are greater than k are inactive.
- Compensation with modulo function:** $\$AA_ENC_COMP_IS_MODULO[0,AXi] = 1$

When compensation with modulo function is activated, the compensation table is repeated cyclically; i.e. the compensation value at position $\$AA_ENC_COMP_MAX$ (interpolation point $\$AA_ENC_COMP[0,k,AXi]$) is immediately followed by the compensation value at position $\$AA_ENC_COMP_MIN$ (interpolation point $\$AA_ENC_COMP[0,0,AXi]$).

For rotary axes with modulo 360° it is therefore suitable to program 0° ($\$AA_ENC_COMP_MIN$) as the starting position and 360° ($\$AA_ENC_COMP_MAX$) as the end position. In this case both compensation values must be entered directly.

| |
|--|
|  CAUTION |
| <p>When the compensation values are entered, it is important that all interpolation points within the defined range be assigned a compensation value (i.e. there should be no gaps). Otherwise, the compensation value that was left over from previous entries at these positions is used for these interpolation points.</p> |

Note

Table parameters that contain position data are interpreted with MD10240 SCALING_SYSTEM_IS_METRIC=0 in inches.

The position data can be automatically re-calculated by performing a manual switchover (see the section titled "Manual switchover of the basic system").

The compensation table can only be loaded when MD32700 ENC_COMP_ENABLE=0 has been set. The value "1" causes the compensation to be activated and write protection to be applied (output alarm 17070).

Example

The following example shows compensation value inputs for machine axis X1 as a program.

```

%_N_AX_EEC_INI
CHANDATA (1)
$AA_ENC_COMP[0,0,X1]=0.0           ; 1st compensation value (interpolation point 0) +0 mm
$AA_ENC_COMP[0,1,X1] = 0.01        ; 2nd compensation value (interpolation point 1) +10 mm
$AA_ENC_COMP[0,2,X1]=0.012        ; 3rd compensation value (interpolation point 2) +12 mm
...
$AA_ENC_COMP[0,120,X1]=0.0         ; last compensation value (interpolation point 120)
$AA_ENC_COMP_STEP[0,X1]=2.0        ; distance between interpolation points 2.0 mm
    
```

```

$AA_ENC_COMP_MIN[0,X1]=-200.0 ; compensation starts at -200.0 mm
$AA_ENC_COMP_MAX[0,X1] = 40.0 ; compensation ends at +40.0 mm
$AA_ENC_COMP_IS_MODULO[0,X1]=0 ; compensation without modulo function M17
    
```

Values for more than 125 interpolation points result in alarm 12400 "Element does not exist".

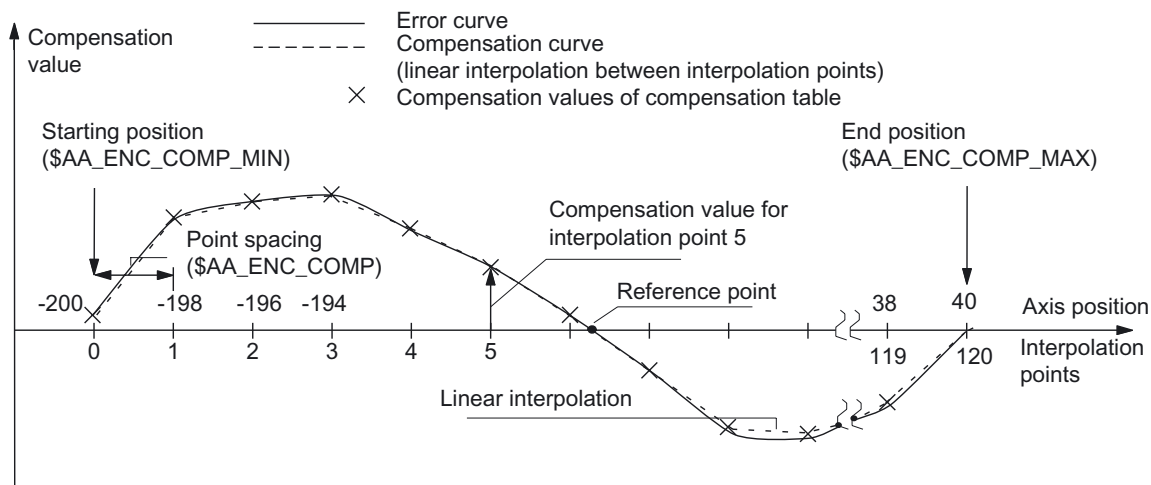


Figure 11-3 Compensation table parameters (system variables for LEC)

11.3.3 Sag compensation and angularity error compensation

Note

This function is under license and is not available with version T/M value.

If the corresponding option is activated without a valid license, alarm 8081 "%1 option(s) that has (have) not been licensed using a license key was (were) set" is output. It will not be possible to operate the machine as normal.

For information on operations relating to "Setting (an) option(s)", please refer to the section titled "Licensing in SINUMERIK 802D sl" in the "Turning, Milling, Grinding, Nibbling" Operating Instructions.

Prerequisites

The sag compensation and angularity error compensation functions cannot be applied for PLC axes.

Function

Unlike for LEC, the basic and compensation axes do **not need to be identical** for "Sag compensation" or "Angularity error compensation", which is why an axis assignment is required in every compensation table.

In order to compensate for sag of one axis (base axis) which results from its own weight, the absolute position of another axis (compensation axis) must be influenced. "Sag compensation" is therefore an **inter-axis compensation**.

As illustrated in the diagram below, the further the machining head moves in the negative Y1 axis direction, the more the cross-arm sags in the negative Z1 axis direction.

The error must be recorded in the form of a compensation table that contains a compensation value for the Z1 axis for every actual value position in the Y1 axis. It is sufficient to enter the compensation values for the interpolation points.

When the Y1 axis traverses, the control calculates the corresponding compensation value in the Z1 axis in interpolation cycles performing linear interpolation for positions between the interpolation points. This compensation is sent to the position control loop as an additional setpoint. A positive compensation value causes the corresponding machine axis to move in the negative direction.

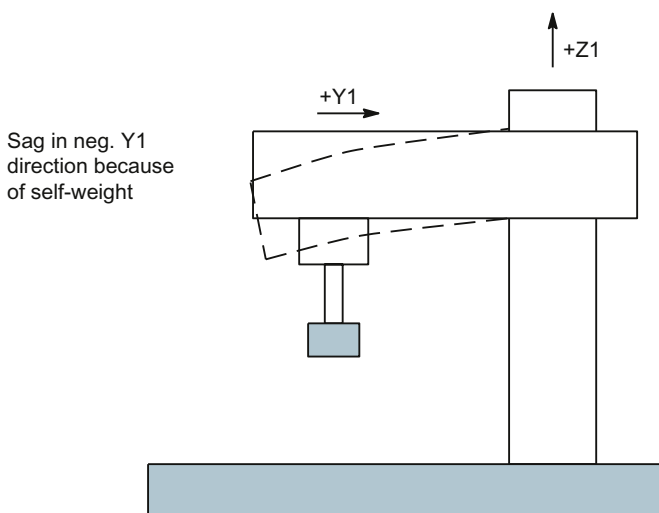


Figure 11-4 Example of sag caused by own weight

Depending on the requirement, several compensation relations can be defined for one axis. The total compensation value results from the sum of all the compensation values of this axis.

Note

You can read the compensation value that is currently effective on the user interface under the "Sag/temperature" variable. To do this, select softkey "System" followed by "Service Display" → "Service Axes".

Setting options

The many ways in which the compensation value for sag compensation can be produced/influenced are listed below (see diagram below).

1. An axis can be defined as the input variable (base axis) for **several** compensation tables (settable via system variables).
2. An axis can be defined as the recipient of the output variable (compensation axis) of **several** compensation tables (settable via system variable). The total compensation value is derived from the sum of the individual compensation values.

The following definition applies for the maximum number of possible compensation tables:

Total maximum number of tables available for all axes: 2

3. An axis can be both a base axis and a compensation axis at any one time. The programmed (required) position setpoint is always used to calculate the compensation values.
4. The scope of action of the compensation (starting and end position of the base axis) and the distance between the interpolation points can be defined for every compensation table (settable via system variables).
5. Compensation can be direction-dependent (settable via system variables).
6. Every compensation table has a modulo function for cyclic evaluation (settable via system variables).
7. A weighting factor by which the table value is multiplied (definable as a setting data which can therefore be altered by the part program, PLC or the user at any time) can be introduced for every compensation table.
8. Compensation tables can be multiplied in pairs (settable via system variables). The product is added to the total compensation value of the compensation axis.

9. The compensation can be activated in the following ways:

– Machine data:

MD32710 CEC_ENABLE [AXi] (sag compensation enable)

enables the sum of all compensation relations for machine axis AXi.

– Evaluation of the compensation table [t] is enabled with the following setting data:

SD41300 CEC_TABLE_ENABLE[t]

(presetting for compensation table)

It is thus possible, for example, to alter the compensation relations either from the part program or from the PLC user program (e.g. switching over the tables), depending on the machining requirements.

10. With setting:

MD10260 CONVERT_SCALING_SYSTEM = 1

(basic system switchover active)

activates the following axial machine data:

MD32711 CEC_SCALING_SYSTEM_METRIC

(system of units of sag compensation)

The measuring system for all tables effective for this axis is set in this machine data. Hereby all position entries are interpreted together with the calculated total compensation value in the configured measuring system. External conversions of position information are no longer necessary with a measuring system change.

Note

Template "_N_CEC.MPF" is provided for commissioning purposes and can be found in the following path when the Toolbox is installed:

C:\Program Files\Siemens\Toolbox

802D_s\V01040500\Techno\...\Compensation_Templates

Once you have adapted the template to meet the requirements of the machine manufacturer, you must transfer it to the control so that it can be executed as a part program.

Monitoring

To avoid excessive velocities and acceleration rates on the machine axis as a result of applying sag compensation, the total compensation value is monitored and limited to a maximum value.

The maximum compensation value is set for each axis with the following axial machine data:

MD32720 CEC_MAX_SUM

(maximum compensation value for sag compensation).

If the resulting total compensation value is greater than the maximum value, alarm 20124 "Sum of compensation values too high" is output. Program processing is not interrupted. The compensation value output as an additional setpoint is limited to the maximum value.

Alteration of the total compensation value is also limited axially.

When the limit value defined by the following machine data is violated, alarm 20125 "Too rapid change in compensation value" is generated:

MD32730 CEC_MAX_VELO (change in velocity with 1)

The program continues to run. The path not covered because of the limitation is made up as soon as the compensation value is no longer subject to limitation.

Complex compensation

Since it is possible to use the position of an axis as the input quantity (base axis) for several tables, to derive the total compensation value of an axis from several compensation relationships (tables) and to multiply tables, it is also possible to implement sophisticated and complex beam sag and angularity error compensation systems.

This function also makes it possible to deal with different error sources efficiently. For example, it is possible to combine a table with a modulo function for a periodic recurring error component with a second table without a modulo function for an aperiodic error component for the same axis.

Leadscrew errors can also be compensated with this function by parameterizing an identical axis for the base and compensation axes. Unlike with LEC, however, measuring-system switchovers are not automatically registered in this case.

Activation

The "sag compensation" function does not become active until the following conditions are fulfilled:

- The option "Sag compensation, multidimensional" has been enabled.
- The function has been activated for the relevant machine axis (compensation axis):
MD32710 CEC_ENABLE [AXi] = 1
- The compensation values have been stored in the non-volatile NC user memory and are active (after power ON).
- Evaluation of the relevant compensation table has been enabled:
SD41300 CEC_TABLE_ENABLE [t] = 1
- The current measuring system of the base and compensation axes has been referenced:
V390x 0000.4 = 1 (referenced/synchronized 1)

As soon as these conditions have been fulfilled the setpoint position of the compensation axis is altered in all modes with reference to the setpoint position of the base axis and the corresponding compensation value and is then immediately traversed by the machine axis.

If the reference is then lost, e.g. because the encoder frequency has been exceeded (V390x 0000.4 = 0), compensation processing is deactivated.

Compensation interpolation points

The number of required interpolation points in the compensation table must be defined for a compensation relationship: With 802D sl, 2 tables with 120 interpolation points each are provided.

Table parameters

The position-related corrections for the relevant compensation relationship are stored as system variables in the compensation table.

The following parameters must be set for the table:

- **Compensation value for interpolation point N of compensation table [t]**

$\$AN_CEC [t, N]$; $t = 0, 1, N = 0$ to 119

The compensation value of the compensation axis must be entered in the table for each individual interpolation point (position of the base axis).

- **Basic axis $\$AN_CEC_INPUT_AXIS[t]$**

Name of machine axis whose setpoint is to be used as the input for the compensation table [t].

- **Compensation axis $\$AN_CEC_OUTPUT_AXIS[t]$**

Name of machine axis to which the output of the compensation table [t] is to be applied.

- **Distance between interpolation points $\$AN_CEC_STEP[t]$**

The distance between interpolation points defines the distance between the input values for the compensation table [t].

$$\$AN_CEC_STEP [t] = \frac{\$AN_CEC_MAX [t] - \$AN_CEC_MIN [t]}{119}$$

- **Starting position $\$AN_CEC_MIN[t]$**

The initial position is the base axis position at which the compensation table [t] begins ($\hat{=}$ interpolation point 0).

The compensation value for the initial position is $\$AN_CEC [t, 0]$.

The compensation value of interpolation point 0 is used for all positions smaller than the initial position (exception: table with modulo functions).

- **End position $\$AN_CEC_MAX[t]$**

The end position is the base axis position at which the compensation table [t] ends ($\hat{=}$ interpolation point k).

The compensation value for the end position is $\$AN_CEC [t, k]$.

The compensation value of interpolation point k is used for all positions larger than the end position (exception: table with modulo functions).

The number of required interpolation points is calculated as follows, where $0 \leq k < 120$:

$$k = \frac{\$AN_CEC_MAX [t] - \$AN_CEC_MIN [t]}{\$AN_CEC_STEP [t]}$$

- **Direction-dependent compensation $\$AN_CEC_DIRECTION[t]$**

This system variable can be used to define whether the compensation table [t] should apply to both travel directions of the base axis or only either the positive or negative direction.

0: Table applies to both directions of travel of the base axis

1: Table applies only to position direction of travel of the base axis

-1: Table applies only to negative direction of travel of the base axis

Possible applications:

Position-dependent backlash compensation can be implemented using two tables, one of which affects the positive traversing direction, the other of which affects the negative traversing direction of the same axis.

- **Table multiplication** \$AN_CEC_MULT_BY_TABLE[t]

With the table multiplication function, the compensation values of any table can be multiplied by those of any other table (or even by the same table). The product is added as an additional compensation value to the total compensation value of the compensation table.

Syntax:

\$AN_CEC_MULT_BY_TABLE[t₁] = t₂

where:

t₁ = index of table 1 of compensation axis

t₂ = number of table 2 of compensation axis: Please note that the number and index **of the same table** are different! In general:

Table number = table index + 1

- **Compensation with modulo function** \$AN_CEC_IS_MODULO[t]

When compensation with modulo function is activated, the compensation table is repeated cyclically, i.e. the compensation value at position \$AN_CEC_MAX[t] (interpolation point \$AN_CEC[t,k]) is immediately followed by the compensation value at position \$AN_CEC_MIN[t] (interpolation point \$AN_CEC[t,0]).

These two compensation values should be the same as otherwise the compensation value jumps from MAX to MIN at the transition point and vice versa.

\$AN_CEC_IS_MODULO[t]= 0: Compensation without modulo function

\$AN_CEC_IS_MODULO[t]= 1: Compensation with modulo function

If modulo compensation is to be implemented with a modulo rotary axis as base axis, the compensation table used has to be modulo calculated as well.

Example:

MD30300 IS_ROT_AX[AX1] = 1; rotary axis

MD30310 ROT_IS_MODULO[AX1] = 1; modulo 360°

\$AN_CEC_INPUT_AXIS[0] = AX1

\$AN_CEC_MIN[0] = 0.0

\$AN_CEC_MAX[0] = 360.0

\$AN_CEC_IS_MODULO[0] = 1

Note

Table parameters containing position data are automatically converted when the scaling system is changed (when the setting in the following machine data is altered):

MD10240 SCALING_SYSTEM_IS_METRIC (basic system metric)

The position information is always interpreted in the current measuring system. Conversion must be implemented externally.

Where the following setting applies:

MD10260 CONVERT_SCALING_SYSTEM = 1 (basic system switchover active),

the scaling system can be configured using the following machine data:

MD32711 CEC_SCALING_SYSTEM (scaling system for sag compensation)

Table example

The following example shows the compensation table for sag compensation of axis Y1. Depending on the position of the Y1 axis, a compensation value is applied to the X1/Z1 axis.

```

%_N_NC_CEC_INI
CHANDATA(1)
$AN_CEC [0,0]          =0           ; 1st compensation value (≠ interpolation point 0)
                        ; for X1: ± 0 µm
$AN_CEC [0,1]          =0.01        ; 2nd compensation value (≠ interpolation point 1)
                        ; for X1: + 10 µm
$AN_CEC [0,2]          =0.012       ; 3rd compensation value (≠ interpolation point 2)
                        ; for X1: + 12 µm
...
$AN_CEC[0,119]        = 0           ; last compensation value
                        ; (≠ interpolation point 120) for X1: ± 0 µm
$AN_CEC_INPUT_AXIS[0] = (AX2)      ; basic axis Y1
$AN_CEC_OUTPUT_AXIS[0] = (AX1)     ; compensation axis X1
$AN_CEC_STEP[0]        =6.73       ; distance between interpolation points 6.73 mm
$AN_CEC_MIN[0]         =-400.0     ; compensation starts at
                        ; Y1 = -400 mm
$AN_CEC_MAX[0]         =400.0      ; compensation ends at Y1 = +400 mm
$AN_CEC_DIRECTION[0]  =0           ; table applies in both
                        ; traversing directions of Y1
$AN_CEC_MULT_BY_TABLE[0] =0
$AN_CEC_IS_MODULO[0]   =0           ; compensation without modulo function
$AN_CEC [1,0]          =0           ; 1st compensation value (≠ interpolation point 0)
                        ; for Z1: ± 0 µm
$AN_CEC [1,1]          =0.01        ; 2nd compensation value (≠ interpolation point 1)
                        ; for Z1: + 10 µm
$AN_CEC [1,2]          =0.012       ; 3rd compensation value (≠ interpolation point 2)
                        ; for Z1: + 12 µm
...
$AN_CEC [1,119]       =0           ; last compensation value
                        ; (≠ interpolation point 120) for Z1: ± 0 µm
$AN_CEC_INPUT_AXIS[1] = (AX2)      ; basic axis Y1
$AN_CEC_OUTPUT_AXIS[1] = (AX3)     ; compensation axis Z1
$AN_CEC_STEP[1]        =6.73       ; distance between interpolation points 6.73 mm
$AN_CEC_MIN[1]         =-400.0     ; compensation starts at
                        ; Y1 = -400 mm
$AN_CEC_MAX[1]         =400.0      ; compensation ends at Y1 = +400 mm
$AN_CEC_DIRECTION[1]  =0           ; table applies in both
                        ; traversing directions of Y1
$AN_CEC_MULT_BY_TABLE[1] =0
$AN_CEC_IS_MODULO[1]   =0           ; compensation without modulo function
M17
    
```

Table multiplication

With the table compensation function, any table can be multiplied by any other table (i.e. even by itself). The multiplication link is established using the system variables described above.

The following example for the compensation of machine foundation sagging illustrates an application of table multiplication.

On large machines, sagging of the foundation can cause inclination of the whole machine. For the boring mill in the second diagram below, for example, it is determined that compensation of the X1 axis is dependent both on the position of the X1 axis itself (since this determines angle of inclination β) and on the height of the boring mill (i.e. the position of the Z1 axis).

To implement compensation, the compensation values of the X1 and Z1 axes must be multiplied according to the following equation:

$$\Delta X1 = Z1 * \sin\beta(X1) \approx Z1 * \beta(X1)$$

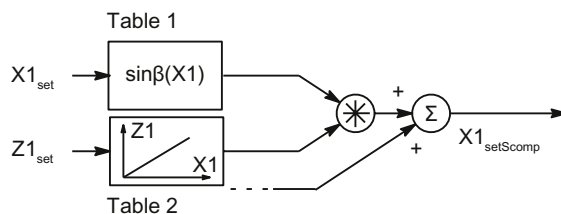


Figure 11-6 Table multiplication

Compensation table 1 (table index = 0) describes the reaction of axis X1 on axis X1 (sine of the position-dependent tilting angle $\beta(X1)$).

Compensation table 2 (table index = 1) describes the reaction of axis Z1 on axis X1 (linear).

In table 1, the multiplication of table 1 (index = 0) with table 2 is to be selected:

$$\$AN_CEC_MULT_BY_TABLE[0] = 2$$

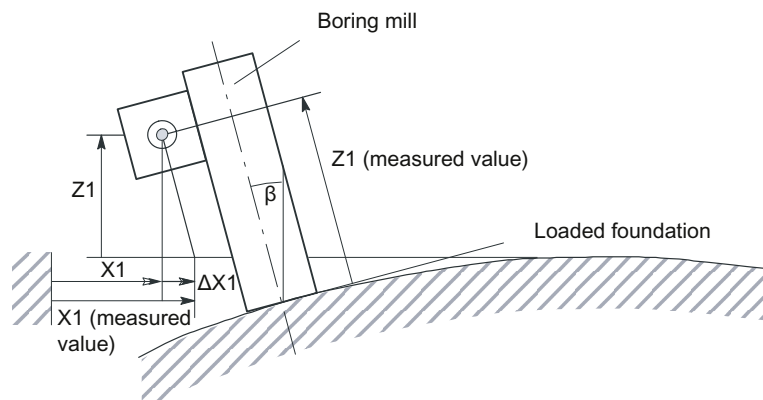


Figure 11-7 Compensation of sag in a foundation

11.3.4 Special features of interpolatory compensation

Measurement

The "Measurement" function supplies the compensated actual position values required by the machine operator or programmer.

TEACH IN

The "TEACH IN" function also uses compensated actual position values to determine the positions to be stored.

Software limit switches

The compensated actual position values are also monitored at the software limit switches.

Position display

The position actual-value indicator in the machine coordinate system shows the actual position value programmed for the axis.

On the "Service Axes" screen in the "System" → "Service Display" operating area, the actual position value determined by the measuring system plus the sum of the backlash compensation and leadscrew error compensation is displayed (= actual position value for measuring system 1).

Compensation value display

The "Service Axes" screen also shows the following compensated actual position values:

| Service display axes | Meaning |
|--|--|
| Absolute compensation value for measuring system 1 | Value displayed is the total compensation value calculated from "Leadscrew error compensation" and "Backlash compensation" for the current actual position of the axis (measuring system 1). |
| Sag compensation value | Value displayed is the total compensation value calculated from "Sag compensation" for the current actual position of the axis. |

Reference point loss

If the reference point of the basic axis is lost (IS "Referenced/synchronized 1" V390x 0000.4 = 0), the LEC and sag compensation functions are deactivated in the affected axes. When the reference point is reached, these compensations are automatically reactivated.

Setting servo enables

As a result of the compensation relationship, a traversing motion by the basic axis may also cause the compensation axis to move, making it necessary for servo enables to be set for these axes too (PLC user program). Otherwise, the compensation function will be restricted.

Traversing signal output

The traversing signals in the compensation axis are output when compensation is activated/deactivated, as well as each time the number of active compensation tables changes.

Any change in the compensation value caused by the basic axis motion does not result in traversing signals being output in the compensation axis.

11.4 Following error compensation (feedforward control)

11.4.1 General

Axis-specific following error

The following error can be reduced to almost zero with the help of the feedforward control. This feedforward control is therefore also called "following error compensation".

Particularly during acceleration in contour curvatures, e.g. circles and corners, this following error leads to undesirable, velocity-dependent contour violations. The SINUMERIK 802D sl control is equipped with the "Speed feedforward control" feedforward control type.

Activation/deactivation in part program

The feedforward control can be activated and de-activated by means of the following high-level language elements in the part program:

- FFWON - Feedforward control ON
- FFWOF - Feedforward control OFF (activation setting)

With MD32630 FFW_ACTIVATION_MODE, the activation (via FFWON) or de-activation (via FFWOF) of a specific axis can be determined:

- FFWON and FFWOF are used to activate and de-activate respectively the feedforward control of all axes/spindles for which MD32630=1 is set.
- MD 32630 should therefore have identical settings for axes that interpolate with each other.

The feedforward control should only be switched on or off while the axis/spindle is stationary to prevent jerk. This is the responsibility of the programmer.

Conditions

The following points should be noted before the feedforward control is applied:

- Rigid machine behavior
- Precise knowledge about the machine dynamic response
- No sudden changes in the position and speed setpoints

Optimization of control loop

The feedforward control is set on an axis/spindle-specific basis. First of all, the current control loop, speed control loop and position control loop must be set to an optimum for the axis/spindle.

Reference:

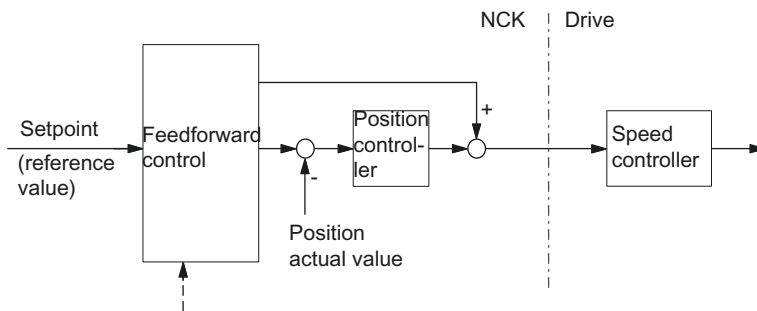
/BE/ Operating Instructions

Parameter assignments

The feedforward control parameters must then be assigned to the relevant axis/spindle and then entered in the machine data (see next section).

11.4.2 Speed feedforward control

In the case of speed feedforward control, a velocity setpoint is also applied directly to the input of the speed controller (see figure below).



MD 32810: EQUIV_SPEEDCTRL_TIME

Figure 11-8 Speed feedforward control

Parameters

In order to achieve a correctly set speed feedforward control, the equivalent time constant of the speed control loop must be determined exactly and entered as machine data MD32810 EQUIV_SPEEDCTRL_TIME (equivalent time constant of the closed speed control loop) during commissioning.

11.5 Data lists

11.5.1 Machine data

| Number | Identifier | Name |
|---------------|-------------------------------|---|
| Axis-specific | | |
| 32450 | BACKLASH[0] | Backlash |
| 32630 | FFW_ACTIVATION_MODE | Feedforward control can be activated from the program |
| 32700 | ENC_COMP_ENABLE[0] | Interpolatory compensation active |
| 32710 | CEC_ENABLE | Sag compensation enable |
| 32711 | CEC_SCALING_SYSTEM_METRIC | Scaling system for sag compensation |
| 32720 | CEC_MAX_SUM | Maximum compensation value for sag compensation |
| 32730 | CEC_MAX_VELO | Change in velocity with CEC |
| 32810 | EQUIV_SPEEDCTRL_TIME[0]...[5] | Equivalent time constant of the speed control loop |
| 36500 | ENC_CHANGE_TOL | Backlash compensation partial section |
| 38000 | MM_ENC_COMP_MAX_POINTS[0] | Interpolation points for encoder/spindle compensation (LEC) (for display only) |

11.5.2 Setting data

| Number | Identifier | Name |
|---------|------------------|---------------------------|
| General | | |
| 41300 | CEC_TABLE_ENABLE | Compensation table enable |

11.5.3 Interface signals

| Number | Bit | Name |
|-----------------------|-----|---------------------------|
| Axis/spindle-specific | | |
| V390x 0000 | .4 | Referenced/synchronized 1 |

Kinematic Transformation (M1)

Note

This function is not available with version T/M value.

12.1 Brief description

Application range

The control transforms programmed traversing instructions from a Cartesian coordinate system into a real machine axis system.

The TRANSMIT transformation is used for the face-end milling of turned parts on lathes (without Y machine axis).

The TRACYL transformation is used to machine the peripheral surfaces of cylindrical bodies. The main application is the milling of grooves. A TRACYL variant is provided for lathes. A second variant is provided for lathes with an additional Y machine axis or for milling machines with a suitable rotary table.

Machine prerequisite

The lathe must be equipped with a C axis-capable main spindle. A second spindle must be able to drive the milling tool. When used with TRACYL, the milling machine must be equipped with a rotary table that is capable of interpolating with the other axes.

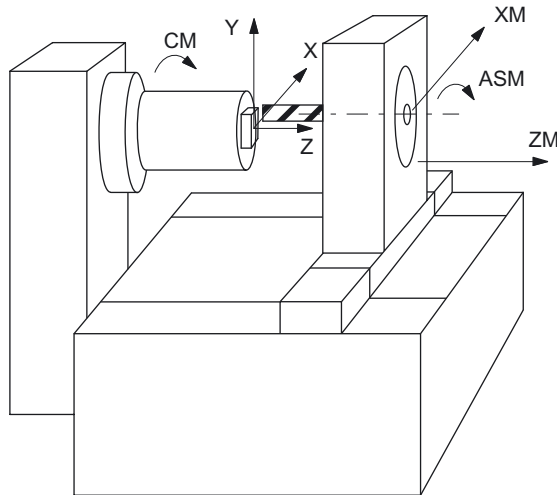
Availability

The TRANSMIT and TRACYL functions are configured using separate machine data sets and switched on or off by means of special instructions in the program.

With SINUMERIK 802D sl, a maximum of two kinematic transformations (TRANSMIT, TRACYL) may be configured and one of them may be activated using the program.

12.2 TRANSMIT

12.2.1 Overview



X, Y, Z Cartesian coordinate system for programming of the face-end machining

ASM Second spindle (work spindle for milling tool, drill)

ZM Z machine axis (linear)

XM X machine axis (linear)

CM C axis (main spindle as rotary axis)

Figure 12-1 Face-end milling of turned parts with TRANSMIT

Required machine kinematics

The two linear axes (XM, ZM) must be mutually perpendicular. The rotary axis (CM) must travel parallel to the linear axis ZM (rotating around ZM). The linear axis XM intersects the rotary axis CM (center of rotation).

Activation/de-activation of TRANSMIT

The TRANSMIT function is activated in the program with

- TRANSMIT in a separate block and de-activated with
- TRAFOOF in a separate block

TRAFOOF deactivates any active transformation function.

Programming - principle

```

N10 G0 X... Z... SPOS=...           ; starting positions, spindle in position control
N20 G17 G94 T...                     ; plane, feed type, select milling tool
N30 SETMS(2)                          ; switchover: master spindle is now the milling
                                       spindle
N40 TRANSMIT                          ; switch on TRANSMIT
N50 G1 G41 F200 X... Y... Z... M3     ; milling of the face with milling tool radius
S...                                  compensation
...
N90 G40 ...
N100 TRAFOOF                          ; switch off TRANSMIT
N110 G18 G95 T...                     ; switch back to turning
N120 SETMS                             ; master spindle is main spindle

```

Explanation:

The movement of the machine axes XM and CM produces the contour on the face-end of the turned part with the milling cutter in accordance with the X-Y path programmed (straight or circular path). The programmed Z axis (infeed) continues to be traversed as the Z axis.

12.2.2 TRANSMIT configuration

General

The TRANSMIT transformation function is configured using machine data settings.

Note

A file containing default machine data is available in the SINUMERIK 802D sl "Toolbox". A fast installation of TRANSMIT is possible by defining specific values and loading this file in the control.

Machine data

The names of the machine data, channel axes and geometry axes from the general machine data (\$MN_AXCONF...) and channel-specific machine data (\$MC_AXCONF...) are also used for a transformation.

The geometry axis assignments specified in MD20050 AXCONF_GEOAX_ASSIGN_TAB only apply when the transformation is de-activated. Additional assignments are specified for a transformation.

Note

The assigned machine axis names, channel axis names and geometry axis names must differ:

- MD10000 AXCONF_MACHAX_NAME_TAB,
- MD20080 AXCONF_CHANAX_NAME_TAB,
- MD20060 AXCONF_GEOAX_NAME_TAB.

Exception for TRANSMIT:

The axis names of MD20060 and MD20080 (geometry and channel axes) can be the same, e.g. X, Y, Z. No Y axis exists here outside the transformation.

Machine data for transformation

| | |
|-------------------------------------|--|
| MD24100 TRAFO_TYPE_1 | = 256 for first TRANSMIT transformation |
| MD24110 TRAFO_AXES_IN_1[n] | Channel axes for transformation 1 |
| MD24120 TRAFO_GEOAX_ASSIGN_TAB_1[n] | Geometry axes for transformation 1 |
| MD24200 TRAFO_TYPE_2 | = 256 for second TRANSMIT transformation |
| MD24210 TRAFO_AXES_IN_2[n] | Channel axes for transformation 2 |
| MD24220 TRAFO_GEOAX_ASSIGN_TAB_2[n] | Geometry axes for transformation 2 |

Required assignment of channel axes for TRANSMIT transformation in machine data MD24110/MD24210:

| | |
|-----------------------|--|
| TRAFO_AXES_IN_1/2[0]= | Channel axis number of axis perpendicular to rotary axis |
| TRAFO_AXES_IN_1/2[1]= | Channel axis number of rotary axis |
| TRAFO_AXES_IN_1/2[2]= | Channel axis number of axis parallel to rotary axis |

Machine data specifically for TRANSMIT

- MD24900 TRANSMIT_ROT_AX_OFFSET_1
Rotational position of Cartesian coordinate system x-y plane with respect to the defined zero position of rotary axis in degrees (0... < 360)
- MD24910 TRANSMIT_ROT_SIGN_IS_PLUS_1
If the rotary axis rotates in an anti-clockwise direction on the X-Y plane opposite to the positive Z axis, then the MD must be set to 1, otherwise to 0.

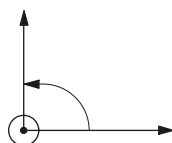


Figure 12-2 Direction of rotation for MD value = 1

- MD24920 TRANSMIT_BASE_TOOL_1

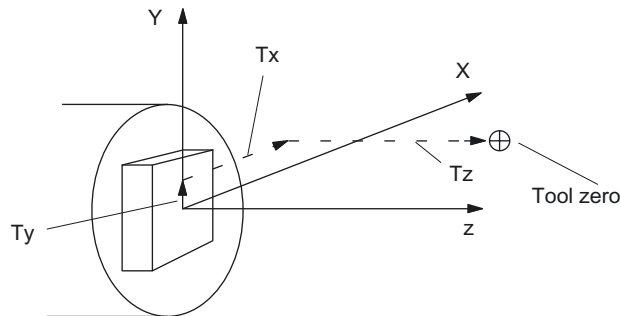
The control is informed of the position of the tool zero point in relation to the origin of the coordinate system declared for TRANSMIT. The MD has three components for the three axes of the Cartesian coordinate system.

Assignment of axis components:

TRANSMIT_BASE_TOOL_1[0]=Tx

TRANSMIT_BASE_TOOL_1[1]=Ty

TRANSMIT_BASE_TOOL_1[2]=Tz (see following figure)



Tx, Ty, Tz - axis components of position

Figure 12-3 Position of tool zero in relation to origin of the Cartesian coordinate system (center of rotation)

- MD24911 TRANSMIT_POLE_SIDE_FIX_1 = 0

Pole traversal continuous

Traversal of pole

The pole is defined as the center of rotation at point X=0, Y=0 of TRANSMIT plane (X machine axis intersects the center of rotation).

In the vicinity of the pole, small positional changes in the geometry axes X, Y generally result in large changes in the machine rotary axis position (exception: path only results in a movement of the XM axis).

Workpiece machining operations close to the pole are therefore not recommended since these may require sharp feedrate reductions to prevent overloading of the rotary axis. Avoid selecting TRANSMIT when the tool is positioned exactly on the pole. Ensure that the path of the tool center point does not travel through the X0/Y0 pole.

Examples: Machine data settings for TRANSMIT

General settings: Axis names: XM->X1, ZM->Z1, CM->SP1

- Machine axis name
MD10000 AXCONF_MACHAX_NAME_TAB[0]="X1"
MD10000 AXCONF_MACHAX_NAME_TAB[1]="Z1"
MD10000 AXCONF_MACHAX_NAME_TAB[2]="SP1"
MD10000 AXCONF_MACHAX_NAME_TAB[3]="SP2"
MD10000 AXCONF_MACHAX_NAME_TAB[4]=""
- Assignment of geometry axis to channel axis
MD20050 AXCONF_GEOAX_ASSIGN_TAB[0]=1
MD20050 AXCONF_GEOAX_ASSIGN_TAB[1]=0
MD20050 AXCONF_GEOAX_ASSIGN_TAB[2]=2
- Geometry axis names in channel
MD20060 AXCONF_GEOAX_NAME_TAB[0]="X"
MD20060 AXCONF_GEOAX_NAME_TAB[1]="Y"
MD20060 AXCONF_GEOAX_NAME_TAB[2]="Z"
- Valid machine axis numbers in channel
MD20070 AXCONF_MACHAX_USED[0]=1
MD20070 AXCONF_MACHAX_USED[1]=2
MD20070 AXCONF_MACHAX_USED[2]=3
MD20070 AXCONF_MACHAX_USED[3]=4
MD20070 AXCONF_MACHAX_USED[4]=0
- Channel axis names in channel
MD20080 AXCONF_CHANAX_NAME_TAB[0]="X"
MD20080 AXCONF_CHANAX_NAME_TAB[1]="Z"
MD20080 AXCONF_CHANAX_NAME_TAB[2]="C"
MD20080 AXCONF_CHANAX_NAME_TAB[3]="SP2"
MD20080 AXCONF_CHANAX_NAME_TAB[4]=""
- Initial setting of master spindle in channel
MD20090 SPIND_DEF_MASTER_SPIND=1

TRANSMIT transformation type:

- Definition of transformation 1 in channel
MD24100 TRAFO_TYPE_1=256
- Axis assignment for the 1st transformation in the channel
MD24110 TRAFO_AXES_IN_1[0] = 1
MD24110 TRAFO_AXES_IN_1[1] = 3
MD24110 TRAFO_AXES_IN_1[2] = 2
MD24110 TRAFO_AXES_IN_1[3]=0
MD24110 TRAFO_AXES_IN_1[4]=0
- Assignment of geometry axes to channel axes for transformation 1
MD24120 TRAFO_GEOAX_ASSIGN_TAB_1[0] = 1
MD24120 TRAFO_GEOAX_ASSIGN_TAB_1[1] = 3
MD24120 TRAFO_GEOAX_ASSIGN_TAB_1[2] = 2

Special TRANSMIT settings:

- Offset of rotary axis
MD24900 TRANSMIT_ROT_AX_OFFSET_1=0
- Sign of rotary axis
MD24910 TRANSMIT_ROT_SIGN_IS_PLUS_1=1
- Vector of base tool
MD24920 TRANSMIT_BASE_TOOL_1[0]=0
MD24920 TRANSMIT_BASE_TOOL_1[1]=0
MD24920 TRANSMIT_BASE_TOOL_1[2]=0

Setting data for the special treatment of the tool offset (only when required):

- Change of tool length component for change of plane
SD42940 TOOL_LENGTH_CONST=18
- Assignment of the tool length offset independent of tool type
SD42950 TOOL_LENGTH_TYPE=2

Settings for second spindle (milling spindle of the lathe):

- MD30300 IS_ROT_AX[AX4]=1
- MD30310 ROT_IS_MODULO[AX4]=1
- MD30320 DISPLAY_IS_MODULO[AX4]=1

12.3 TRACYL

- MD35000 SPIND_ASSIGN_TO_MACHAX[AX4]=2
- SD43300 ASSIGN_FEED_PER_REV_SOURCE[AX4]=0

Note

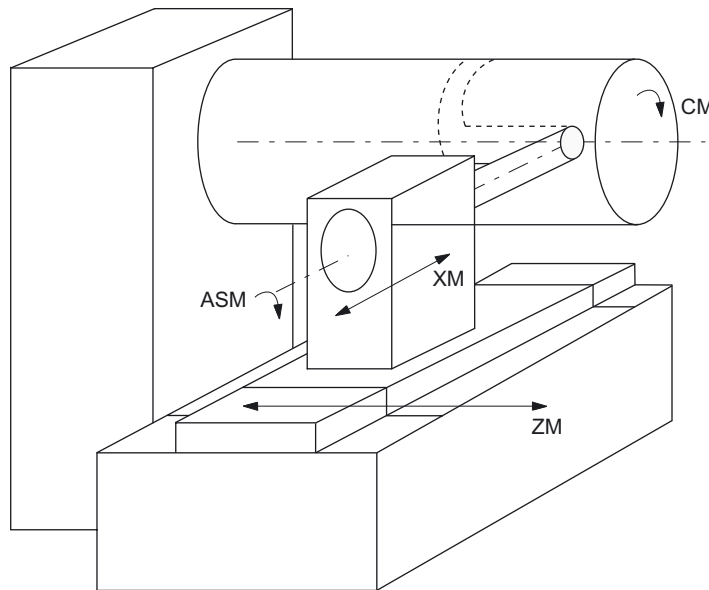
A special handling of milling tools on lathes with respect to length compensation is possible.

See Chapter "Tool: Compensation and Monitoring (W1)"

12.3 TRACYL

12.3.1 Overview

Standard lathe (without Y machine axis)



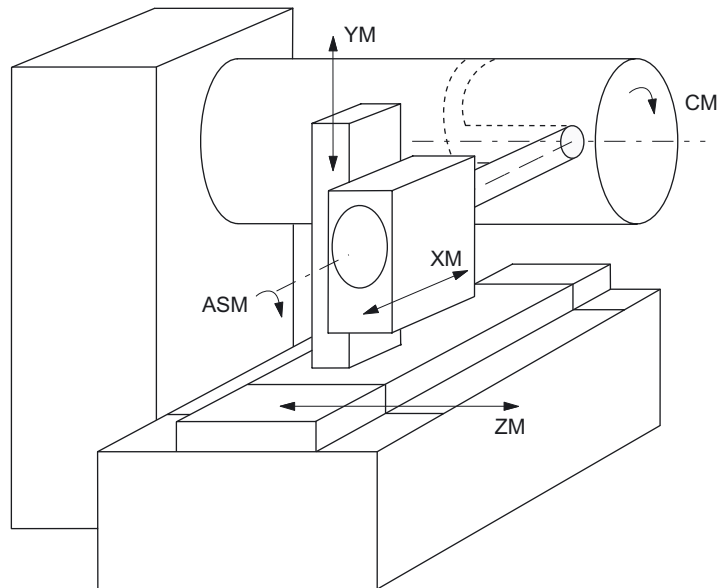
- Legend:
- XM Infeed axis, perpendicular to rotary axis
 - ZM Axis is parallel to rotary axis
 - CM Rotary axis
 - ASM Work spindle

Figure 12-4 Machining grooves on a cylinder surface with X-C-Z kinematics

Required machine kinematics

The two linear axes (XM, ZM) must be mutually perpendicular. The rotary axis (CM) must travel parallel to the linear axis ZM (rotating around ZM). The linear axis XM intersects the rotary axis CM (center of rotation).

Machine with Y axis



Legend:
 XM Infeed axis, perpendicular to rotary axis
 YM Additional axis
 ZM Axis is parallel to rotary axis
 CM Rotary axis
 ASM Work spindle

Figure 12-5 Machining grooves on a cylinder surface with X-Y-Z-C kinematics

Extended machine kinematics

The YM linear axis is also available to enable the machine kinematics requirements to be met (see above). This is arranged perpendicular to XM and ZM respectively and, with these, forms a right-handed Cartesian coordinate system.

This type of kinematics is typical for milling machines and makes it possible to machine grooves where the groove wall and groove base are mutually perpendicular – provided the milling tool diameter is less than the groove width (groove wall offset). These grooves can otherwise only be machined using milling tool diameters which fit precisely.

Grooves in transverse section

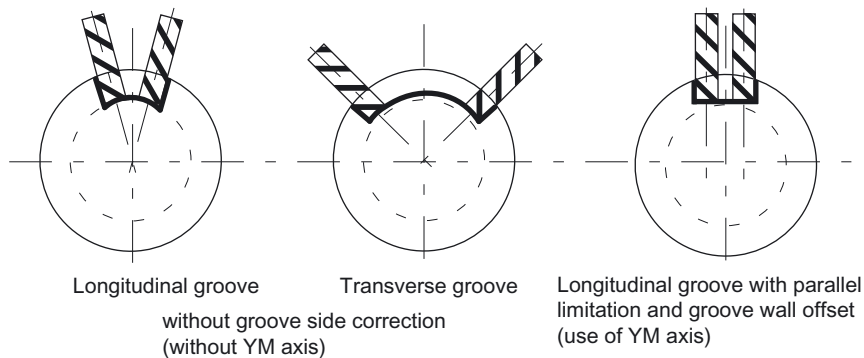


Figure 12-6 Grooves with and without groove wall offset

Activation/deactivation of TRACYL

The TRACYL function is activated in the program with

- TRACYL(d) in a separate block and deactivated with
- TRAFOOF in a separate block

d - machining diameter of the cylinder in mm

TRAFOOF deactivates any active transformation function.

Programming - principle

```

; without YM axis
; the geometry axes X, Y, Z are programmed
N10 G0 X... Z... ; starting positions, spindle in position control
SPOS=...
N20 G19 G94 T... ; plane, feed type, select milling tool
N30 SETMS(2) ; switchover: master spindle is now the milling spindle
N40 TRACYL(24.876) ; switch on TRACYL, diameter: 24.876 mm
N50 G1 F200 X... M3 ; feed, switch on milling spindle
S...
N600 G41 F200 Y... Z... ; milling of the cylinder surface with milling tool radius
compensation
...
N90 G40 ...
N100 TRAFOOF ; switch off TRACYL
N110 G18 G95 T... ; switch back to turning
N120 SETMS ; master spindle is main spindle
    
```

Explanation:

The movement of the machine axes ZM and CM produces this contour on the peripheral surface of the cylindrical workpiece with the milling cutter in accordance with the Y-Z path programmed (straight or circular). The programmed X axis (infeed) continues to be traversed as the X axis.

The cylinder unrolled at the outside diameter d results in peripheral surface with the Y-Z programming plane (G19). This is also used to determine the rotational direction of the circle for G2, G3.

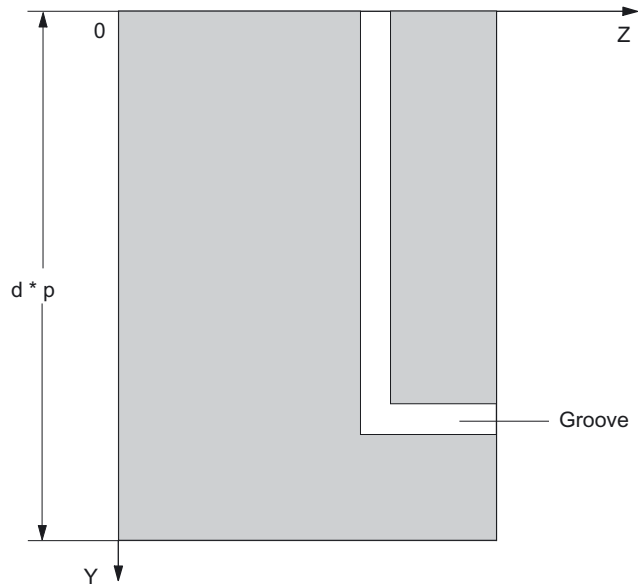


Figure 12-7 Peripheral surface of cylinder G19 (Y-Z plane)

OFFN address

Distance of groove side wall from the reference contour (also see "TRACYL programming example")

Programming: OFFN=...; Distance in mm

As a rule, the groove center line is programmed. OFFN determines the groove width when the milling radius compensation is active (G41, G42). Set OFFN=0 once the groove has been completed.

12.3.2 TRACYL configuration

General

The TRACYL transformation function is configured using machine data settings.

Note

A file containing default machine data is available in the SINUMERIK 802D "Toolbox". A fast installation of TRACYL is possible by defining specific values and loading this file in the control.

General machine data

The names of the machine data, channel axes and geometry axes from the general machine data (\$MN_AXCONF...) and channel-specific machine data (\$MC_AXCONF...) are also used for a transformation.

The geometry axis assignments specified in MD20050 AXCONF_GEOAX_ASSIGN_TAB only apply when the transformation is deactivated. Additional assignments are specified for a transformation.

Note

The assigned machine axis names, channel axis names and geometry axis names must differ:

- MD10000 AXCONF_MACHAX_NAME_TAB
- MD20080 AXCONF_CHANAX_NAME_TAB
- MD20060 AXCONF_GEOAX_NAME_TAB

Exception for TRACYL:

The axis names of MD20060 and MD20080 (geometry and channel axes) can be the same for the TRACYL transformation (e.g. X, Y, Z.), if no Y axis exists outside the transformation. This is usually the case for lathes.

Machine data for transformation

| | |
|-------------------------------------|--|
| MD24100 TRAFO_TYPE_1 | Def. for first TRACYL transformation *) |
| MD24110 TRAFO_AXES_IN_1[n] | Channel axes for transformation 1 |
| MD24120 TRAFO_GEOAX_ASSIGN_TAB_1[n] | Geometry axes for transformation 1 |
| MD24200 TRAFO_TYPE_2 | Def. for second TRACYL transformation *) |
| MD24210 TRAFO_AXES_IN_2[n] | Channel axes for transformation 2 |
| MD24220 TRAFO_GEOAX_ASSIGN_TAB_2[n] | Geometry axes for transformation 2 |
| *) =512/513 without/with YM axis | |

Required assignment of channel axes for TRACYL transformation in machine data MD24110:

| | |
|---|--|
| Configuration without YM axis: | |
| TRAFO_AXES_IN_1[0]= | Channel axis number of axis radial to rotary axis |
| TRAFO_AXES_IN_1[1]= | Channel axis number of rotary axis |
| TRAFO_AXES_IN_1[2]= | Channel axis number of axis parallel to rotary axis |
| Configuration without existing YM axis: | |
| TRAFO_AXES_IN_1[3] | Channel axis number of axis parallel to peripheral cylinder surface and perpendicular to rotary axis (→ YM axis) |

Machine data specifically for TRACYL

- MD24900 TRACYL_ROT_AX_OFFSET_1

Rotational position: rotary axis setting, when $Y=0$ (in degrees $0 \dots < 360$)

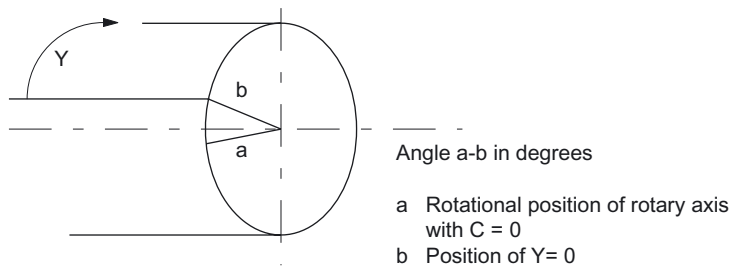


Figure 12-8 Rotational position of axis in the peripheral cylinder surface

- MD24910 TRACYL_ROT_SIGN_IS_PLUS_1

If the rotary axis rotates in an anti-clockwise direction on the X-Y plane opposite to the positive Z axis, then the MD must be set to 1, otherwise to 0.

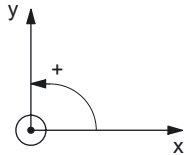


Figure 12-9 Direction of rotation for MD value =1

- MD24920 TRACYL_BASE_TOOL_1

The control is informed of the position of the tool zero point in relation to the origin of the coordinate system declared for TRACYL. The MD has three components for the three axes of the Cartesian coordinate system.

Assignment of axis components in MD24920:

- TRACYL_BASE_TOOL_1[0]=Tx
- TRACYL_BASE_TOOL_1[1]=Ty
- TRACYL_BASE_TOOL_1[2]=Tz (see following figure)

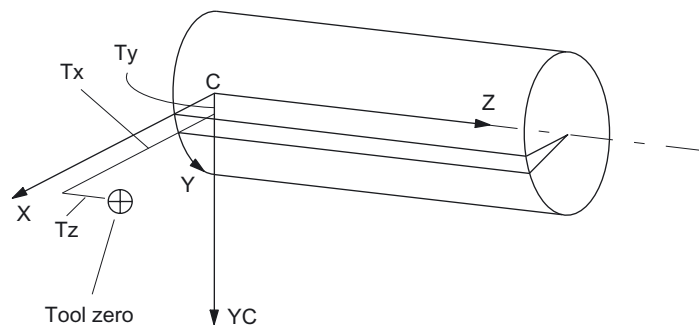


Figure 12-10 Position of tool zero in relation to machine zero

Example: Machine data settings for TRACYL with a standard lathe

General settings: Axis names: XM->X1, ZM->Z1, CM->SP1

- Machine axis name
MD10000 AXCONF_MACHAX_NAME_TAB[0]="X1"
MD10000 AXCONF_MACHAX_NAME_TAB[1]="Z1"
MD10000 AXCONF_MACHAX_NAME_TAB[2]="SP1"
MD10000 AXCONF_MACHAX_NAME_TAB[3]="SP2"
MD10000 AXCONF_MACHAX_NAME_TAB[4]=""
- Assignment of geometry axis to channel axis
MD20050 AXCONF_GEOAX_ASSIGN_TAB[0]=1
MD20050 AXCONF_GEOAX_ASSIGN_TAB[1]=0
MD20050 AXCONF_GEOAX_ASSIGN_TAB[2]=2
- Geometry axis names in channel
MD20060 AXCONF_GEOAX_NAME_TAB[0]="X"
MD20060 AXCONF_GEOAX_NAME_TAB[1]="Y"
MD20060 AXCONF_GEOAX_NAME_TAB[2]="Z"
- Machine axis number valid in channel
MD20070 AXCONF_MACHAX_USED[0]=1
MD20070 AXCONF_MACHAX_USED[1]=2
MD20070 AXCONF_MACHAX_USED[2]=3
MD20070 AXCONF_MACHAX_USED[3]=4
MD20070 AXCONF_MACHAX_USED[4]=0
- Name of channel axis in the channel
MD20080 AXCONF_CHANAX_NAME_TAB[0]="X"
MD20080 AXCONF_CHANAX_NAME_TAB[1]="Z"
MD20080 AXCONF_CHANAX_NAME_TAB[2]="C"
MD20080 AXCONF_CHANAX_NAME_TAB[3]="SP2"
MD20080 AXCONF_CHANAX_NAME_TAB[4]=""
- Initial setting of master spindle in channel
MD20090 SPIND_DEF_MASTER_SPIND=1

TRACYL transformation type for second transformation:

- Without groove wall offset (no YM axis)
MD24100 TRAFO_TYPE_2=512
- Axis assignment in channel
MD24110 TRAFO_AXES_IN_2[0]=1
MD24110 TRAFO_AXES_IN_2[1]=3
MD24110 TRAFO_AXES_IN_2[2]=2
MD24110 TRAFO_AXES_IN_2[3]=0
MD24110 TRAFO_AXES_IN_2[4]=0
- Assignment of geometry axes to channel axes
MD24120 TRAFO_GEOAX_ASSIGN_TAB_2[0]=1
MD24120 TRAFO_GEOAX_ASSIGN_TAB_2[1]=3
MD24120 TRAFO_GEOAX_ASSIGN_TAB_2[2]=2

Special TRACYL settings:

- Offset of rotary axis
MD24800 TRACYL_ROT_AX_OFFSET_1=0
- Sign of rotary axis
MD24810 TRACYL_ROT_SIGN_IS_PLUS_1=1
- Vector of base tool
MD24820 TRACYL_BASE_TOOL_1[0]=0
MD24820 TRACYL_BASE_TOOL_1[1]=0
MD24820 TRACYL_BASE_TOOL_1[2]=0

Setting data for the special treatment of the tool offset (only when required):

- Change of tool length component for change of plane
SD42940 TOOL_LENGTH_CONST=18
- Assignment of the tool length offset independent of tool type
SD42950 TOOL_LENGTH_TYPE=2

Settings for second spindle (milling spindle of the lathe):

- MD30300 IS_ROT_AX[AX4]=1
- MD30310 ROT_IS_MODULO[AX4]=1
- MD30320 DISPLAY_IS_MODULO[AX4]=1

12.3 TRACYL

- MD35000 SPIND_ASSIGN_TO_MACHAX[AX4]=2
- SD43300 ASSIGN_FEED_PER_REV_SOURCE[AX4]=0

Note

A special handling of milling tools on lathes with respect to length compensation is possible.

See Chapter "Tool: Compensation and Monitoring (W1)"

12.3.3 Programming example, TRACYL

Machining grooves with groove wall compensation

MD24100_TRAFO_TYPE_1 = 513

Contour

It is possible to machine a groove which is wider than the tool by using address **OFFN=...** to program the compensation direction (G41, G42) in relation to the programmed reference contour and the distance of the groove side wall from the reference contour.

Tool radius

The tool radius in relation to the groove side wall is automatically taken into account with G41, G42. The full functionality of the plane tool radius compensation is available (steady transition at outer and inner corners as well as solution of bottleneck problems).

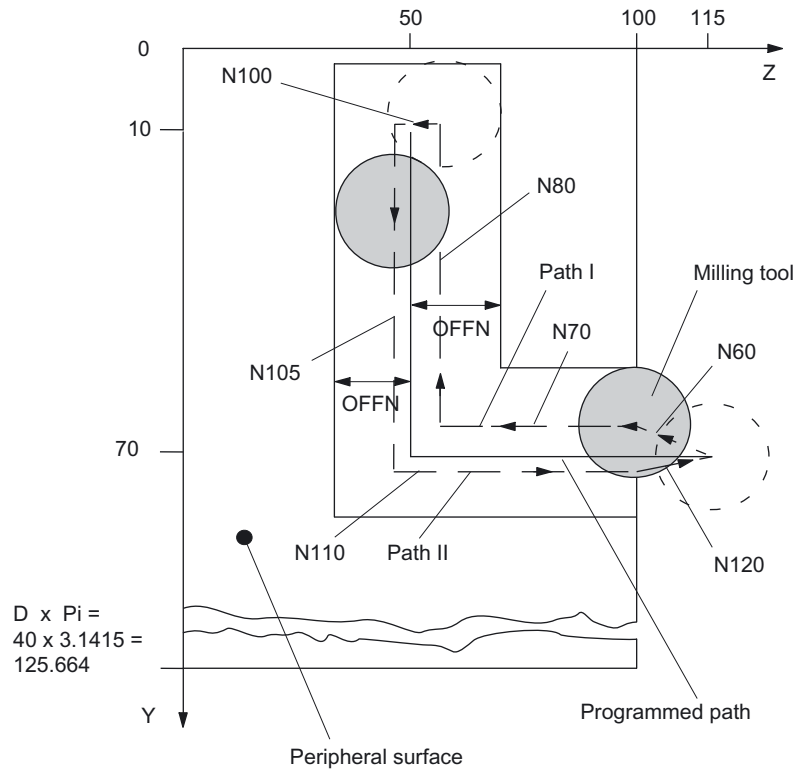


Figure 12-11 Groove with groove wall offset - figure as example

Program example for X-Y-Z-C machine kinematics

TRACYL is used for the milling of grooves on a peripheral cylinder surface. During this process, the "Path I" and "Path II" sections are processed using different OFFN values.

CC is the channel axis name of the rotary axis, milling radius of T1, D1: 8.345 mm

```

N1 SPOS=0 ; Transfer of spindle to position control
; (only for lathes)
N5 T1 D1 ; Tool selection
N10 G500 G0 G64 X50 Y0 Z115 ; Positioning of machine, Y in center of rotation
CC=200 DIAMOF
N20 TRACYL(40) ; Transformation selection, reference diameter
; for surface: 40 mm
N30 G19 G90 G94 G1 F500 ; Machining plane is cylinder surface Y/Z
N40 OFFN=12.35 Y70 Z115 ; Define groove wall offset, starting position,
; Y is now transformer axis
N50 X20 M2=3 S2=300 ; Feed tool to groove base,
; switch on milling spindle

```

```

; Approach of groove wall:
N60 G1 G42 Y70 Z100 ; TRC selection to approach groove wall

; Machining groove section path I:
N70 Z50 ; Groove part parallel to cylinder plane
N80 Y10 ; Groove part parallel to circumference
N90 OFFN=11.5 ; Change groove wall offset

; Machining groove section path II:
N100 G1 G42 Y10 Z50 ; TRC selection to approach groove wall for path II
N105 Y70 ; Groove part parallel to circumference
N110 Z100 ; Revert to initial value

; Retraction from groove wall:
N120 G1 G40 Y70 Z115 ; TRC deselection, retract from groove wall
N130 G0 X25 M2=5 ; Retraction, stop milling spindle
N140 TRAFOOF ; Switch off TRACYL
N150 G0 X50 Y0 Z115 CC=200 ; Return to starting point
OFFN=0
N160 M30

```

12.4 Special features of TRANSMIT and TRACYL

POWER ON/RESET

The system response after POWER ON or RESET (program end) is determined by the settings stored in the following machine data:

- MD20110 RESET_MODE_MASK (access to this MD only in protection level 1/1)
- MD20140 TRAFO_RESET_VALUE (active transformation after RESET).

Please note on selection

- Tool radius compensation must be deselected (G40).
- The frame which was active prior to TRANSMIT / TRACYL is deselected by the control. (G500).
- The control deselects an active working area limit for axes affected by the transformation (WALIMOF).
- Continuous path control and rounding are interrupted.
- An instructed intermediate movement block with chamfer or radius is not inserted.

Please note on deselection

- Tool radius compensation must be deselected (G40).
- Continuous path control and rounding are interrupted.
- An instructed intermediate movement block with chamfer or radius is not inserted.
- Following TRANSMIT / TRACYL deselection, zero offsets (Frame) and all settings used for the turning operation are to be reset.

Operating modes, operating mode changeover

- The program processing with TRANSMIT/ TRACYL is performed in AUTOMATIC.
- It is possible to interrupt the AUTOMATIC operation and change over to JOG. When returning to AUTOMATIC mode, the operator must ensure a problem-free repositioning of the tool.
- Axes cannot be referenced when a transformation is active.

12.5 Data lists**12.5.1 Machine data**

| Number | Identifier | Name |
|------------------|-----------------------------|--|
| Channel-specific | | |
| 20110 | RESET_MODE_MASK | Definition of control basic setting after run-up and RESET/part program end (access only possible at protection level 1/1) |
| 20140 | TRAFO_RESET_VALUE | Initial setting: Transformation active after Reset |
| 22534 | TRAFO_CHANGE_M_CODE | M code for transformation changeover |
| 24100 | TRAFO_TYP_1 | Type of 1st transformation, possibly with axis sequence |
| 24110 | TRAFO_AXES_IN_1 | Axis assignment at input of 1st transformation |
| 24120 | TRAFO_GOEAX_ASSIGN_TAB_1 | Geo-axis assignment for 1st transformation |
| 24200 | TRAFO_TYP_2 | Type of 2nd transformation, possibly with axis sequence |
| 24210 | TRAFO_AXES_IN_2 | Axis assignment at input of 2nd transformation |
| 24220 | TRAFO_GOEAX_ASSIGN_TAB_2 | Geo-axis assignment for 2nd transformation |
| 24800 | TRACYL_ROT_AX_OFFSET_1 | Deviation of rotary axis from zero position in degrees (1st TRACYL) |
| 24810 | TRACYL_ROT_SIGN_IS_PLUS_1 | Sign of rotary axis for TRACYL (1st TRACYL) |
| 24820 | TRACYL_BASE_TOOL_1 | Distance of tool zero point from origin of geo-axes (1st TRACYL) |
| 24900 | TRANSMIT_ROT_AX_OFFSET_1 | Deviation of rotary axis from zero position in degrees (1st TRANSMIT) |
| 24910 | TRANSMIT_ROT_SIGN_IS_PLUS_1 | Sign of rotary axis for TRANSMIT (1st TRANSMIT) |
| 24911 | TRANSMIT_POL_SIDE_FIX_1 | Limitation of working range in front of/behind pole, 1st transformation |
| 24920 | TRANSMIT_BASE_TOOL_1 | Distance of tool zero point from origin of geo-axes (1st TRANSMIT) |

12.5.2 Interface signals

| Number | Bit | Name |
|------------------|-----|-----------------------|
| Channel-specific | | |
| V3800 0001 | .6 | Transformation active |

Measurement (M5)

13.1 Brief description

Channel-specific measuring

A measurement mode is programmed in a part program block (with or without DDTG). A trigger event (edge of the probe) is defined additionally, which will trigger the measurement process. The instructions apply to all axes programmed in this particular block. The program with the measurement process in AUTOMATIC mode is executed and can be employed for workpiece or tool measuring.

Tool measuring in JOG

SINUMERIK 802D includes operator support for the measurement process in JOG mode specially for measuring tools on lathes and milling machines. Channel-specific measuring is integrated into this sequence. The PLC user program must include the required functionality. The measured offset values of the tool are available in the tool offset memory at the end of the measuring sequence.

The exact operating instructions are contained in:

Reference:

/BP_/ Operation and Programming

13.2 Hardware requirements

13.2.1 Probes that can be used

General

In order to measure tool and workpiece dimensions, a touch-trigger probe is required that supplies a constant signal (rather than a pulse) when deflected.

The probe must operate virtually bounce-free. Most sensors can be adjusted mechanically to ensure that they operate in this manner.

Different types of probes supplied by a variety of manufacturers are available on the market. Probes are therefore divided into three groups according to the number of directions in which they can be deflected (see figure below).

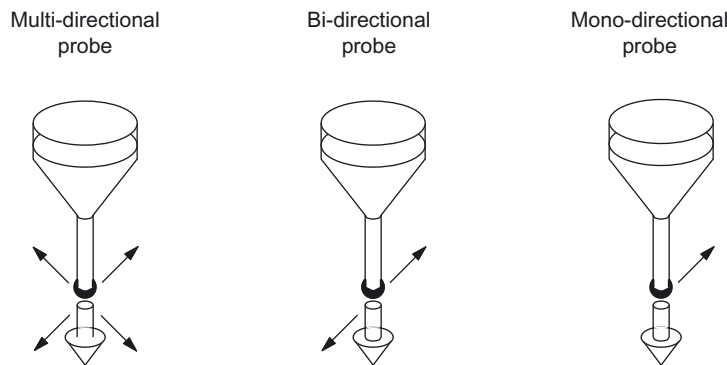


Figure 13-1 Probe types

Table 13- 1 Probe assignment

| Probe type | Lathes | | Milling and machining centers |
|-------------------|-------------------|------------------------|-------------------------------|
| | Tool measurements | Workpiece measurements | Workpiece measurements |
| Multi-directional | X | X | X |
| Bi-directional | – | X | X |
| Mono-directional | – | – | X |

Bi-directional probes must be used on lathes for workpiece measurements, whereas a mono-probe can also be used for this purpose for milling and machining centers.

Multidirectional probe (3D)

This probe type can be used unconditionally for measuring tool and workpiece dimensions.

Bidirectional probe

This probe type is handled in the same way as a mono probe in milling and machining centers. Bi-directional probes can be used to take workpiece measurements on lathes.

Mono-directional probe

This probe type can be used, with only a few restrictions, to take workpiece measurements on milling and machining centers.

The spindle must be capable of being positioned with the **SPOS** NC function if the measurement is to be carried out in different axis directions/axes. The probe must therefore be aligned according to the measuring task.

Switching performance

The signal level of the connected probe (deflected/non-deflected condition) must be communicated to the control via the MD13200 MEAS_PROBE_LOW_ACTIVE[0].

13.2.2 Probe connection

The probe for SINUMERIK 802D sl is connected to the terminals of X20. The particular assignment is determined by the macro you are using. Thus, all measuring inputs of the axis drive modules are operated whose axes are involved in measuring. For the probe, use an external voltage (24 V) whose reference potential must be connected to X20, pin 12.

To ensure optimum interference immunity when connecting probes, shielded lines must be used.

Reference:

/BA/ Instruction Manual, Section "Configuring SINAMICS S120 with 802D sl"

13.3 Channel-specific measuring

13.3.1 Measuring mode

Measuring commands MEAS and MEAW

The measuring operation is activated from the part program. A trigger event and a measuring mode are programmed. Two different measuring modes are available:

- **MEAS:** Measurement with deletion of distance-to-go

Example: N10 G1 F300 X300 Z200 MEAS=-1

Trigger event is the falling edge (-) of probe 1: from deflected to non-deflected status.

- **MEAW:** Measurement without deletion of distance-to-go

Example: N20 G1 F300 X300 Y100 MEAW=1

Trigger event is the rising edge of probe 1: from non-deflected to deflected status.

The measurement block is terminated when the probe signal has arrived or the programmed position has been reached. The measurement job can be cancelled with RESET.

Reference:

/BP_/ Operation and Programming

Note

If a GEO axis (axis in the WCS) is programmed in a measuring block, the measured values are stored for all current GEO axes.

13.3.2 Measurement results

Reading measurement results in the program

The results of the measuring command can be read in the part program via system variables.

- System variable **\$AC_MEA[1]**

Query measurement job status signal.

The variable is deleted at the beginning of a measurement. The variable is set as soon as the probe fulfills the activation criterion (rising or falling edge). Execution of the measurement job can thus be checked in the part program.

- System variable **\$AA_MM[axis]**

Access to measured value in the machine coordinate system (MCS) Read in part program. <axis> stands for the name of the measurement axis (X, Y, ...).

- System variable **\$AA_MW[axis]**

Access to measured value in the workpiece coordinate system. Read in part program. <axis> stands for the name of the measurement axis (X, Y, ...).

References:

/BP_/ Operation and Programming

PLC service display

The measuring signal can be controlled via the "PLC status" menu in the "System → "PLC" operating area:

IS "Probe 1 activated" (V2700 0001.0).

The current measuring status of the axis is shown by the IS "Measurement active" (measuring block with this axis running).

13.4 Measurement accuracy and functional testing

13.4.1 Measuring accuracy

Accuracy

The propagation time of the measuring signal is determined by the hardware used. The delay times are in the μs range plus the probe response time.

The measurement uncertainty is calculated as follows:

Measurement uncertainty = measuring signal propagation time x traversing velocity

Correct results can only be guaranteed for traversing velocity where not more than one triggering signal arrives per position controller cycle.

13.4.2 Probe functional test

Example of functional test

The functional test for the probe is conducted favorably via an NC program.

```

%_N_PRUEF_MESSTASTER_MPF           ;Testing program probe connection
N10 R10                             ;Flag for trigger status
N20 R11 MESSWERT_IN_X
N30 T1 D1                           ; Preselect tool offset for probe
N40 ANF: G0 G90 X0 F150             ;Starting position and meas. velocity
N50 MEAS=1 G1 X100                 ; Measurement at measuring input 1 in the X axis
N60 STOPRE
N70 R10=$AC_MEA[1]                 ; Read switching signal at 1st measuring input
N80 IF R10==0 GOTOF FEHL1           ;Evaluation of signal
N90 R11=$AA_MW[X]                  ;Read in measured value in workpiece coordinates
N95 M0
N100 M2
N110 FEHL1: MSG                     ; Probe not switching!
N120 M0
N130 M2

```

Example of repeat accuracy

This program allows the measuring scatter (repeat accuracy) of the entire measuring system (machine-probe-signal transmission) to be calculated.

In the example, ten measurements are taken in the X axis and the measured value recorded in the workpiece coordinates.

It is possible to determine the so-called "random dimensional deviations" which are not subject to any trend.

```

%_N_CHECK_ACCURATE_MPF
N05 R11                             ; Switching signal
N06 R12=1                            ; Counter
N10 R1 to R10                        ; MEAS_VAL_IN_X
N15 T1 D1                            ; Start conditions, preselect tool offset for
; probe
N20 ANF: G0 X0 F150                 ;Prepositioning in the measured axis
N25 MEAS=+1 G1 X100                 ; Measurement at 1st measuring input with
;rising switching edge, in the X axis
N30 STOPRE                          ; Stop decoding for subsequent evaluation of the
; result (automatically executed when reading
; MEA)
N35 R11= $AC_MEA[1]                 ; Read switching signal at 1st measuring input
N37 IF R11==0 GOTOF FEHL1           ;Check switching signal
N40 R[R12]=$AA_MW[X]                ;Read measured value in workpiece ;coordinates

```

```
N50 R12=R12+1
N60 IF R12<11 GOTOB ANF          ;Repeat 10 times
N65 M0
N70 M02
N80 FEHL1: MSG                  ; Probe not switching
N90 M0
N95 M02
```

The measurement results R1 to R10 can be read after selecting the parameter display.

13.5 Tool measuring in JOG

Measuring principle

The employed tool is traversed to the probe by the user in JOG mode using the traverse keys or handwheel. When the probe switches, the movement is stopped automatically and switched internally to AUTOMATIC mode, and a measuring program is launched. The measuring program controls the real measurement sequence with a second approach of the probe and further positioning. In the end the tool offsets are entered and JOG mode is activated again.

Advantage: The entered offset values before measuring the tool can deviate entirely from the actual values. The tools must not be "pre-measured".

Note

The tool is "re-measured", not its wear.

Softkeys and templates are provided for use by the user in JOG mode. This supports the user during tool measuring.

Reference:

/BP_/ Operation and Programming

| |
|---------------|
| NOTICE |
|---------------|

| |
|---|
| The PLC user program must be created with the necessary sequences. The functionality is not available beforehand. |
|---|

| |
|--|
| Extreme caution must be taken when approaching the probe. The probes only have a limited deflection path. They will be damaged or destroyed if this is exceeded! Observe the manufacturers instructions! |
|--|

| |
|---|
| In particular, the approach speed should be reduced to such an extent that the probe can always be stopped promptly. "Rapid traverse override" may not be active. |
|---|

The screen forms provided and the sequence depend on the technology. Accordingly, the following used tool types can be measured:

Turning technology

- Turning tool (geometry length 1 and length 2)
- Drill (geometry length 1)

Milling technology

- Milling tool (geometry length 1 and geometry radius)
- Drill (geometry length 1)

Tool offsets

The screens initially include the active tool T and the active offset number D for the target of the measurement result entry. A different tool can be specified by the PLC via the interface, or the user can enter a different tool T and/or offset number D.

| NOTICE |
|---|
| If a tool or offset number different to the active values has been entered, this must first be made known to the NC for working after measurements have been made with this tool/tool offset, e.g. by programming and start in MDA mode. Only then can the control unit calculate the correct tool offsets. |

A **tool length compensation** is automatically entered into the GEO component of the active/specified tool offset D of the active/specified tool, and the associated "wear" and "adapter" components are deleted.

When measuring the **cutter radius** it is assumed that no further offset is applied to the axes of the cutter radius level (values in the axes of the "adapter" component and GEO lengths 2 and 3 are equal to zero). The result for the radius is entered in the "geometry" component. The associated "adapter" and "wear" components of both axes of the level are deleted.

Probe

The tool measuring probe is a touch probe at a fixed location or is swiveled into the working area by means of a mechanical device. If the probe plate is of rectangular design, the edges should be aligned parallel to the axis. The tool/calibration tool is traversed against the measuring probe. The probe must be calibrated before a measurement is taken. This means that the precise probe triggering points in relation to the machine zero are known.

Preparation, probe calibration

- Select JOG mode.
- The following values should be entered in the opened window via the "Settings" softkey: return plane, safety clearance, JOG feed, variable increment and direction of rotation of the spindle for general use in JOG and for tool measuring.
- The following value must be entered in the window which opens when pressing the "Probe data" softkey:
 - Feed for automatic probe approach in the measuring program.
 - Probe triggering points (the values are set during calibration).
If the precise values are known, they can be entered manually. The probe does not then need to be calibrated).
- The adjustment sequence of the probe (calibration) is controlled via the "tool measuring" and "tool calibration" softkeys and the opening window. The tool used in this case is the calibration tool with precisely known and entered dimensions.
The calibration tool for the milling technology is of "cutter" type, and for turning it is of "turning tool" type with a cutter radius of zero. The entered cutting edge position is not taken into account in this case.

The internal sequence is the same as in measuring. The measuring results, however, are stored in the data for the probe triggering points - not in the tool offsets.

Note

The internal NC programs for measuring or calibrating are configured so that measuring is carried out with the rising edge of the probe.

Measuring sequence

JOG mode is selected. The measuring feed is entered. The probe is calibrated or the precise measuring trigger points are entered.

- Depending on the tool type, the measuring sequence is controlled via the "tool measuring" softkey and further softkeys.
- The IS "Measuring in JOG is active" (V1700 0003.7) is transmitted to the PLC from the HMI by pressing the "tool measuring" softkey. PLC can specify a different T number to the active one via the IS "T number for tool measuring in JOG" (VD1900 5004). If the probe switches when the selected axis is traversed, NCK outputs the IS "Probe 1 active" (V27000001.0). The PLC then sets IS "Feed disable" (V3200 0006.0) and NCK stops the axis movement. Feed disable is maintained as long as a traverse key is depressed in JOG and the IS "Measuring in JOG is active" (V1700 0003.7) is set. After this the PLC outputs the IS "Reset" (V3000 000.7). The traverse movement in JOG is thereby cancelled.
- HMI recognizes switching of the probe and outputs the change mode to AUTOMATIC, IS command "AUTOMATIC mode" (V1800 0000.0) after the traverse key has been released (immediately after handwheel jog). PLC transfers this to the NCK (V3000 0000.0).
AUTOMATIC mode is set to active by the NCK (IS "Active mode AUTOMATIC" (V3100 0000.0)) and is displayed in the HMI screen. PLC cancels the IS "Feed disable (V3200 0006.0). The HMI then outputs the IS "Mode change disable" (V1800 0000.4) to the PLC.

If the PLC recognizes this signal (is only applied for one PLC cycle), the PLC outputs the IS "Mode change disable " (V3000 0000.4) to the NCK.

An NC measuring program has been loaded to the NCK by the HMI. This is activated now. The automatic direction of approach to the probe and the traverse path including the safety clearance is calculated in this measuring program.

The HMI outputs the command to start the measuring program to the PLC via the IS "Start measuring in JOG" (V1800 0000.6). The signals in the V1800 area are only applied for a single PLC cycle. The IS "Start measuring in JOG" is therefore stored intermediately in the PLC. The NC measuring program is launched by the PLC by outputting the IS "NC START" (V3200 0007.1) to the NCK.

- The axis is repositioned by the NC program, the probe is approached again, and finally retracted. The HMI then transmits the command to switch back to JOG mode (V1800 0000.2) to the PLC. The "Change mode disable" interface signal (V3000 0000.4) is then reset by the PLC. The PLC outputs JOG mode (V3000 0000.2) to the NCK and the NCK returns the IS "JOG mode active" (V3100 0000.2) to the NCK.
- The next direction of approach/axis for traversing to the probe is selected with the "Next step" softkey. The further procedure is analogous - until all directions/axes have been traversed.

After measuring or probe calibration is complete the function can be deselected via the "Back" softkey. This also resets the IS "Measuring in JOG active" (V1700 0003.0). It is also reset when the operating area is exited. The automatic program can be cancelled via IS "Reset" (V3000 0000.7) or measuring in JOG can be closed via the the "Back" softkey. This also cancels any set IS "Feed disable" (V3200 0006.0) and IS "Change mode disable" (V3000 0000.4) or intermediately saved signals.

PLC user program

The required functionality corresponding with the above-described procedure in the PLC user program must be provided by the user.

The toolbox for SINUMERIK 802D sl supplied by SIEMENS includes a user example in the PLC library. You can use this. In this case it should be noted that PLC_INI (SBR32) and MCP_NCK (SBR38) must always be opened in OB1 as these transfer the signals of the MEAS_JOG (SBR43) subroutine to the NCK/HMI.

13.6 Data lists

13.6.1 Machine data

| Number | Identifier | Name |
|---------|-----------------------|------------------------------------|
| General | | |
| 13200 | MEAS_PROBE_LOW_ACTIVE | Switching characteristics of probe |

13.6.2 Interface signals

| Number | Bit | Name |
|--|--------|--|
| HMI signals (from HMI to PLC) | | |
| V1700 0003 | .7 *** | Measuring in JOG active |
| V1800 0000 | .0 | AUTOMATIC mode (request by HMI) |
| V1800 0000 | .1 | MDA mode (request by HMI) |
| V1800 0000 | .2 | JOG mode (request by HMI) |
| V1800 0000 | .4 | Change mode disable (request by HMI) |
| V1800 0000 | .6**** | Start measuring in JOG (request by HMI) |
| V1800 0001 | .2 | REF machine function (request by HMI) |
| HMI signals (from PLC to HMI) | | |
| VD1900 5004 | | Tool number for tool measuring in JOG (input by PLC) |
| General (from NCK to PLC) | | |
| V2700 0001 | .0 | Probe 1 is actuated |
| Axis/spindle-specific (from axis to PLC) | | |
| V390x 0002 | .3 | Measurement active |

EMERGENCY OFF (N2)

14.1 Brief description

| NOTICE |
|---|
| It is the duty of the machine manufacturer to observe national and international standards (see the notes on standards in the following paragraph). The SINUMERIK 802D sl supports the machine manufacturer in the implementation of the EMERGENCY STOP function in accordance with the specifications in this Description of Functions. The responsibility for the EMERGENCY STOP function (its triggering, execution and acknowledgment) rests exclusively with the machine manufacturer. |

Note

Particular reference should be made to the following standards for the EMERGENCY STOP function:

- EN ISO 12100-1
 - EN ISO 12100-2
 - EN 418
 - EN 60204-1
-

EMERGENCY STOP in the control system

The control system supports the machine manufacturer in implementing an EMERGENCY STOP function on the basis of the following features:

- Activation of EMERGENCY STOP sequence in the NC via a PLC input.
- The EMERGENCY STOP procedure in the NC reduces the speed of all axes and spindles as quickly as possible.
- Unlocking of the EMERGENCY STOP button does not reset the EMERGENCY STOP state. Resetting the control device does not restart the machine.
- After the EMERGENCY STOP state has been cancelled, it is not necessary to reference axes or synchronize spindles (positions are corrected).

EMERGENCY STOP pushbutton

The Siemens machine control panel (MCP) for 802D sl is equipped with a mushroomhead pushbutton (emergency stop button with one NC and one NO contact each) referred to below as the EMERGENCY STOP pushbutton.

14.2 EMERGENCY STOP sequence

Requirements

Actuation of the EMERGENCY STOP pushbutton or a signal derived directly from the button must be taken to the control system (PLC) as a PLC input. In the PLC user program, this PLC input must be transferred to IS "EMERGENCY STOP" (V2600 0000.1) in the NC.

Resetting of the EMERGENCY STOP pushbutton or a signal derived directly from the button must be taken to the control system (PLC) as a PLC input. In the PLC user program, this PLC input must be transferred to IS "Acknowledge EMERGENCY STOP" (V2600 0000.2) in the NC.

Sequence in the NC

The predefined (in EN 418) sequence of internal functions implemented to obtain the EMERGENCY STOP state is as follows in the control system:

1. Part program execution is interrupted. All axes and spindles are braked along a braking ramp defined in MD36610 AX_EMERGENCY_STOP_TIME.
2. The IS "802-READY" (V3100 0000.3) is reset.
3. The IS "EMERGENCY STOP active" (V2700 0000.1) is set.
4. Alarm 3000 is set.
5. On expiry of a delay that is set for specific axes/spindles in MD36620 SERVO_DISABLE_DELAY_TIME (shutdown delay, controller enable), the controller enable is cancelled. It must be noted that MD36620 must be specified at least as long as MD36610.

Sequence on the machine

The sequence of EMERGENCY STOP functions on the machine is determined solely by the machine manufacturer. Attention should be paid to the following points in connection to the sequence on the NC:

- The sequence of operations in the NC is started with IS "EMERGENCY STOP" (V2600 0000.1). When the axes and spindles have come to a halt, the power supply must be interrupted, in compliance with EN 418.
- The PLC I/Os (digital outputs) are not affected by the sequence in the NC. If individual outputs are required to attain a particular state in the event of an EMERGENCY STOP, the machine manufacturer must include functions for this purpose in the PLC user program.

| |
|--|
| NOTICE |
| The interruption of the power feed to the equipment is the responsibility of the machine manufacturer. If the internal functions in the NC should not be executed in the predetermined sequence in the event of an EMERGENCY STOP, then IS EMERGENCY STOP (V2600 0000.1) may not be set at any time up to the point that an EMERGENCY STOP defined by the machine manufacture in the PLC user program is reached. As long as the EMERGENCY STOP interface signal has not been set and no other alarm is active, all interface signals are effective in the NC. Any EMERGENCY STOP state defined by the manufacturer can therefore be assumed. |

14.3 EMERGENCY STOP acknowledgment

Acknowledge EMERGENCY STOP

The EMERGENCY STOP state is reset only if IS "Acknowledge EMERGENCY STOP" (V2600 0000.2) followed by IS "Reset" (V3000 0000.7) are set. It must be noted in this respect that IS "Acknowledge EMERGENCY STOP" and IS "Reset" must be set (together) for a long enough period for IS "EMERGENCY STOP active" (V2700 0000.1) to be reset (see Fig. 1–1).

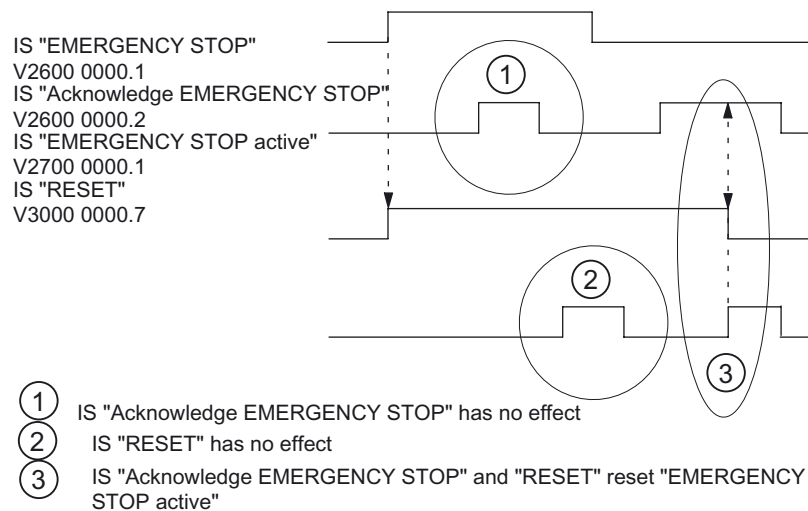


Figure 14-1 Reset emergency stop

Resetting the EMERGENCY STOP state has the following effects:

- IS "EMERGENCY STOP active" is reset.
- The controller enable is switched in.
- IS "Position control active" is set.
- IS "802-READY" is set.
- Alarm 3000 is cleared.
- The part program is aborted.

PLC I/Os

The PLC user program must switch the PLC I/Os to the correct state for operation of the machine.

Reset

The EMERGENCY STOP state cannot be reset solely by IS "Reset" (V3000 0000.7) (see diagram above).

Power off/on

Power off/on (POWER ON) cancels the EMERGENCY OFF state unless IS "EMERGENCY OFF " (V2600 0000.1) is still set.

14.4 Data lists**14.4.1 Machine data**

| Number | Identifier | Name |
|---------------|--------------------------|---|
| Axis-specific | | |
| 36610 | AX_EMERGENCY_STOP_TIME | Length of the braking ramp for error states |
| 36620 | SERVO_DISABLE_DELAY_TIME | Shutdown delay controller enable |

14.4.2 Interface signals

| Number | Bit | Name |
|----------------------------|-----|--|
| General | | |
| V2600 0000 | .0 | Braking on the contour with EMERGENCY STOP |
| V2600 0000 | .1 | EMERGENCY STOP |
| V2600 0000 | .2 | Acknowledge EMERGENCY STOP |
| V2700 0000 | .1 | EMERGENCY STOP active |
| Operating mode signal area | | |
| V3000 0000 | .7 | Reset |

Punching and Nibbling (N4)

Note

This function is only available for the G/N plus and pro versions.

15.1 Brief Description

Subfunctions

The functions specific to punching and nibbling operations comprise the following:

- Stroke control
- Automatic path segmentation
- Rotatable punch and die
- Clamp protection

They are activated and deactivated via language commands.

15.2 Stroke control

15.2.1 General information

Functionality

The stroke control is used in the actual machining of the workpiece. The punch is activated via an NC output signal when the position is reached. The punching unit acknowledges its punching motion with an input signal to the NC. No axis may move within this time period. Repositioning takes place after the punching operation.

High-speed signals

"High-speed signals" are used for direct communication between the NC and punching unit. Combined with the punch, they allow a large number of holes to be punched per minute since the punch positioning times are interpreted as machining delays.

PLC signals

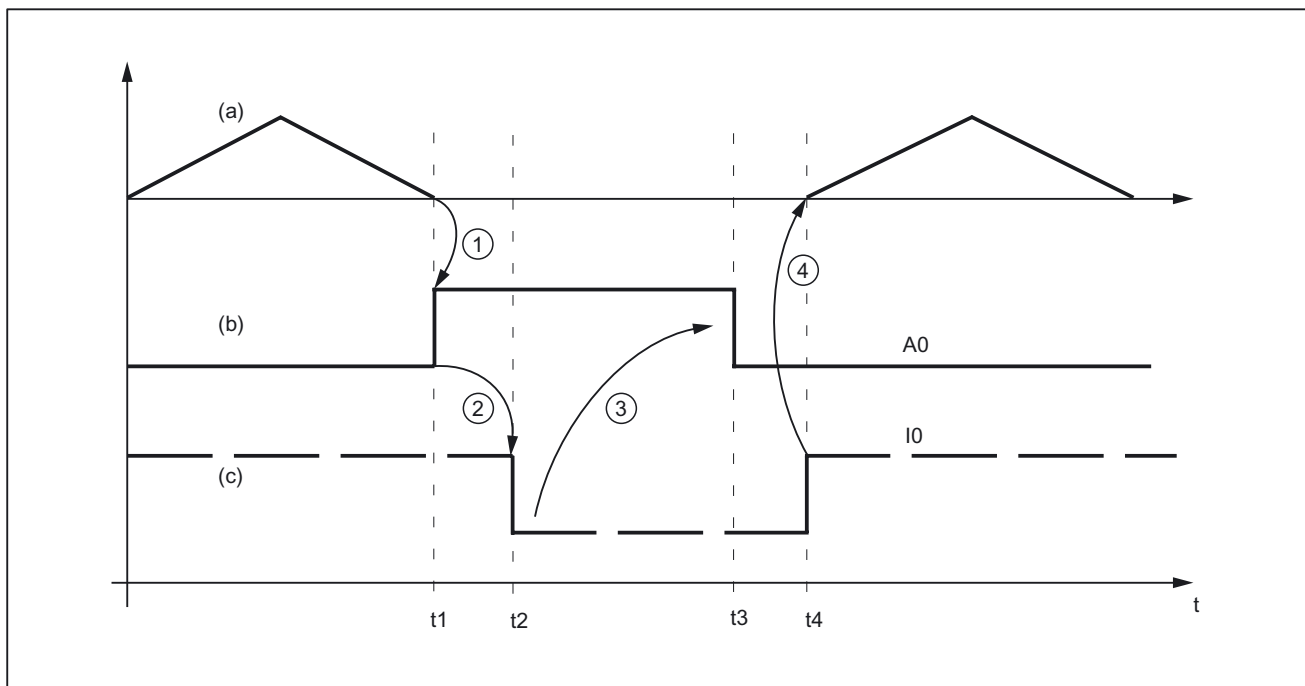
PLC interface signals are used for non-time-critical functions such as enabling and monitoring.

15.2.2 High-speed signals

Functionality

High-speed signals are used to synchronize the NC and punching unit. On the one hand, they are applied via a high-speed output to ensure that the punch stroke is not initiated until the metal sheet is stationary. On the other, they are applied via a high-speed input to ensure that the sheet remains stationary while the punch is active.

The high-speed digital inputs and outputs on the control are used to drive the punching unit. The following signal chart illustrates the signal sequence.



- (a) Axis motion of the machine as function $v(t)$
- (b) "Stroke initiation" signal
- (c) "Stroke active" signal

Figure 15-1 Signal chart

Note

The "Stroke active" signal is high-active for reasons relating to open-circuit monitoring.

The chronological sequence of events for punching and nibbling is controlled by the two signals A_0 and E_0 :

| | |
|-------|--|
| A_0 | Set by the NCK and identical to stroke initiation. |
| E_0 | Defines the status of the punching unit and identical to the "Stroke active" signal. |

The signal states characterize and define times t_1 to t_4 in the following way:

| | |
|-------|---|
| t_1 | The motion of the workpiece (metal sheet) in relation to the punch is completed at instant t_1 . Depending on the criterion defined for stroke initiation (refer to "Criteria for stroke initiation"), high-speed output A_0 is set for punch initiation ①. |
| t_2 | The punching unit signals a punch movement via high-speed input E_0 at instant t_2 . This is triggered by signal A_0 ②. For safety reasons, signal E_0 is high-active (in the case of an open circuit, "Stroke active" is always set and the axes do not move). The "Stroke active" signal is not reset again until the tool has moved away from the metal sheet (t_4). |
| t_3 | The NC reacts to the "Stroke active" signal at instant t_3 by canceling the "Stroke initiation" signal ③. From this point in time onwards, the NC is in a waiting state. It simply waits for cancellation of the "Stroke active" signal so it can initiate the next axis motion. The next stroke can be initiated only after signal A_0 has disappeared. |
| t_4 | The punching operation is complete at instant t_4 , i.e. the punch has exited from the metal sheet again. The NC reacts to a signal transition in signal E_0 by starting an axis motion ④. The reaction of the NC to a signal edge change ④ is described in the section headed "Axis start after punching". |

Note

The stroke time is determined by the period $\Delta t_h = t_4 - t_1$.

Reaction times at instant t_4 between the signal transition of E_0 and the start of the axis motion must also be added.

15.2.3 Criteria for stroke initiation

Initiate a stroke

The stroke initiation must be set, at the earliest, for the point in time at which it can be guaranteed that the axes have reached a standstill. This ensures that at the instant of punching, there is absolutely no relative movement between the punch and the metal sheet in the machining plane.

The following diagram shows the various criteria that can be applied to stroke initiation.

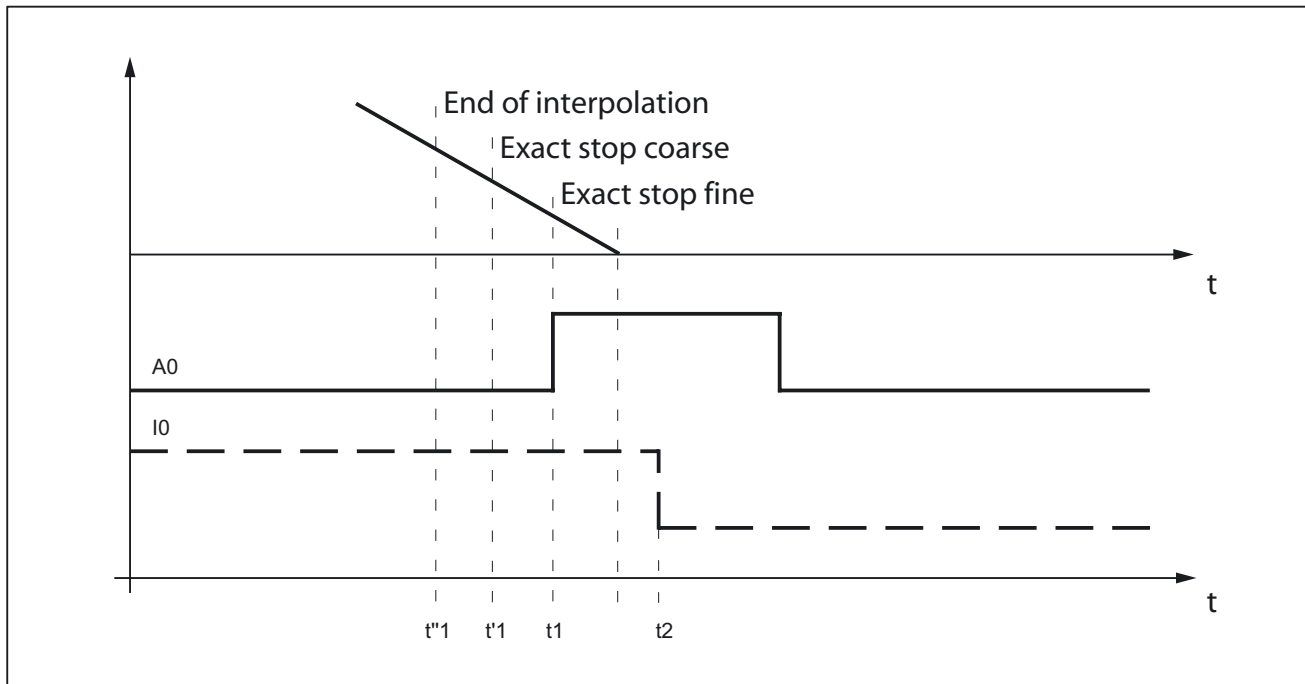


Figure 15-2 Signal chart: Criteria for stroke initiation

The time interval between t_1 and t_2 is determined by the reaction of the punching unit to setting of output A_0 . This cannot be altered, but can be utilized as a lead time for minimizing dead times.

The diagram above shows the default setting with which the output is set when the "Exact stop fine window" is reached (G601; default setting of G group 12). The punch initiation times t''_1 and t'_1 are programmed by means of G602 and G603 (see table below).

| Programming | Activation | Description |
|-------------|-------------------------------------|--|
| G603 | Stop interpolation | The interpolation reaches the block end. In this case, the axes continue to move until the overtravel has been traversed, i.e. the signal is output at an appreciable interval before the axes have reached zero speed (see t''_1). |
| G602 | Reach the coarse in-position window | The signal is output once the axes have reached the coarse in-position window. If this criterion is selected for stroke initiation output, then the instant of stroke initiation can be varied through the size of interpolation window (see t'_1). |
| G601 | Reach the fine in-position window | In this case, it can always be ensured that the machine will have reached a standstill at the instant of punching provided that the axis data is set appropriately. However, this variant also results in a maximum dead time (see t_1). |

G603

Depending on velocity and machine dynamics, approximately 3 - 5 interpolation cycles are processed at the end of interpolation before the axes reach zero speed.

- MD26018 NIBBLE_PRE_START_TIME

In connection with MD26018, it is possible to delay, and therefore optimize, the instant between reaching the end of interpolation and setting the high-speed output for "Stroke ON".

- SD42402 NIBPUNCH_PRE_START_TIME

SD42402 can be changed from the part program and therefore adapted to the punching process depending on the progress of the part program run.

The following applies for the delay time:

MD26018 = 0 → SD42402 is effective

MD26018 ≠ 0 → MD26018 is effective

If the "Punching with dwell time, $P_{DELAYON}$ " is active, then the dwell time programmed in connection with this function is active. Neither MD26018 nor SD42402 is operative in this case.

15.2.4 Axis start after punching

Input signal "Stroke ON"

The start of an axis motion after stroke initiation is controlled via input signal "Stroke ON".

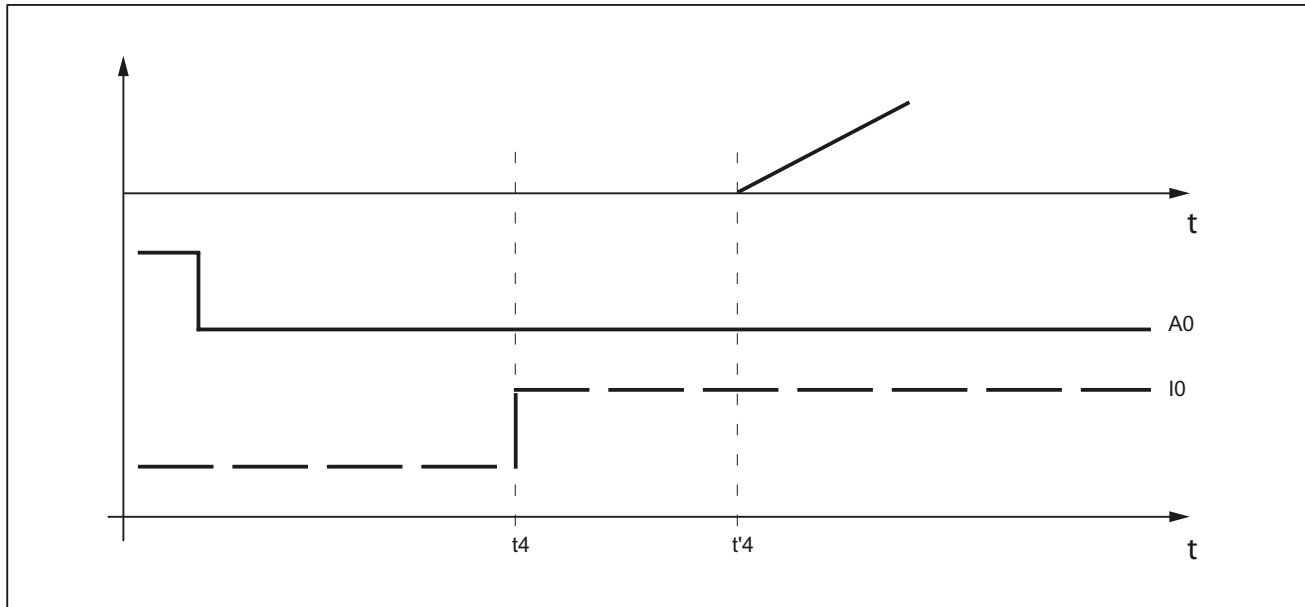


Figure 15-3 Signal chart: Axis start after punching

In this case, the time interval between t_4 and t'_4 acts as a switching-time-dependent reaction time. It is determined by the interpolation sampling time and the programmed punching/nibbling mode.

PON/SON

When the punching unit is controlled via `PON/SON`, the maximum delay time is calculated to be:

$$|t'_4 - t_4| = 3 \times \text{interpolation cycles}$$

PONS/SONS

If the punch is controlled by `PONS/SONS`, the delay time is determined by:

$$|t'_4 - t_4| \leq 3 \times \text{position controller cycles}$$

Prerequisites: Stroke time $(t_4 - t_2) > 4$ interpolation cycles

15.2.5 PLC signals specific to punching and nibbling

Function

In addition to the signals used for direct stroke control, channel-specific PLC interface signals are also available. These are used both to control the punching process and to display operational states.

Signals

| Signal | Activation |
|---|--|
| V3200 0003.0 (stroke enable) | The NC waits until the enable signal is available before continuing the part program. Every punch initiation of the NC is stopped for this time. |
| V3200 0003.2 (stroke suppression) | Allows the part program to be processed without initiating a punching operation (dry run). With active path segmentation, the axes traverse in "Stop and go" mode. |
| V3200 0003.4 (delayed stroke) | Activates delayed stroke output if permitted by PDELAYON. |
| V3200 0003.1 (manual stroke initiation) | Enables the operator to initiate a punching operation (controlled via the PLC) without executing the part program. The manual stroke initiation is acknowledged with IS V3300 0006.1. |

15.2.6 Punching and nibbling-specific reactions to standard PLC signals

Feed stop

With IS "Feed stop" (V3200 1000.3, V3200 1004.3 and V3200 1008.3), the NC reacts as follows with regard to the stroke control:

| | |
|--|---|
| Signal is detected in advance of instant t_1 : | Stroke initiation is suppressed. The next stroke is not initiated until the next start or until the "Feed stop" signal has been cancelled. Machining is then continued as if there had been no interruption. |
| Signal is detected at instant t_1 : | The current stroke is executed to completion. The NC then dwells in the state characterized by t_4 . To allow it to respond in this manner, time monitoring of the "Stroke active" and "Stroke initiation" signals is dispensed with. |

15.2.7 Signal monitoring

Oscillating signal

Owing to aging of the punch hydraulics, overshooting of the punch may cause the "Stroke active" signal to oscillate at the end of a stroke.

In this case, an alarm (22054 "Undefined punching signal") can be generated depending on MD26020 NIBBLE_SIGNAL_CHECK.

Reset response

In the case of an NCK reset, IS V3300 0006.0 (stroke initiation active) is cancelled immediately without acknowledgement by the high-speed input.

A currently activated stroke cannot be suppressed.

15.3 Activation and deactivation

15.3.1 Language commands

Punching and nibbling functions are activated and deactivated via configurable language commands. These replace the special M functions that were used in earlier systems.

Reference:

/BPN/ Operation and Programming, Nibbling

Groups

The language commands are subdivided into the following groups:

| Group 35 | |
|--|---|
| The actual punching and nibbling-specific functions are activated and deactivated by means of the following language commands: | |
| PON | = punching ON |
| SON | = nibbling ON |
| PONS | = punching ON, activated in the position controller |
| SONS | = nibbling ON, activated in the position controller |
| SPOF | = punching/nibbling OFF |

| Group 36 | |
|--|--------------------------|
| This group includes the commands which have only a preparatory character and which determine the real nature of the punching function: | |
| PDELAYON | = punching with delay ON |

| Group 36 | |
|---|---------------------------|
| PDELAYOF | = punching with delay OFF |
| Since the PLC normally needs to perform some preliminary tasks with respect to these preparatory functions, they are programmed before the activating commands. | |

| Group 38 | |
|---|------------------------------|
| This group contains the commands for switching over to a second punch interface. It can be used, for example, for a second punching unit or set of hammer shears. A second I/O pair which can be used for punching functionality is defined via machine data. | |
| SPIF1 | = first interface is active |
| SPIF2 | = second interface is active |

Note

Only one function at a time can be active within a G code group (similar, for example, to the various interpolation modes G0, G1, G2, G3, etc. which are also mutually exclusive).

SPOF

Punching and nibbling OFF

The *SPOF* function terminates all punching and nibbling functions. In this state, the NCK responds neither to the "Stroke active" signal nor to the PLC signals specific to punching and nibbling functions.

If *SPOF* is programmed together with a travel command in one block (and in all further blocks if punching/nibbling is not activated with *SON* or *PON*), the machine approaches the programmed position without the initiation of a punching operation. *SPOF* deselects *SON*, *SONS*, *PON* and *PONS* and corresponds to the Reset condition.

Programming example:

```

:
N20 G90 X100 SON           ; activate nibbling
N25 X50 SPOF              ; deactivate nibbling,
                           ; position without stroke initiation
:

```

SON

Nibbling ON

SON activates the nibbling function and deselects the other functions in G group35 (e.g. *PON*).

In contrast to punching, the first stroke is made at the start point of the block with the activating command, i.e. before the first machine motion.

SON has a modal action, i.e. it remains active until either *SPOF* or *PON* is programmed or until the program end is reached.

The stroke initiation is suppressed in blocks without traversing information relating to the axes designated as punching or nibbling axes (typically those in the active plane). If a stroke still needs to be initiated, then one of the punching/nibbling axes must be programmed with a 0 traversing path. If the first block with SON is a block without traversing information of the type mentioned, then only one stroke takes place in this block since the start and end points are identical.

Programming example:

```
:  
N70 X50 SPOF           ; position without punch initiation  
N80 X100 SON           ; activate nibbling, initiate a stroke before the  
                       ; motion (X=50) and on completion of the programmed  
                       ; movement (X=100)  
:
```

SONS

Nibbling ON (in position control cycle)

SONS behaves in the same way as SON. The function is activated in the position control cycle, thus allowing time-optimized stroke initiation and an increase in the punching rate per minute.

PON

Punching ON

PON activates the punching function and deactivates SON.

PON has a modal action like SON.

In contrast to SON, however, a stroke is not executed until the end of the block or, in the case of automatic path segmentation, at the end of a path segment. PON has an identical action to SON in the case of blocks which contain no traversing information.

Programming example:

```
:  
N100 Y30 SPOF          ; position without punch initiation  
N110 Y100 PON          ; activate punching, punch initiation at the end of  
                       ; positioning operation (Y=100)  
:
```

PONS

Punching ON (in position control cycle)

PONS behaves in the same way as PON. For explanation, please refer to SONS.

PDELAYON

Punching with delay ON

PDELAYON is a preparatory function. This means that PDELAYON is generally programmed before PON. The punch stroke is output with a delay when the programmed end position is reached.

The delay time in seconds is programmed in setting data:

SD42400 PUNCH_DWELLTIME

If the defined value cannot be divided as an integer into the interpolation clock cycle, then it is rounded to the next divisible integer value.

The function has a modal action.

PDELAYOF

Punching with delay OFF

PDELAYOF deactivates punching with delay function, i.e. the punching process continues normally. PDELAYON and PDELAYOF form a G code group.

Programming example:

SPIF2 activates the second punch interface, i.e. the stroke is controlled via the second pair of high-speed I/Os (see machine data MD26004 and MD26006).

```
:  
N170 PDELAYON X100 SPOF           ; position without punch initiation, activate  
                                   ; delayed punch initiation  
N180 X800 PON                     ; activate punching. The punch stroke is output  
                                   ; with a delay when the end position is reached.  
N190 PDELAYOF X700                 ; deactivate punching with delay, normal punch  
                                   ; initiation ON. End of programmed motion  
:  
:
```

SPIF1

Activation of first punch interface

SPIF1 activates the first punch interface, i.e. the stroke is controlled via the first pair of high-speed I/Os (see machine data MD26004 and MD26006).

The first punch interface is always active after a reset or control system power up. If only one interface is used, then it need not be programmed.

SPIF2

Activation of second punch interface

SPIF2 activates the second punch interface, i.e. the stroke is controlled via the second pair of high-speed I/Os (see machine data MD26004 and MD26006).

Programming example:

```

:
N170 SPIF1 X100 PON          ; At the end of the block, a stroke is initiated at
                             ; the first high-speed output. The "Stroke active"
                             ; signal is monitored at the first input.
N180 X800 SPIF2             ; The second stroke is initiated at the second high-
                             ; speed output. The "Stroke active" signal is
                             ; monitored at the second input.
N190 SPIF1 X700             ; All further strokes are controlled with the first
                             ; interface.
:
    
```

15.3.2 Functional expansions

Alternate interface

Machines that alternately use a second punching unit or a comparable medium can be switched over to a second I/O pair.

The second I/O pair can be defined via the following machine data:

MD26004 NIBBLE_PUNCH_OUTMASK

MD26006 NIBBLE_PUNCH_INMASK

The interface is switched by command `SPIF1` or `SPIF2`. Full punching/nibbling functionality is available on both interfaces.

Example: Hardware assignment for stroke control

Define the high-speed byte in each case on the CPU as a high-speed punch interface:

MD26000 PUNCHNIB_ASSIGN_FASTIN = 'H00030001' → Byte 1

MD26002 PUNCHNIB_ASSIGN_FASTOUT = 'H00000001'

Remark: The first and second bits are inverted.

Screen form for high-speed input and output bits:

| | | |
|---------------------------------|-----|--|
| MD26004 NIBBLE_PUNCH_OUTMASK[0] | = 1 | First interface output bit → Bit 1 SPIF1 |
| MD26004 NIBBLE_PUNCH_OUTMASK[1] | = 2 | Second interface output bit → Bit 2 SPIF2 |
| MD26006 NIBBLE_PUNCH_INMASK[0] | = 1 | First interface input bit → Bit 1 SPIF1 |
| MD26006 NIBBLE_PUNCH_INMASK[1] | = 2 | Second interface input bit → Bit 2 SPIF2 |

Automatically activated pre-initiation time

Dead times due to the reaction time of the punching unit can be minimized if the stroke can be initiated before the interpolation window of the axes is reached. The reference time for this is the interpolation end. The stroke is automatically initiated with G603 and delayed by the set value in relation to the time that the end of interpolation is reached.

The delay time for stroke initiation can be adjusted in machine data:

MD26018 NIBBLE_PRE_START_TIME

Example:

With an IPO cycle of 9 ms, a stroke shall be released two cycles after reaching the interpolation end:

⇒ MD26018 NIBBLE_PRE_START_TIME = 0,018 [s]

A pre-initiation time can also be programmed in setting data:

SD42402 NIBPUNCH_PRE_START_TIME

This setting takes effective only if MD26018 = 0 has been set.

Monitoring of the input signal

If the "stroke active" signal is fluctuating between strokes due to plunger overshoots, for example, the message "undefined punching signal" can be also be output when interpolation is stopped.

The message output is dependent on the setting in machine data:

MD26020 NIBBLE_SIGNAL_CHECK

| | |
|-------------|----------|
| MD26020 = 0 | No alarm |
| MD26020 = 1 | Alarm |

Minimum period between two strokes

A minimum time interval between two consecutive strokes can be programmed in setting data:

SD42404 MINTIME_BETWEEN_STROKES

Example:

There must be a minimum delay of at least 1.3 seconds between two stroke initiations irrespective of physical distance:

⇒ SD42404 MINTIME_BETWEEN_STROKES = 1.3 [s]

If a punching dwell time (PDELAYON) is also programmed, then the two times are applied additively.

If a pre-initiation time at G603 is programmed, it will be effective only if the end of interpolation is reached before the time set in SD 42404:

The programmed time becomes operative immediately. Depending on the size of the block buffer, strokes that have already been programmed can be executed with this minimum interval. The following programming measures (example) can be taken to prevent this:

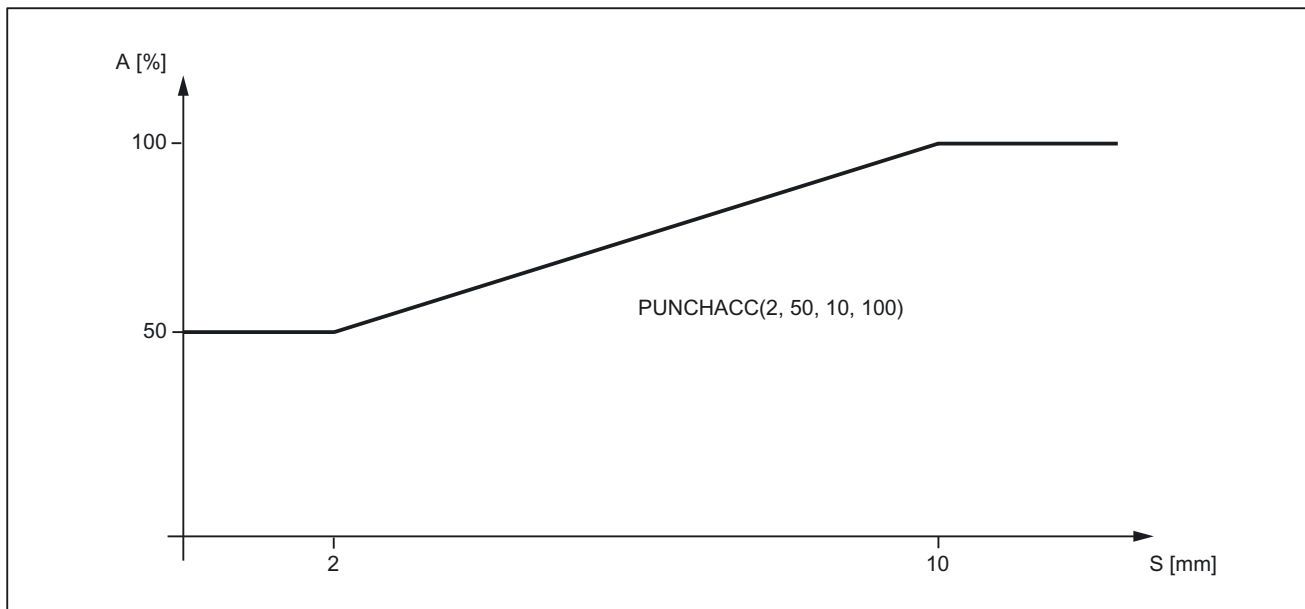
```
N...
N100 STOPRE
N110 $SC_MINTIME_BETWEEN_STROKES = 1.3
```

The function is not active when SD42404 = 0.

Travel-dependent acceleration

An acceleration characteristic can be defined with PUNCHACC (Smin, Amin, Smax, Amax). This command can be used to define different acceleration rates depending on the distance between holes.

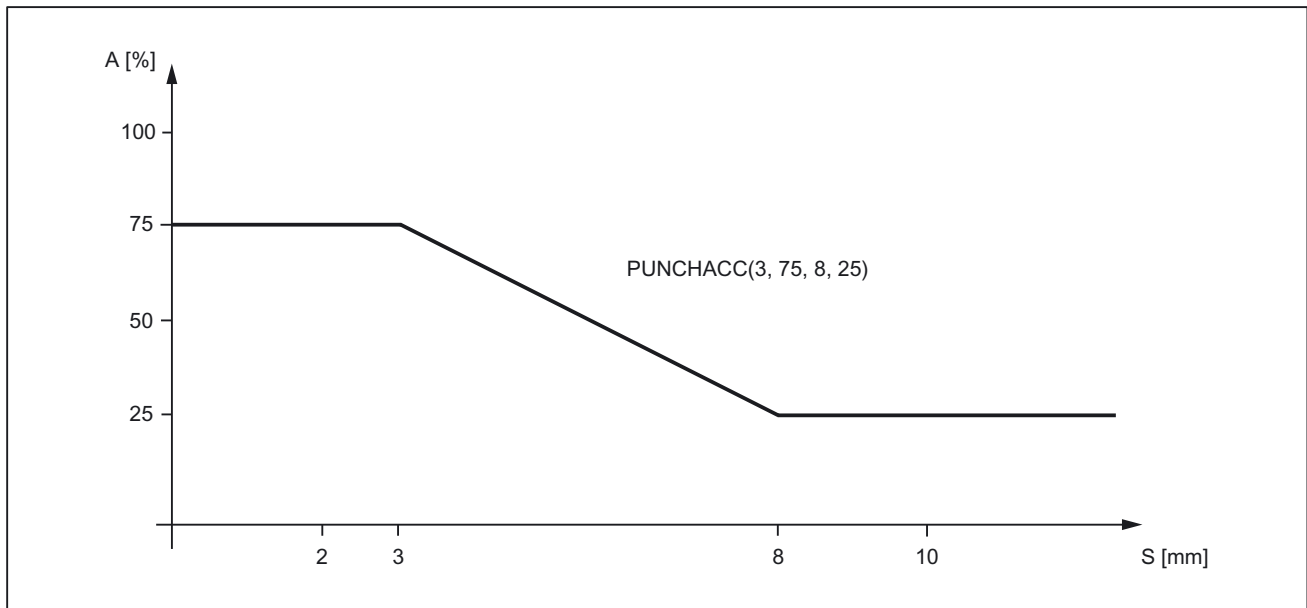
Example 1



The characteristic defines the following acceleration rates:

| Distance between holes | Acceleration |
|------------------------|--|
| < 2 mm | The axis accelerates at a rate corresponding to 50% of maximum acceleration. |
| 2 - 10 mm | Acceleration is increased to 100%, proportional to the spacing. |
| > 10 mm | The axis accelerates at the maximum rate (100%). |

Example 2



The characteristic defines the following acceleration rates:

| Distance between holes | Acceleration |
|------------------------|--|
| < 3 mm | The axis accelerates at a rate corresponding to 75% of maximum acceleration. |
| 3 - 8 mm | Acceleration is reduced to 25%, proportional to the spacing. |
| > 10 mm | The axis accelerates at the maximum rate (25%). |

If a reduced acceleration rate has already been programmed via `ACC`, then the acceleration limits defined with `PUNCHACC` refer to the reduced acceleration rate.

The function is deselected with:

$$S_{\min} = S_{\max} = 0$$

The acceleration rate programmed beforehand with `ACC` remains operative.

Block search

In the case of a search for a block containing a nibbling function, it is possible to program whether the punch stroke is executed at the block beginning or suppressed.

The setting is programmed in machine data:

MD11450 \$MN_SEARCH_RUN_MODE

| Bit | Value | Meaning |
|-----|-------|---|
| 5 | 0 | Punch stroke at beginning of block is suppressed. |
| | 1 | Punch stroke at beginning of block is executed. |

See Section "Block search: Processing of certain program sections (K1)"

15.4 Automatic path segmentation

15.4.1 General information

Function

One of the following two methods can be applied to automatically segment a programmed traversing path:

- Path segmentation with maximum path segment programmed via language command `SPP`
- Path segmentation with a number of segments programmed via language command `SPN`

Both functions generate sub-blocks independently.

In earlier systems

- `SPP<number>` corresponds to `E<number>`
- `SPN<number>` corresponds to `H<number>`

Since addresses `E` and `H` now represent auxiliary functions, language commands `SPP` and `SPN` are used to avoid conflicts. The new procedure is therefore not compatible with those implemented in earlier systems. Both language commands (`SPP` and `SPN`) can be configured.

Note

The values programmed with `SPP` are either mm or inch settings depending on the initial setting (analogous to axes).

The automatic path segmentation function ensures that the path is divided into equidistant sections with linear and circular interpolation.

When the program is interrupted and automatic path segmentation is active (`SPP/SPN`), the contour can be reentered only at the beginning of the segmented block. The first punch stroke is executed at the end of this sub-block.

`SPP` and `SPN` can be activated only if geometry axes are configured.

SPP

The automatic path segmentation function `SPP` divides the programmed traversing path into sections of equal size according to the segment specification.

The following conditions apply:

- Path segmentation is active only when `SON` or `PON` is active.
(Exception: MD26014 PUNCH_PATH_SPLITTING = 1)
- `SPP` is modally active, i.e. the programmed segment remains valid until it is programmed again, but it can be suppressed on a block-by-block (non-modal) basis by means of `SPN`.
- The path segments are rounded off by the control system if required so that a total programmed distance can be divided into an integral number of path sections.

- The path segment unit is either mm/stroke or inch/stroke (depending on axis settings).
- If the programmed `SPP` value is greater than the traversing distance, then the axis is positioned on the programmed end position without path segmentation.
- `SPP = 0`, reset or program end delete the programmed `SPP` value. The `SPP` value is not deleted when punching/nibbling is deactivated.

SPN

The automatic path segmentation function `SPN` divides the traversing path into the programmed number of path segments.

The following conditions apply:

- Path segmentation is active only when `SON` or `PON` is active.
(Exception: MD26014 `PUNCH_PATH_SPLITTING = 1`)
- `SPN` has a non-modal action.
- Any previously programmed `SPP` value is suppressed for the block containing `SPN`, but is re-activated again in the following blocks.

Supplementary conditions

- The path segmentation function is operative with linear and circular interpolation.
The interpolation mode remains unchanged, i.e. circles are traversed when circular interpolation is selected.
- If a block contains both `SPN` (number of strokes) and `SPP` (stroke path), then the number of blocks is activated in the current block while the stroke path is activated in all blocks that follow.
- Path segmentation is active only in conjunction with punching or nibbling functions.
(Exception: MD26014 `PUNCH_PATH_SPLITTING = 1`).
- Any programmed auxiliary functions are output before, during the first or after the last sub-block.
- In the case of blocks without traversing information, the rules which govern the programming of `SON` and `PON` also apply to `SPP` and `SPN`. In other words, a stroke is initiated only if an axis motion has been programmed.

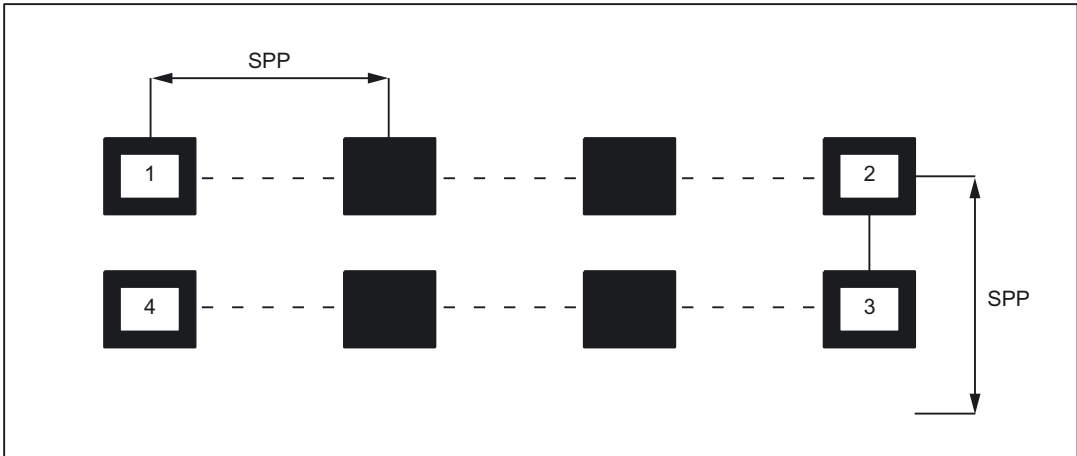
15.4.2 Operating characteristics with path axes

MD26010

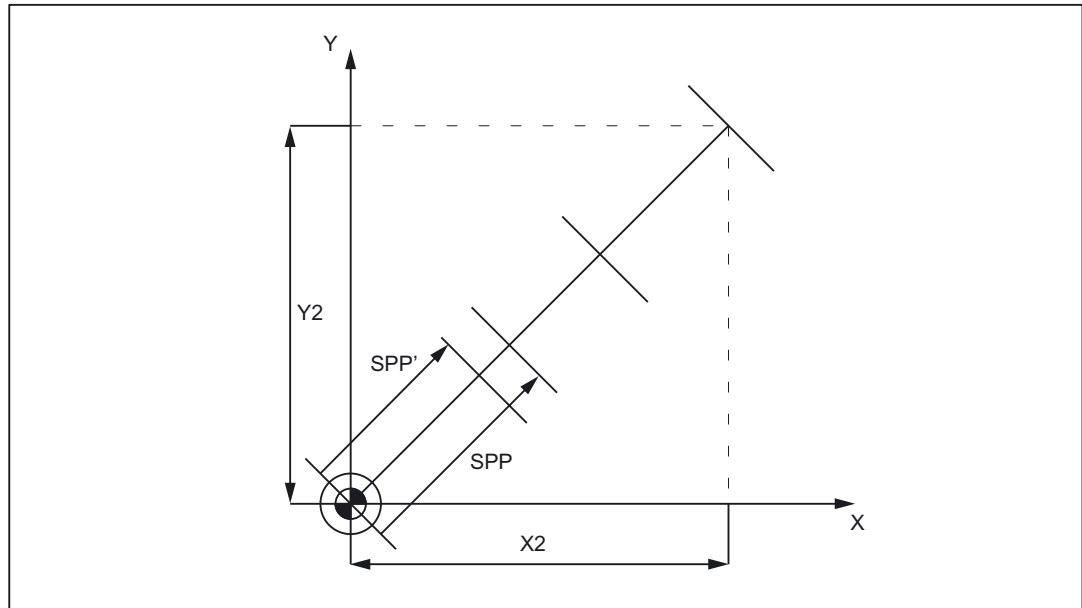
All axes defined and programmed via machine data MD26010 `PUNCHNIB_AXIS_MASK` are traversed along path sections of identical size with `SPP` and `SPN` until the programmed end point is reached. This also applies to rotatable tool axes if programmed. The response can be adjusted for single axes.

Example of SPP

```
N1 G01 X0 Y0 SPOF          ; Position without punch initiation
N2 X75 SPP=25 SON          ; Nibble with feed value 25 mm; initiate punch before
                           ; the first movement and after each path segment.
:
:
N3 Y10                    ; Position with reduced SPP value, because traversing
                           ; distance < SPP value, and initiate punch after
                           ; movement.
:
:
N4 X0                      ; Reposition with punch initiation after each path
                           ; segment.
:
```



If the programmed path segmentation is not an integral multiple of the total path, then the feed path is reduced.



X2/Y2: Programmed traversing distance
SPP: Programmed SPP value
SPP': Automatically rounded-off offset distance

Figure 15-4 Path segmentation

Example of SPN

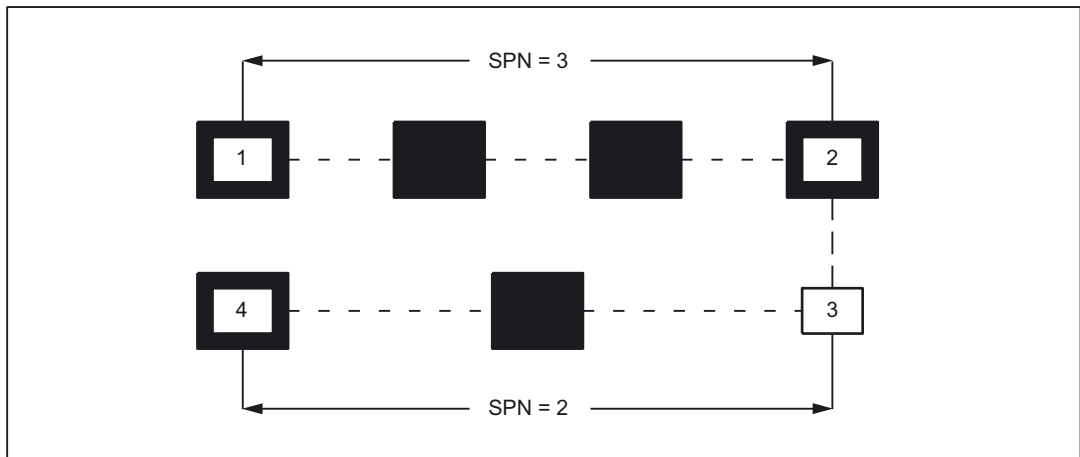
The number of path segments per block is programmed via *SPN*.

A value programmed via *SPN* takes effect on a non-modal basis for both punching and nibbling applications. The only difference between the two modes is with respect to the first stroke. This is normally executed at the beginning of the first segment with nibbling operations and at its end with punching operations. This means that when *n* segments are programmed, *n* strokes are executed with punching operations but *n*+1 with nibbling. Furthermore, where no traversing information is available, only a single stroke is executed, even if several are programmed. Should it be necessary to generate several strokes at one position, then the corresponding number of blocks without traversing information must be programmed.

```

N1 G01 X0 Y0 SPOF                    ; Position without punch initiation
N2 X75 SPN=3 SON                    ; Activate nibbling. The total path is divided into
                                     ; three segments. A stroke is initiated before the
                                     ; first movement and at the end of each segment.
...
N3 Y10 SPOF                         ; Position without punch initiation
N4 X0 SPN=2 PON                     ; Activate punching. The total path is divided into two
                                     ; segments. Since punching is active, the first stroke
                                     ; is initiated at the end of the first segment.
...

```



Example

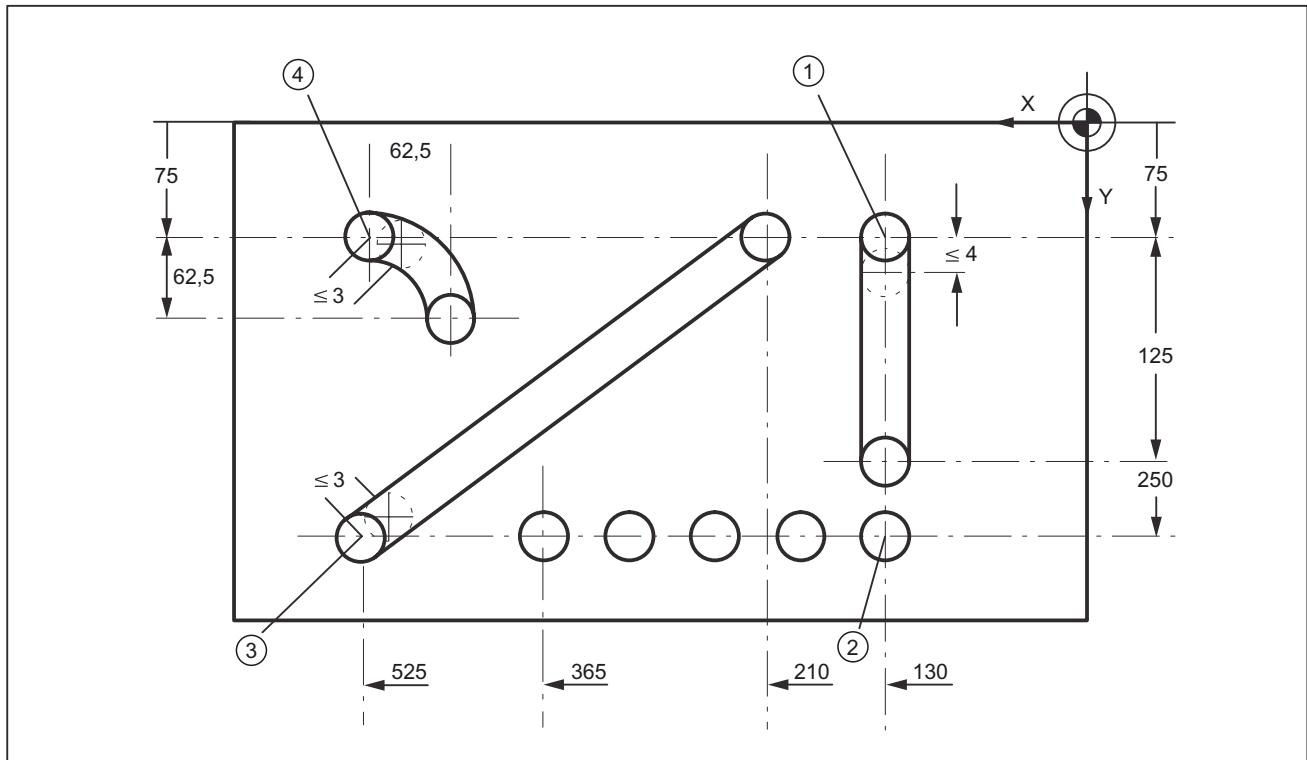


Figure 15-5 Workpiece

Extract from program

```
N100 G90 X130 Y75 F60 SPOF ; Position at starting point ① of
                               ; vertical nibbling path sections
N110 G91 Y125 SPP=4 SON      ; End point coordinates (incremental);
                               ; path segment: 4 mm, activate nibbling
N120 G90 Y250 SPOF          ; Absolute dimensioning, position at
                               ; starting point ② of horizontal
                               ; nibbling path section
N130 X365 SPN=4 SON          ; End point coordinates, four segments,
                               ; activate nibbling
N140 X525 SPOF              ; Position at starting point ③ of
                               ; oblique nibbling path section
N150 X210 Y75 SPP=3 SON     ; End point coordinates path segment: 3
                               ; mm, activate nibbling
N140 X525 SPOF              ; Position at starting point ④ of
                               ; nibbling section on pitch circle path
N170 G02 G91 X-62.5 Y62.5 I0 J62.5 SON ; Incremental circular interpolation
                               ; with interpolation parameters,
                               ; activate nibbling
N180 G00 G90 Y300 SPOF      ; Position
```

15.4.3 Response in connection with single axes

The path of single axes programmed in addition to path axes is distributed evenly among the generated intermediate blocks as standard.

In the following example, the additional rotary axis C is defined as a synchronous axis.

Example

The axis is also defined as "punching and nibbling axis":
MD26010 PUNCHNIB_AXIS_MASK = 1

Various responses of the synchronized axis can be generated depending on the machine data MD26016 PUNCH_PARTITION_TYPE.

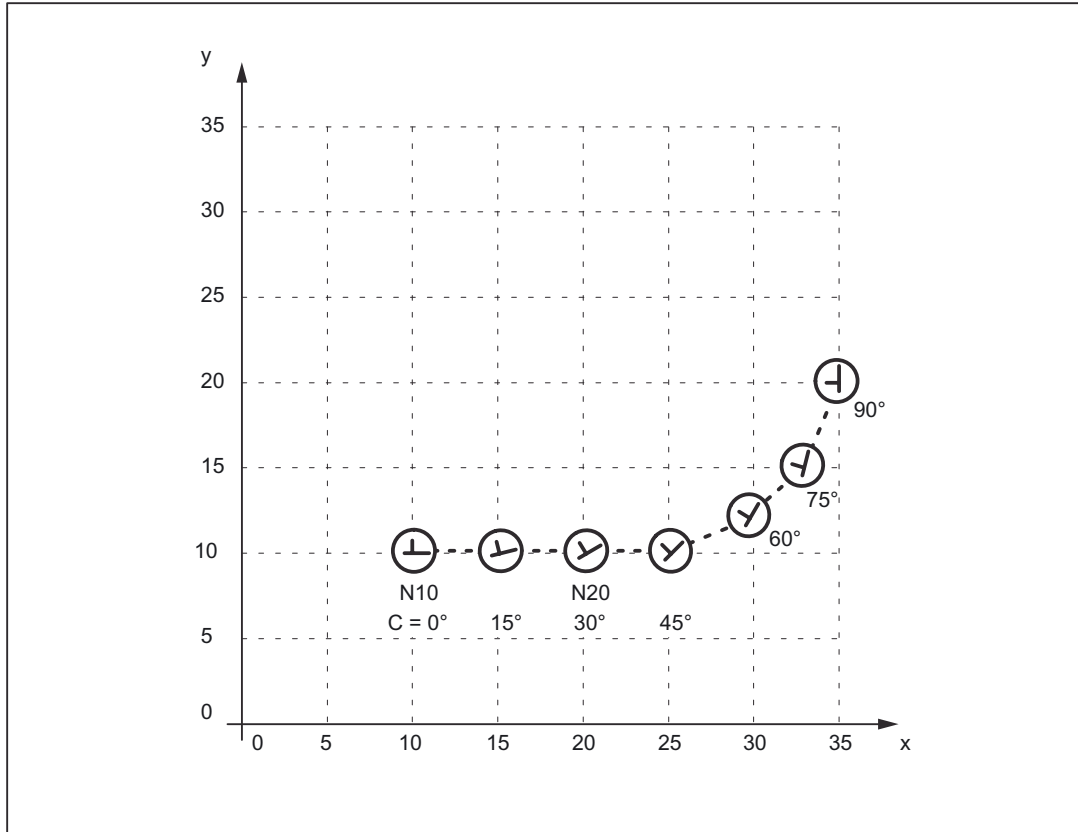
Programming example:

```
N10 G90 G1 PON X10 Y10 C0 F10000
N20 SPP=5 X25 C45
N30 G3 SPN=3 X35 Y20 I0 J10 C90
```

MD26016 PUNCH_PARTITION_TYPE = 0 (default setting)

With this setting, the axes behave as standard, i.e. the programmed special axis motions are distributed among the generated intermediate blocks of the active path segmentation function in all interpolation modes.

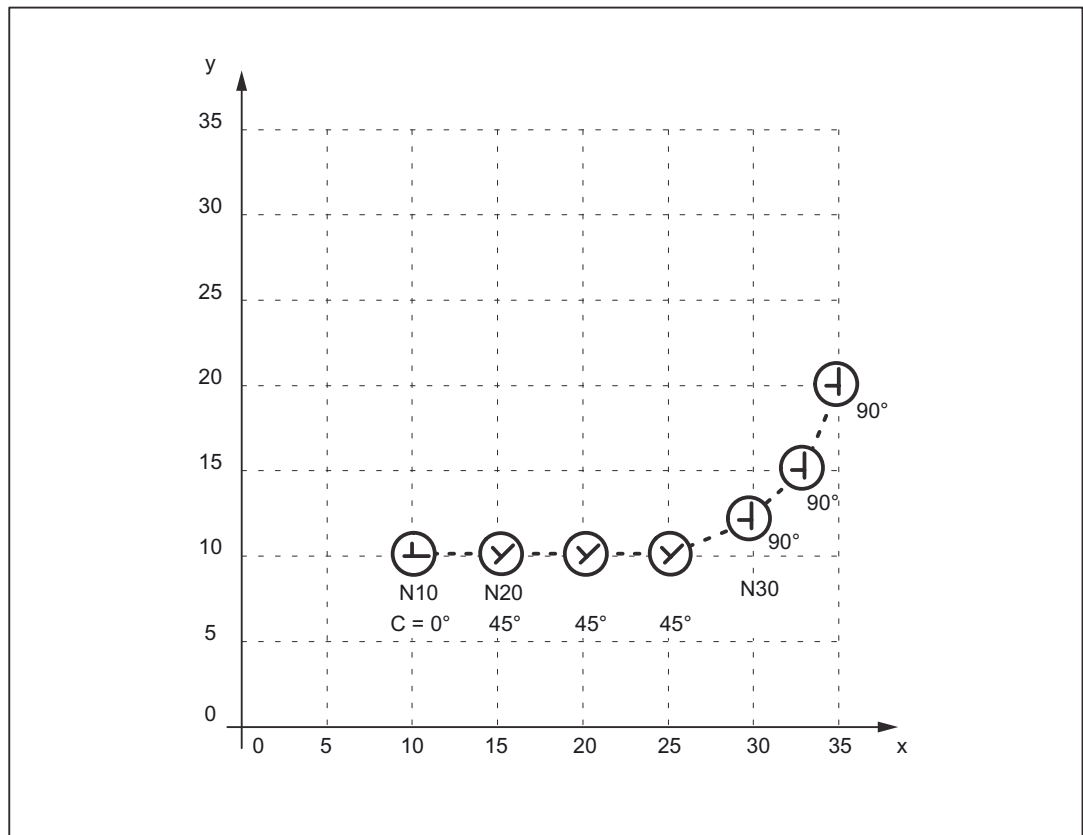
In block N20, the C axis is rotated through 15° in each of the three intermediate blocks. The axis response is the same in block N30, in the case of circular interpolation (three sub-blocks, each with 15° axis rotation).



MD26016 PUNCH_PARTITION_TYPE = 1

In contrast to the behavior described above, here the synchronous axis traverses the entire programmed rotation path in the first sub-block of the selected path segmentation function.

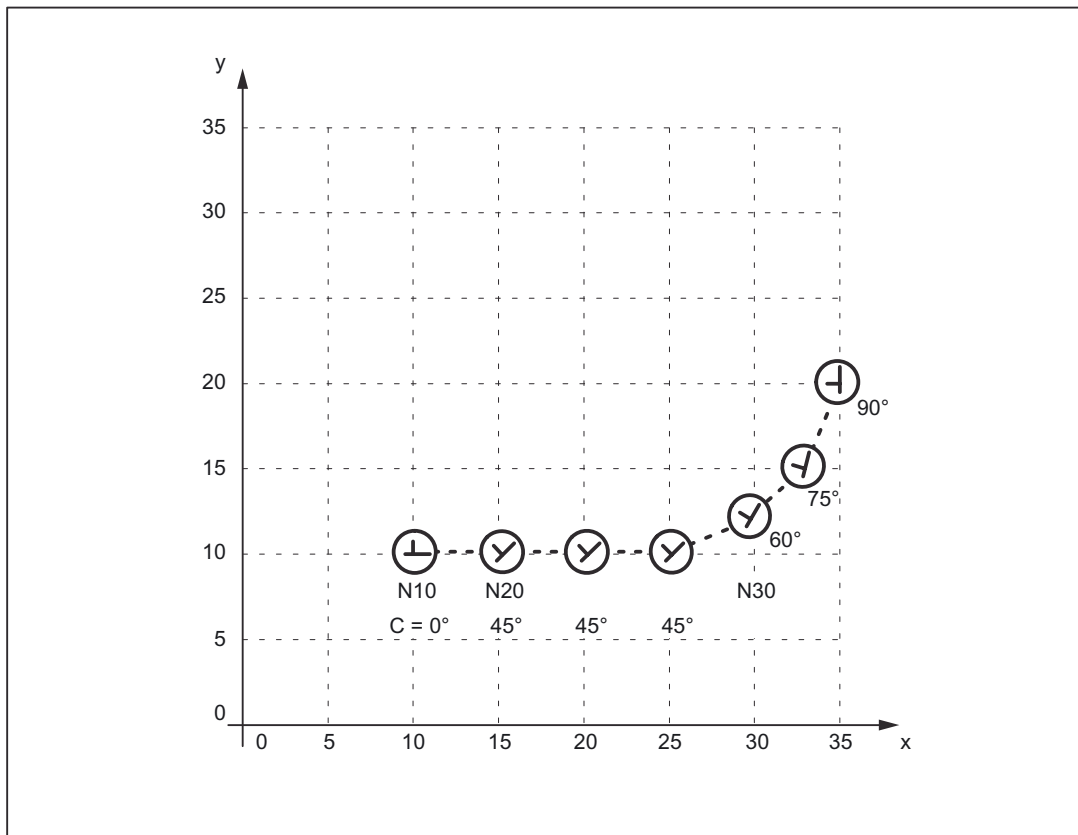
Applied to the example, the C axis already reaches the programmed end position C = 45 when it reaches X position X = 15. It behaves in the same way in the circular interpolation block below.



MD26016 PUNCH_PARTITION_TYPE = 2

MD26016 = 2 is set in cases where the axis must behave as described above in linear interpolation mode, but according to the default setting in circular interpolation mode (see 1st case).

The axis behavior for the example is then as follows: In block N20, the C axis is rotated to C = 45° in the first sub-block. The following circular interpolation block rotates the C axis through 15° in every sub-block.



The axis response illustrated in the diagram above can be particularly useful when applied to the axis of a rotatable tool in cases where it is used to place the tool in a defined direction (e.g. tangential) in relation to the contour, but where the tangential control function **must not** be applied. However, it is not a substitute for the tangential control function since the start and end positions of the rotary axis must always be programmed.

Note

Additional offset motions of special axes (in this case, rotary axis C) are implemented via a zero offset.

Supplementary conditions

- If the C axis is not defined as a "Punch-nibble axis", then the C axis motion path is not segmented in block N30 in the above example nor is a stroke initiated at the block end.
- If the functionality described above is to be implemented in a variant not specific to nibbling applications, but with alignment of the special axis, then stroke initiation can be suppressed by the following PLC interface signal:

V3200 0003.2 (stroke suppression)

(Application: e.g. alignment of electron beam during welding)

A similar response can be programmed with the following machine data setting:

MD26014 PUNCH_PATH_SPLITTING = 1

In this case, the path is segmented irrespective of punching or nibbling functions.

15.5 Rotatable tool

15.5.1 General information

Function overview

The following two functions are provided for nibbling/punching machines with rotatable punch and lower die:

- Coupled motion
for synchronous rotation of punch and die
- Tangential control
for normal alignment of rotary axes for punches in relation to workpiece

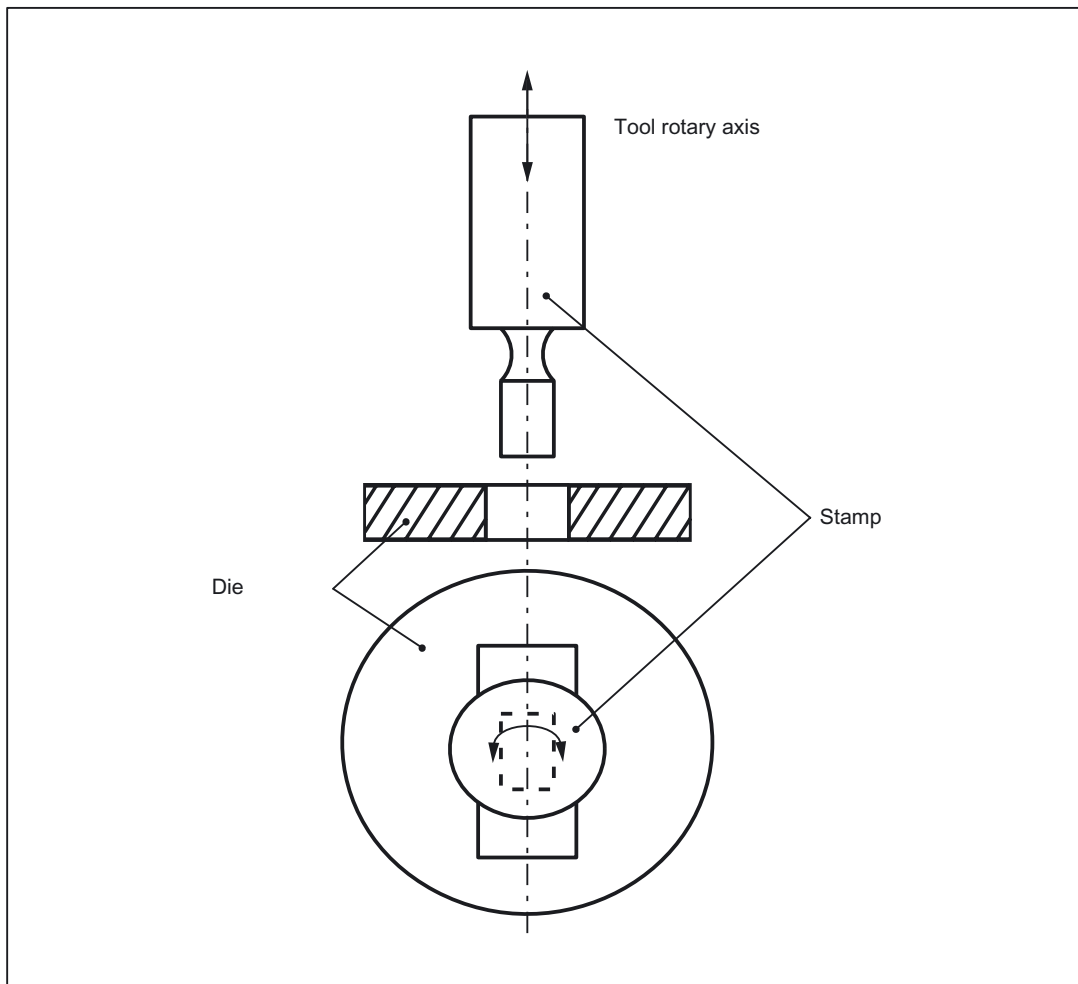


Figure 15-6 Illustration of a rotatable tool axis

15.5.2 Coupled motion of punch and die

Function

Using the standard function "Coupled motion", it is possible to assign the axis of the die as a coupled motion axis to the rotary axis of the punch.

Activation

The "Coupled motion" function is activated or deactivated with language commands `TRAILON` and `TRAILOF` respectively.

Reference:

/BPN/ Operation and Programming, Nibbling

Example

Example of a typical nibbling machine with rotatable punches where C is the punch axis and C1 the die axis:

```
:  
TRAILON (C1, C, 1)      ; Enabling the coupled-motion grouping  
G01 X100 Y100 C0 PON   ; Initiate stroke with C axis/C1 axis position C=0=C1  
X150 C45                ; Initiate stroke with C axis/C1 axis position C=45=C1  
:  
M30
```

Basic position

No coupled-motion groupings are active after power up. Once the two tool axes have approached the reference point, the coupled-motion grouping is not generally separated again.

15.5.3 Tangential control

Function

The rotary tool axes on punching/nibbling machines are aligned tangentially to the programmed path of the master axes by means of the "Tangential control" function.

Activation

The "Tangential control" function is activated and deactivated with language commands `TANGON` and `TANGOF` respectively.

Reference:

/BPN/ Operation and Programming, Nibbling

Mode of operation

The tangential axis is coupled to the interpolation of the master axes. It is therefore not possible to position the axis at the appropriate punching position tangentially to the path independently of velocity. This may lead to a reduction in machining velocity if the dynamics of the rotary axis are unfavorable in relation to those of the master axes. Additional offset angles can be programmed directly via language command TANGON.

Note

If the tool (punch and die) is positioned by two separate drives, then the functions "Tangential control" and "Coupled motion" can be used.

Notice: The "Tangential control" function must be activated first followed by "Coupled motion".

The tangential control function automatically aligns the punch vertically to the direction vector of the programmed path. The tangential tool is positioned before the first punching operation is executed along the programmed path. The tangential angle is always referred to the positive X axis. A programmed additional angle is added to the calculated angle.

The tangential control function can be used in the linear and circular interpolation modes.

Example: Linear interpolation

The punching/nibbling machine has a rotatable punch and die with separate drives.

Programming example:

```

:
N2 TANG (C, X, Y, 1, "B") ; Define master and slave axes, C is slave axis for X
                           and Y in the base coordinate system
N5 G0 X10 Y5 ; Start position
N8 TRAILON (C1, C, 1) ; Activate coupled motion of rotatable tool axes C/C1
N10 Y10 C225 PON F60 ; C/C1 axis rotates to 225° Rstroke
N15 X20 Y20 C45 ; C/C1 axis rotates to 45° Rstroke
N20 X50 Y20 C90 SPOF ; C/C1 axis rotates to 90°, no stroke initiation
N25 X80 Y20 SPP=10 SON ; Path segmentation: four strokes are executed with
                           tool rotated to 90°
N30 X60 Y40 SPOF ; Position
N32 TANGON (C, 180) ; Activate tangential control, offset angle of
                           rotatable tool axes 180°
N35 X30 Y70 SPN=3 PON ; Path segmentation, three strokes with active
                           tangential control and an offset angle of 180°
N40 G91 C45 X-10 Y-10 ; C/C1 rotates to 225° (180° + 45° INC), tangential
                           control deactivated because no path segmentation R
                           stroke
N42 TANGON (C, 0) ; Tangential control without offset
N45 G90 Y30 SPN=3 SON ; Path segmentation, three strokes with active
                           tangential control but without offset angle
N50 SPOF TANGOF ; Deactivate stroke initiation + tangential control
N55 TRAILOF (C1, C) ; Deactivate coupled motion of rotatable tool axes C/C1
N60 M2
:

```

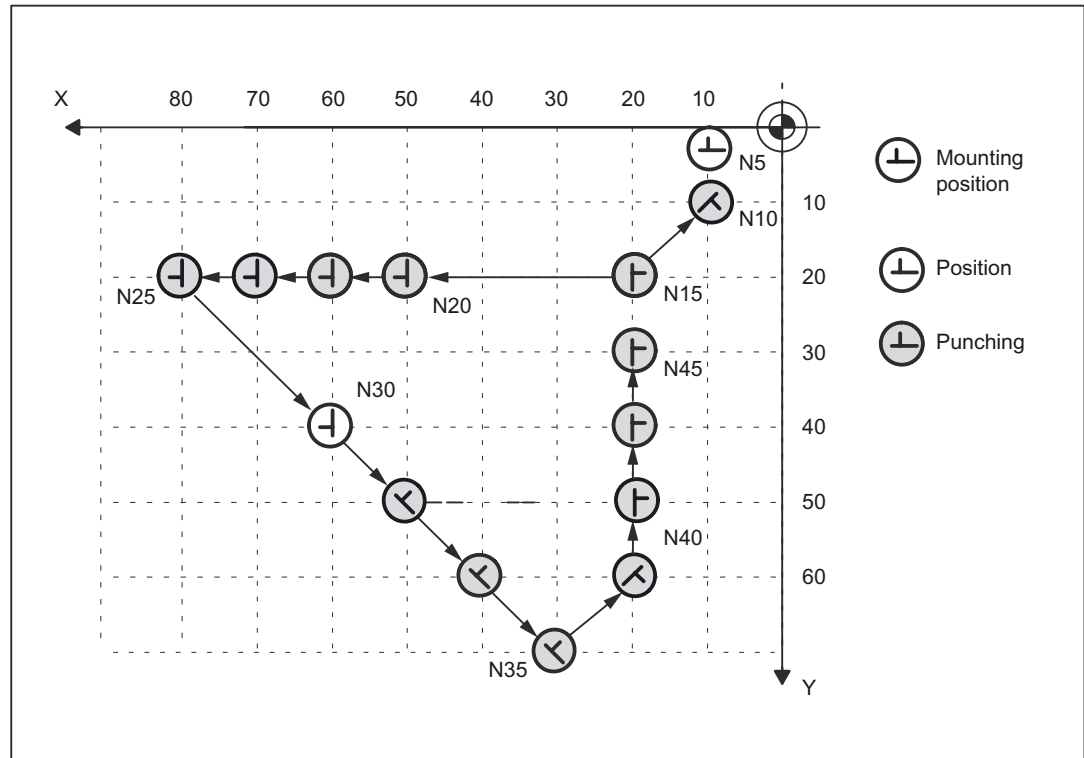



Figure 15-7 Illustration of programming example in XY plane

Example: Circular interpolation

In circular interpolation mode, particularly when path segmentation is active, the tool axes rotate along a path tangentially aligned to the programmed path axes in each sub-block.

Programming example:

```

:
N2 TANG (C, X, Y, 1, "B") ; Define master and slave axes, C is
                           ; slave axis for X and Y in the base
                           ; coordinate system
N5 G0 F60 X10 Y10 ; Start position
N8 TRAILON (C1, C, 1) ; Activate coupled motion of rotatable
                       ; tool axes C/C1 for punch and die.
N9 TANGON (C, -90) ; Activate tangential control with
                   ; offset 270°
N10 G02 X30 Y30 I20 J0 SPN=2 PON ; Circular interpolation with path
                                  ; segmentation, two strokes are executed
                                  ; with 270° offset angle and tangential
                                  ; alignment along circular path
N15 G0 X70 Y10 SPOF ; Position
N17 TANGON (C, 90) ; Activate tangential control with
                   ; offset 90°
N20 G03 X35,86 Y24,14 CR=20 SPP=16 SON ; Circular interpolation, path
                                          ; segmentation, four strokes are

```

| | |
|--|---|
| | executed with 90° offset angle and tangential alignment along circular path |
| N25 G0 X74,14 Y35,86 C0 PON | ; Rotation of tool axes to 0°, stroke |
| N27 TANGON (C, 0) | ; Activate tangential control with offset 0° |
| N30 G03 X40 Y50 I-14,14 J14,14 SPN=5 SON | ; Circular interpolation, path segmentation, five strokes with 0° offset angle and tangential alignment along circular path |
| N35 G0 X30 Y65 C90 SPOF | ; Position without active tangential control |
| N40 G91 X-10 Y-25 C180 | ; Positioning, C axis rotates to 270° |
| N43 TANGOF | ; Deactivate tangential control |
| N45 G90 G02 Y60 I0 J10 SPP=2 PON | ; Circular interpolation, path segmentation, two strokes without tangential control where C=270° |
| N50 SPOF | ; Punching OFF |
| N55 TRAILOF (C1, C) | ; Deactivate coupled motion of rotatable tool axes C/C1 |
| N60 M2 | |

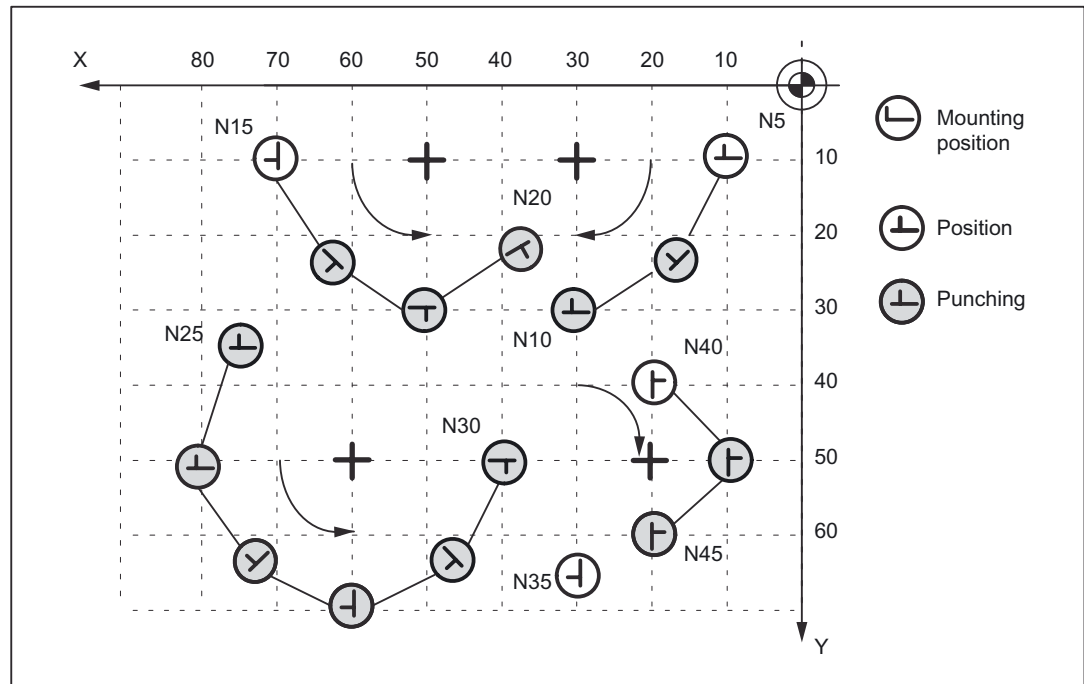


Figure 15-8 Illustration of programming example in XY plane

15.6 Protection zones

Clamping protection zone

The "Clamping protection zone" function monitors whether clamps and tool could represent a mutual risk.

Note

No by-pass strategies are implemented for cases where the clamp protection is violated.

Reference:

/BPN/ Operation and Programming, Nibbling, Section Clamping protection

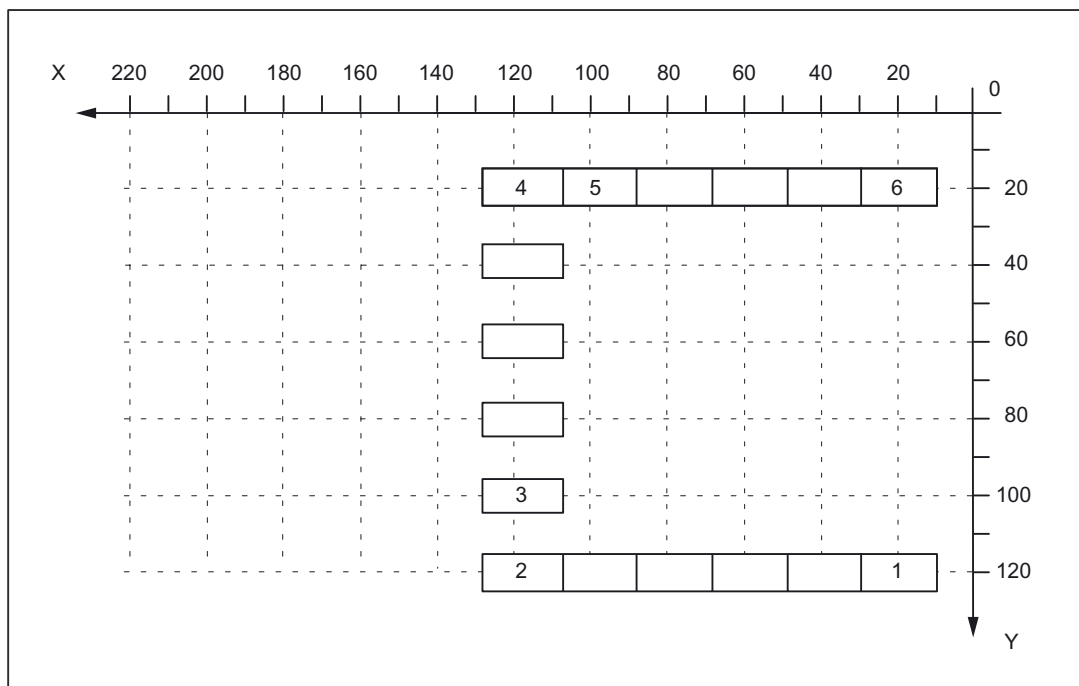
15.7 Examples of defined start of nibbling operation

Example 1

Example of defined start of nibbling operation

```

:
N10 G0 X20 Y120 SPP= 20          ; Position 1 is approached
N20 X120 SON                      ; Defined start of nibbling, first stroke at "1",
                                ; last stroke at "2"
N30 Y20                          ; Defined start of nibbling, first stroke at "3",
                                ; last stroke at "4"
N40 X20                          ; Defined start of nibbling, first stroke at "5",
                                ; last stroke at "6"
N50 SPOF
N60 M2
    
```



Example 2

This example utilizes the "Tangential control" function. Z has been selected as the name of the tangential axis.

```

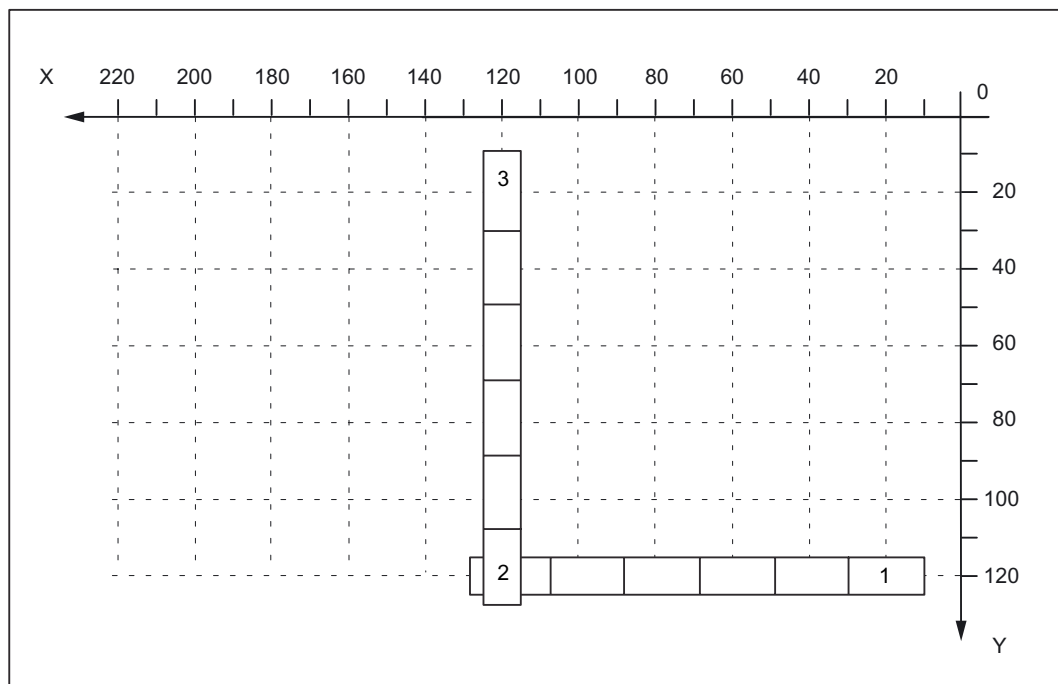
:
N5 TANG (Z, X, Y, 1, "B")        ; Define tangential axis
N8 TANGON (Z, 0)                 ; Select tangential control
N10 G0 X20 Y120                  ; Position 1 is approached
N20 X120 SPP=20 SON              ; Defined start of nibbling, tangential control
    
```

15.7 Examples of defined start of nibbling operation

```

selected, first stroke at "1", last stroke at
"2"
N30 SPOF TANGOF           ; Deselect nibbling mode and deselect tangential
                           control
N38 TANGON (Z, 90)        ; Select tangential control
N40 Y20 SON               ; Defined start of nibbling, tangential control
                           selected, first stroke at "2" rotated 90 degrees
                           in relation to block N20, last stroke at "3"
N50 SPOF TANGOF           ; Deselect nibbling mode and deselect tangential
                           control
N60 M2

```



Examples 3 and 4 for defined start of nibbling

Example 3 Programming of SPP

```

:
N5 G0 X10 Y10             ; Position
N10 X90 SPP=20 SON        ; Defined start of nibbling, 5 punch initiations
N20 X10 Y30 SPP=0         ; One punch is initiated at end of path
N30 X90 SPP=20            ; 4 punches initiated at intervals of 20 mm
N40 SPOF
N50 M2

```

Example 4 Programming of SPN

```

:
N5 G0 X10 Y10 ; Position
N10 X90 SPN=4 SON ; Defined start of nibbling, 5 punch initiations
N20 X10 Y30 PON ; One punch is initiated at end of path
N30 X90 SPN=4 ; 4 punches initiated
N40 SPOF
N50 M2
    
```

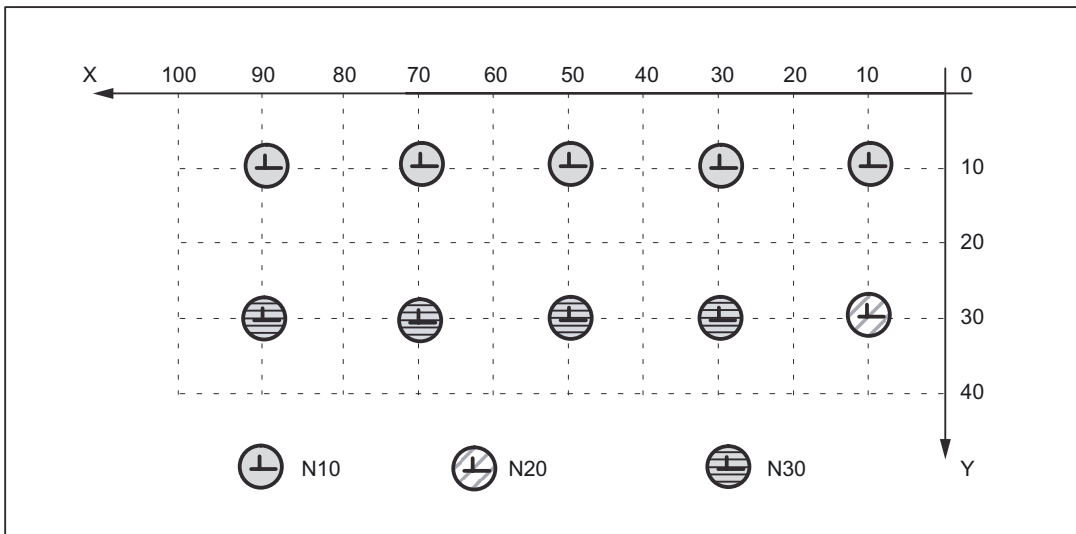


Figure 15-9 Examples 3 and 4 for defined start of nibbling

Examples 5 and 6 without defined start of nibbling

Example 5 Programming of SPP

```

:
N5 G0 X10 Y30 ; Position
N10 X90 SPP=20 PON ; No defined start of nibbling, 4 punches initiated
N15 Y10 ; One punch is initiated at end of path
N20 X10 SPP=20 ; 4 punches initiated at intervals of 20 mm
N25 SPOF
N30 M2
    
```

Example 6 Programming of SPN

```

:
N5 G0 X10 Y30           ; Position
N10 X90 SPN=4 PON       ; No defined start of nibbling, 4 punches initiated
N15 Y10                 ; One punch is initiated at end of path
N20 X10 SPN=4           ; 4 punches initiated
N25 SPOF
N30 M2
    
```

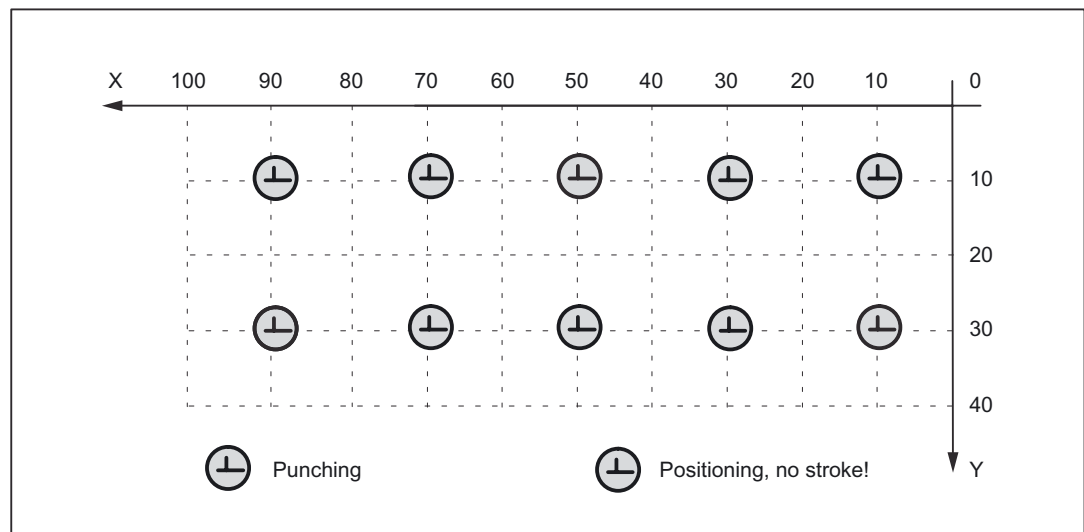


Figure 15-10 Examples 5 and 6 without defined start of nibbling

Example 7 Application example of SPP programming

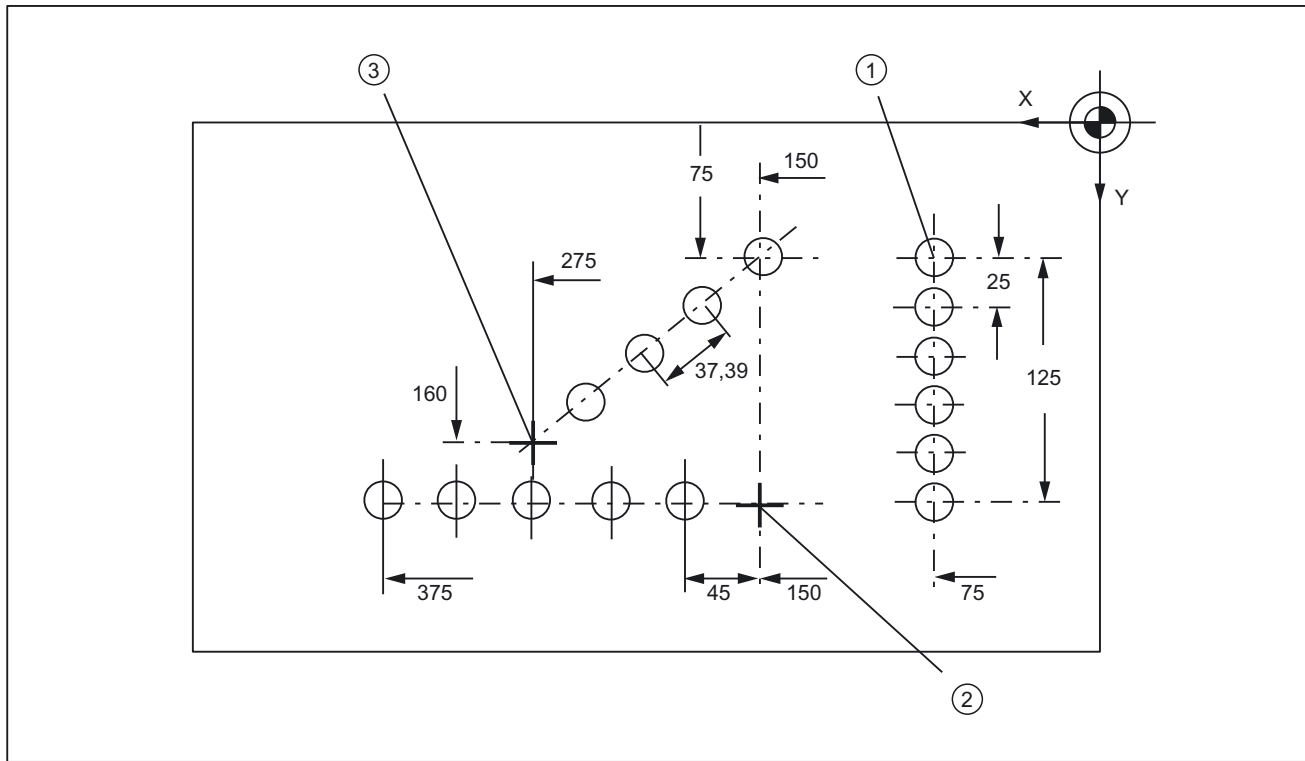


Figure 15-11 Workpiece

Extract from program:

```

N100 G90 X75 Y75 F60 PON           ; Position at starting point 1 of vertical line of
                                   holes, punch one hole
N110 G91 Y125 SPP=25 PON           ; End point coordinates (incremental), path
                                   segment: 25 mm, activate punching
N120 G90 X150 SPOF                 ; Absolute dimensioning, position at starting
                                   point 2 of horizontal line of holes
N130 X375 SPP=45 PON               ; End point coordinates, path segment: 45 mm
N140 X275 Y160 SPOF               ; Position at starting point 3 of oblique line of
                                   holes
N150 X150 Y75 SPP=40 PON           ; End point coordinates, programmed path segment:
                                   40 mm, calculated path segment: 37,79 mm
N160 G00 Y300 SPOF                ; Position
    
```


15.8 Data lists

15.8.1 Machine data

| Number | Identifier | Name |
|------------------|-------------------------|---|
| General | | |
| 11450 | SEARCH_RUN_MODE | Block search parameter settings |
| Channel-specific | | |
| 26000 | PUNCHNIB_ASSIGN_FASTIN | Hardware assignment for input-byte with stroke control |
| 26002 | PUNCHNIB_ASSIGN_FASTOUT | Hardware assignment for output-byte with stroke control |
| 26004 | NIBBLE_PUNCH_OUTMASK[n] | Mask for quick output bits |
| 26006 | NIBBLE_PUNCH_INMASK[n] | Mask for quick input bits |
| 26010 | PUNCHNIB_AXIS_MASK | Definition of punching and nibbling axes |
| 26014 | PUNCH_PATH_SPLITTING | Activation of automatic path segmentation |
| 26016 | PUNCH_PARTITION_TYPE | Behavior of single axes with active automatic path segmentation |
| 26018 | NIBBLE_PRE_START_TIME | Automatically activated pre-initiation time |
| 26020 | NIBBLE_SIGNAL_CHECK | Monitoring of the input signal |

15.8.2 Setting data

| Number | Identifier | Description |
|------------------|-------------------------|---|
| Channel-specific | | |
| 42400 | PUNCH_DWELL_TIME | Dwell time |
| 42402 | NIBPUNCH_PRE_START_TIME | Pre-start time |
| 42404 | MINTIME_BETWEEN_STROKES | Minimum time interval between two consecutive strokes |

15.8.3 Interface signals

| Number | Bit | Name |
|----------------------|-----|---|
| Signals to channel | | |
| V3200 0003 | .0 | Stroke enable |
| V3200 0003 | .1 | Manual stroke initiation |
| V3200 0003 | .2 | Stroke suppression |
| V3200 0003 | .3 | Stroke inoperative |
| V3200 0003 | .4 | Delayed stroke |
| V3200 0003 | .5 | Manual stroke initiation |
| Signals from channel | | |
| V3300 0006 | .0 | Stroke initiation active |
| V3300 0006 | .1 | Acknowledgement of manual stroke initiation |

Transverse Axes (P1)

16.1 Transverse axis definition

Geometry axis as transverse axis

The **geometry axis X** is defined as a transverse axis. The transverse axis is important for lathe functions.

16.2 Diameter programming

Activation and deactivation

The diameter or radius can be programmed for transverse axes.

Program commands "DIAMON" and "DIAMOF" can be used to activate and deactivate the diameter programming of a transverse axis. DIAMON and DIAMOF belong to G group 29 and are modal.

JOG mode

If DIAMON is active, then the increments entered for machine functions INC (incremental dimension) and handwheel traverse in JOG are interpreted and traversed as diameter values for the associated transverse axis (traversal in WCS with this axis).

Setpoint / actual value display

If the "DIAMON" function is activated for a transverse axis, then the position, distance to go and REPOS offset are displayed in the diameter when the workpiece coordinate system (WCS) is selected.

The display is always in the radius in the machine coordinate system (MCS).

Offsets

All offsets (e.g. tool offsets, programmable and settable zero offsets) are always entered, programmed and displayed as radius values (even if they are active in the transverse axis and command DIAMON is active).

Working area limitation, software limit switches, feed values

This data is always entered, programmed and displayed as radius values.

Conversion of diameter values to internal radius values

With diameter programming active, the internal radius values are converted for the transverse axis (i.e. the programmed values are halved):

- Programmed end positions
- Absolute interpolation parameters (e.g. I, J, K) for G2/G3 programming. Absolute interpolation parameters are in relation to the coordinate origin of the WCS. Interpolation parameters programmed relatively are not converted.

References:

"Operation and Programming"

Conversion of internal radius values to diameter values

For active diameter programming the measurement results when measuring in the **WCS** are converted for the transverse axis with the "MEAS", "MEASW" functions (i.e. the internal radius values are doubled) and stored.

For measurement or reading in the **MCS** the calculated values are stored as radius values.

16.3 Constant cutting rate: G96

Functionality

Requirement: A controlled spindle must be present.

With activated G96 function, the spindle speed is adapted to the currently machined workpiece diameter (position of the transverse axis = geometry axis X) such that a programmed cutting rate S remains constant on the tool edge (spindle speed times diameter = constant).

The S word is evaluated as the cutting rate as of the block with G96. G96 is modally effective until cancellation by another G function of the group (G94, G95, G97).

Programming

| | |
|------------------------|--|
| G96 S... LIMS=... F... | Constant cutting speed ON |
| G97 | Constant cutting speed OFF |
| Switchgear protection | S Cutting rate, unit m/min |
| LIMS= | Upper limit speed of the spindle with G96, G97 effective |
| F | Feedrate in mm/revolution – as for G95 |

Reference:

/BP_/ Operation and Programming

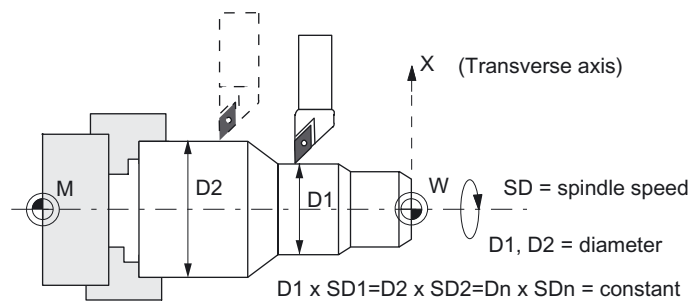


Figure 16-1 Constant cutting rate G96

Positioning Axes (P2)

17.1 Concurrent positioning axis

If axes are available on a machine tool for auxiliary movements, they can only be triggered from the PLC during machining operation.

Note

A positioning axis controlled by the PLC can also be an indexing axis. However, it is not possible to control spindles in versions G/N plus and pro.

Function

Concurrent positioning axes are positioning axes with the following properties:

- Activation from the PLC need not take place at block limits, but can be implemented at any time in any operating mode (even when a part program is already being processed in the channel).
- The part program continues to run uninhibited, even if the concurrent positioning axis has not reached the position defined by the PLC.

Activation from PLC

The concurrent positioning axis is activated via the user interface area 380x 3000 / 390x 3000 from the PLC.

- Feedrate (with feedrate setting = 0, the feedrate is determined by the setting in the following machine data):
MD32060 POS_AX_VELO (initial setting for positioning axis velocity)
- Absolute dimension (G90), incremental dimension (G91),
Absolute dimension along the shortest path for rotary axes (rotary axis name = DC(value))

The following functions are defined:

- Linear interpolation (G01)
- Feedrate in mm/min or degrees/min (G94)
- Exact stop (G09)
- Settable zero offsets currently selected are valid

Applications

Typical applications for concurrent positioning axes include:

- Tool magazines with manual loading and unloading during machining
- Tool magazines with tool preparation during machining

17.2 Permanently assigned PLC axis

Function

The option of controlling an axis from the PLC is controlled via the machine data MD30460 BASE_FUNCTION_MASK.

Commissioning

The permanently assigned PLC axis is activated by machine data

MD30460 BASE_FUNCTION_MASK on Bit 5 = 1

If the PLC axis has rotary axis functionality, the following must also be set:

- MD30300 IS_ROT_AX = 1
- MD30310 ROT_IS_MODUL == 1
- MD30320 DISPLAY_IS_MODULO = 1

During acceleration the axis becomes a neutral axis. When a traverse request is transferred via the VDI interface, a neutral axis without preceding axis replacement, automatically changes to a competing positioning axis (PLC axis).

After completing the traverse, if, for instance, the programmed end position has been reached, or if the traverse has been interrupted, the axis again becomes the neutral axis. In this condition the axis can be moved or referenced via

JOG with the traverse keys or the handwheel.

Note

A permanently assigned PLC axis cannot be replaced.

The following functions are **not** possible for a permanently assigned PLC axis:

- Replacement to command axis
- Replacement to oscillation axis
- Assignment of the axis to the NC program

The permanently assigned PLC axis reports its properties via the VDI interface with IS V390x 0011.7 (permanently assigned PLC axis) to the PLC.

Initialization

The function is activated by the positive edge of the V380x 3000.7 signal (start positioning axis). The interface signal must remain at logical "1" until the function has been acknowledged positively or negatively by the following interface signals:

- V390x 3000.6 = 1 (position reached)
- V390x 3000.1 = 1 (error during traversing)
- V390x 3000.0 = 1 (axis cannot be started)

IS V390x 3000.7 = 1 (Positioning axis active) indicates that the function is active and that the output signals are valid.

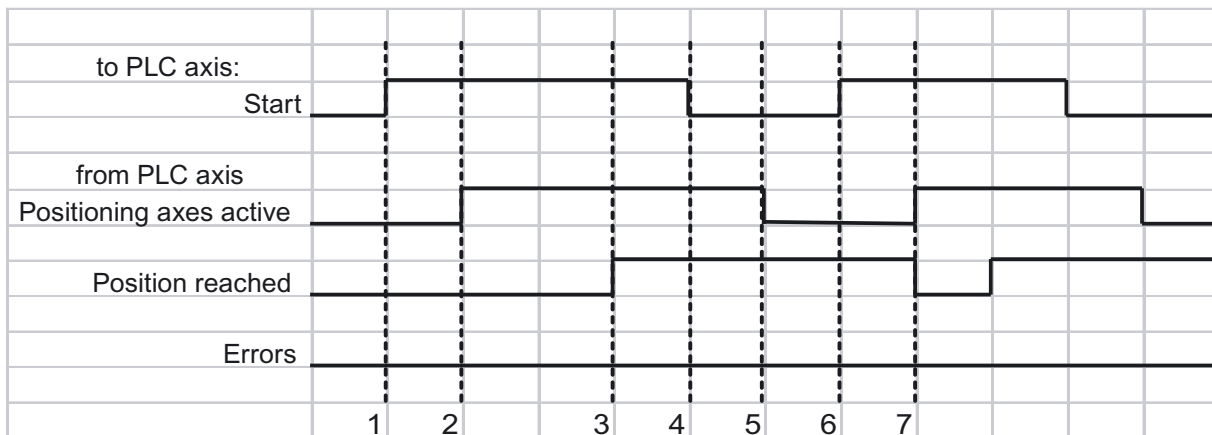


Figure 17-1 Pulse diagram normal condition

1. First function activation via positive edge of *Start*
2. *Positioning axis active* = 1 shows that the function is active and that the output signals are valid
3. Positive acknowledgement *Position reached* = 1 and *positioning axis active* = 1
4. Reset function activation after receipt of acknowledgment
5. Signal change via function
6. Second function activation via positive edge of *Start*
7. *Positioning axis active* = 1 shows that the function is active and that the output signals are valid

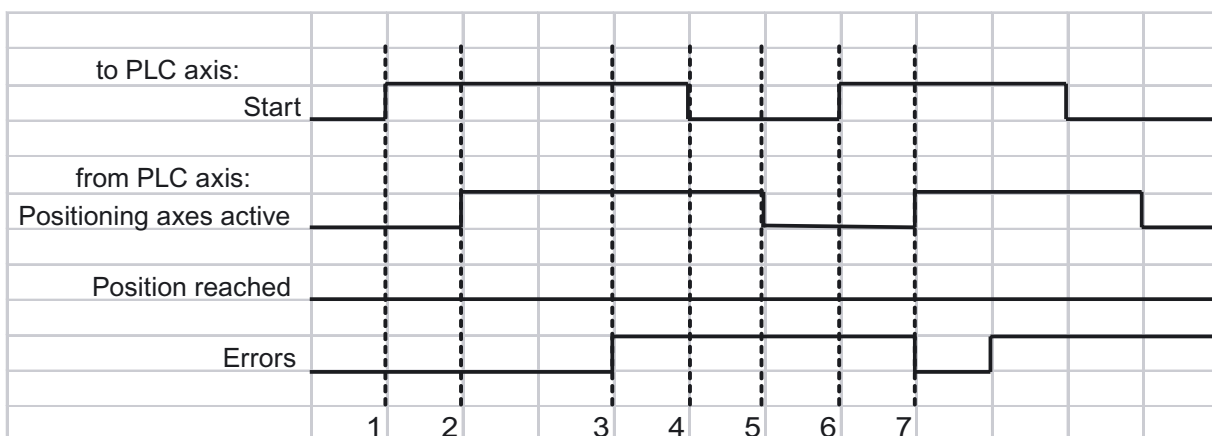


Figure 17-2 Pulse diagram error condition

8. First function activation via positive edge of *Start*
9. *Positioning axis active* = 1 shows that the function is active and that the output signals are valid
10. Negative acknowledgement *Error* = 1 and *Positioning axis active* = 1
11. Reset function activation after receipt of acknowledgment
12. Signal change via function

13. Second function activation via positive edge of *Start*

14. *Positioning axis active* = 1 shows that the function is active and that the output signals are valid

Cancel

The PLC axis control cannot be aborted by means of parameter "Start", but only by means of the axial interface signals (e.g. delete distance). The axial interface also returns status signals of the axis that may need to be evaluated (e.g. exact stop, traverse command).

Axis disable

With the axis disabled, an axis controlled via PLC axis control will not move. Only a simulated actual value is generated. (Behavior as with NC programming).

Fault detection

If a PLC axis control could not be executed, this is indicated by the error signal (V390x 3000.1 or V390x 3000.0) with 'logical 1'. The cause of the error is coded as an error number.

Axis control

If an axis is in the neutral state, it can be controlled by the signals AXRESET, AXSTOP and AXRESUME, effective in the channel.

If during acceleration it is detected that an axis which was defined as a geometry axis simultaneously has been defined as permanently assigned PLC axis, the axis function will be denied with alarm 4320 axis 1% function MA_BASE_FUNCTION_MASK Bit5 and MC_AXCONF_GEOAX_ASSIGN_TAB.

If a change to a PLC axis is not possible

In case of a traverse request from the VDI interface area, an illegal change to a PLC axis is ended with the start signal "Competing positioning axis active" at the VDI interface.

Note

Competing positioning axes get their block end points exclusively from the PLC and can be started at any point in time.

17.3 Data lists

17.3.1 Machine data

| Number | Identifier | Description |
|-----------------------|--------------------|-------------------------------|
| Axis/spindle-specific | | |
| 30460 | BASE_FUNCTION_MASK | Axis functions |
| 32060 | POS_AX_VELO | Feedrate for positioning axis |

17.3.2 Interface signals

| Number | Bit | Name |
|---------------------------|-----|-------------------------------|
| Signals to channel | | |
| V3200 0006 | .0 | Feed disable |
| V3200 0007 | .1 | NC Start |
| V3200 0007 | .4 | NC stop axes plus spindle |
| Signals from channel | | |
| V3300 0004 | .3 | All axes stationary |
| Signals to axis/spindle | | |
| V380x 3000 | .7 | Start positioning axis |
| V380x 5004 | .1 | AxReset |
| V380x 5004 | .2 | AxResume |
| V380x 5004 | .6 | AxStop, stop |
| V380x 5004 | .7 | PLC controls axis |
| Signals from axis/spindle | | |
| V390x 0011 | .7 | PLC axis permanently assigned |
| V390x 3000 | .0 | Axis cannot be started |
| V390x 3000 | .1 | Error during traversing |
| V390x 3000 | .6 | Position reached |
| V390x 3000 | .7 | Positioning axes active |

17.3.3 Error messages

Error messages

When an error occurs in a PLC axis, an error code is output via VB390x 3003.

Table 17- 1 Error messages via NCK

| Dec. | Hex. | Meaning |
|---|------|--|
| Errors that occur as a result of how the NCK is handled | | |
| 30 | 1e | The axis/spindle has been transferred to the NC while still in motion. |
| 50 | 32 | Permanently assigned PLC axis jogs or references |
| 60 | 3C | Permanently assigned PLC axis channel status does not currently permit a start |
| 100 | 64 | Corresponds to alarm number 16830 |
| 105 | 69 | Corresponds to alarm number 16770 |
| 106 | 6a | Corresponds to alarm number 22052 |
| 107 | 6b | Corresponds to alarm number 22051 |
| 108 | 6c | Corresponds to alarm number 22050 |
| 109 | 6d | Corresponds to alarm number 22055 |
| 110 | 6e | Velocity/speed is negative |
| 111 | 6f | Setpoint speed is zero |
| 112 | 70 | Invalid gear stage |
| 115 | 73 | Programmed position has not been reached |
| 117 | 75 | G96/G961 is not active in the NC |
| 118 | 76 | G96/G961 is still active in the NC |
| 120 | 78 | Not an indexing axis |
| 121 | 79 | Indexing position error |
| 125 | 7d | DC (shortest path) not possible |
| 126 | 7e | Minus absolute value not possible |
| 127 | 7f | Plus absolute value not possible |
| 130 | 82 | Software limit switch plus |
| 131 | 83 | Software limit switch minus |
| 132 | 84 | Working area limitation plus |
| 133 | 85 | Working area limitation minus |
| 135 | 8/ | Corresponds to alarm number 17501 |
| 136 | 88 | Corresponds to alarm number 17503 |
| System or other serious alarms | | |
| 200 | c8 | Corresponds to system alarm number 450007 |

References

The alarms are described in:

/DIA/ Diagnostics Guide

Reference Point Approach (R1)

18.1 Fundamentals

Why reference?

The control must be synchronized with the position measurement system of each machine axis so that the control can detect the exact machine zero when it is switched on. This process is known as referencing.

The spindle process (synchronizing) is largely described in the Chapter "Spindles".

Position measurement systems

The following position measuring systems can be mounted on the motor:

- Incremental rotary measuring system
- Absolute rotary measuring system

The referencing for the mounted position measuring systems can be set with MD34200 ENC_REFP_MODE (referencing mode).

Output cam

An output cam for referencing may be required for linear axes, and its signal has the following tasks:

- Selection of the direction of travel when approaching the zero mark (synchronized pulse)
- Selection of the zero mark, where required.

BERO

A BERO (inductive proximity switch) can be deployed as the encoder for the synchronized pulse (instead of the zero mark of the position encoder) (preferred for rotary axes, spindles). Here connection is made to the 802D sl via terminal X20. Special inputs to BERO inputs are switched via macro parameterization.

Reference:

/BE/ 802D sl Instruction Manual, Section "Configuring SINAMICS S120 with 802D sl"

IS "Active machine function REF" (V3100 0001.2)

The reference point approach is performed with the REF machine function activated (IS "active machine function REF"). The REF machine function can be selected in JOG modes (IS "REF machine function" (V3000 0001.2)).

Axis-specific referencing

Axis-specific referencing is started separately for each machine axis with the "plus/minus traversing keys" interface signal (V380x 0004.7 / .6). All axes can be referenced at the same time. If the machine axes are to be referenced in a particular sequence, the following options are available:

- The operator must observe the correct sequence when starting.
- The PLC user program checks the sequence on start-up or defines the sequence itself.
- The order is defined in MD34110 REFP_CYCLE_NR (see channel-specific referencing)

Channel-specific referencing

Channel-specific referencing is started with the "activate referencing" interface signal (V3200 0001.0). The control acknowledges a successful start with IS "Referencing active" (V3300 0001.0). Each machine axis assigned to the channel can be referenced with channel-specific referencing (this is achieved internally on the control by simulating the plus/minus traversing keys). Axis-specific MD34110 REFP_CYCLE_NR (axis sequence for channel-spec. referencing) can be used to define the sequence in which the machine data is referenced. If all axes entered in MD34110 REFP_CYCLE_NR have reached their end points, the "All axes referenced" interface signal (V3300 0004.2) is enabled.

Special features

- Referencing is aborted with "Reset" interface signal (V3000 0000.7). All axes that have not reached their reference point by this time are considered to be not referenced. IS "Referencing active" is reset and alarm 20005 is signaled.
- Working area limiting and software limit switches are not active for non-referenced machine axes.
- The defined axis-specific accelerations are observed at all times during referencing (except when alarms occur).
- The reference point approach can be started only with the direction key for the direction stored in MD34010 REFP_CAM_DIR_IS_MINUS.

Referencing in the part program

One or more axes that have lost their reference can be referenced at the same time. The sequence of the individual phases is identical to axis-specific referencing, except that the process is started with the G74 command instead of the plus/minus traversing keys and is done via the machine axis identifiers.

Reference:

/BP_/ Operation and Programming

Note

MD20700 REFP_NC_START_LOCK = 1 prevents a part program from being started (alarm output) if not all required axes are referenced.

18.2 Referencing with incremental measuring systems

Time sequence

The referencing sequence for incremental measuring systems can be subdivided into three phases:

1. Phase: Traversing to the reference cam
2. Phase: Synchronization with the zero mark
3. Phase: Traversing to the reference point

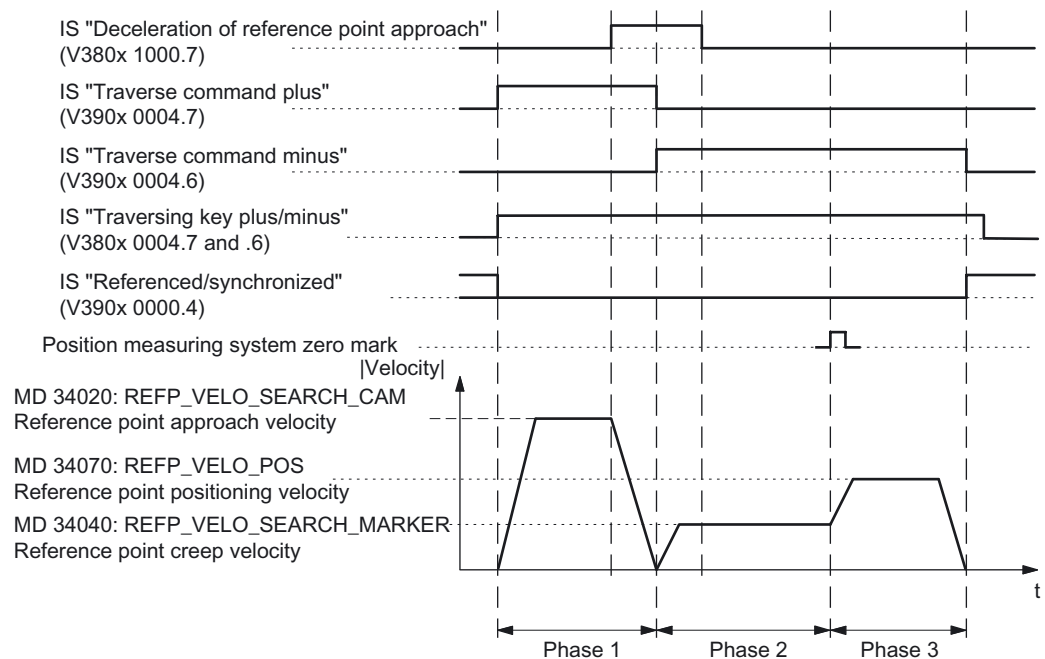


Figure 18-1 Referencing sequence with incremental measuring system (example)

Characteristics of traversing to the reference point cam (phase 1)

- The feedrate override and feedrate stop is in effect.
- The machine axis can be stopped/started.
- The cam must be reached within the traversing distance in MD34030 REFP_MAX_CAM_DIST, otherwise a corresponding alarm is triggered.
- The machine axis must come to a halt at the cam, otherwise a corresponding alarm is triggered.

Characteristics when synchronizing with the zero pulse (phase 2)

- Feedrate override is not active. Feedrate override 100% is active. Termination occurs at feedrate override 0%.
- Feedrate stop is active, the axis comes to a halt and a corresponding alarm is displayed.
- The machine axis cannot be stopped and restarted with NC stop/NC start.
- Monitoring of the zero mark is active with MD34060 REFP_MAX_MARKER_DIST.

Characteristics of traversing to the reference point (phase 3)

- The feedrate override and feedrate stop is in effect.
- The machine axis can be stopped and re-started with NC stop/NC start.
- If reference point offset is smaller than the braking distance of the machine axis from the reference point positioning velocity to stop, the reference point is approached from the opposite direction.

Different motion sequences during referencing:

| Referencing type | Synchronizing pulse (zero mark, BERO) | Motion sequence |
|---|---|-----------------|
| With reference cam (MD34000 REFP_CAM_IS_ACTIVE = 1) | Synchronizing pulse before cam, reference coordinate before synchronizing pulse = without reversal: (MD34050 REFP_SEARCH_MARKER_REVERSE = 0) | |
| | Synchronizing pulse on cam, reference coordinate after synchronizing pulse = with reversal: (MD34050 REFP_SEARCH_MARKER_REVERSE = 1) | |

| Referencing type | Synchronizing pulse (zero mark, BERO) | Motion sequence |
|--|--|--|
| Without reference cam (MD34000 REFP_CAM_IS_ACTIVE = 0) | Reference coordinate after synchronizing pulse | <p>The diagram shows a velocity profile starting at 'Start'. It first moves at a constant velocity V_M (creep velocity) until it reaches a 'Synchronizing pulse' (represented by a vertical bar). After the pulse, the velocity increases to V_P (positioning velocity) and continues until it reaches a reference point. The distance from the start of the pulse to the reference point is labeled R_V (reference point offset). The velocity profile ends with a deceleration phase.</p> |
| <p>V_C - reference point approach velocity (MD34020 REFP_VELO_SEARCH_CAM) V_M - reference point creep velocity (MD34040 REFP_VELO_SEARCH_MARKER) V_P - reference point positioning velocity (MD34070 REFP_VELO_POS) R_V - reference point offset (MD34080 REFP_MOVE_DIST + MD34090 REFP_MOVE_DIST_CORR) R_K - reference point coordinate (MD34100 REFP_SET_POS)</p> | | |

What is the minimum length of a reference cam?

Example of case: Synchronizing pulse before cam, reference coordinate before synchronizing pulse = synchronizing pulse search with falling cam edge.

The reference cam must be long enough, so that when the cam is approached with the reference point approach velocity, the braking operation ends at the cam (the axis comes to a standstill at the cam), and the cam is exited in the opposite direction with the reference point creep velocity (exit with constant velocity).

To calculate the minimum length of the cam, the larger of the two velocities must be inserted into the formula:

$$\text{Min. length} = \frac{(\text{reference point approach speed or creep speed})^2}{2 \times \text{axis acceleration (MD 32300: MAX_AX_ACCEL)}}$$

If the machine axis does not come to a halt at the reference cam (interface signal "Reference point approach delay" (V380x 1000.7) is reset), alarm 20001 is output. Alarm 20001 can occur if the reference cam is too short and the machine axis travels over it when decelerating in phase 1.

If the reference cam extends to the end of travel of the axis, an inadmissible starting point for referencing (after the cam) can be excluded.

Reference cam adjustment

The reference cam must be calibrated exactly. The following factors influence the response time of the control when detecting the reference cam ("Reference point approach delay" interface signal):

- Switching accuracy of the reference cam switch
- Delay of the reference cam switch (NC contact)
- Delay at the PLC input
- PLC cycle time
- Internal processing time

Practice has shown that the signal edge of the reference cam, which is required for synchronizing, is aligned between two synchronized pulses (zero marks). This can be achieved by:

- Set MD34080 REFP_MOVE_DIST = MD34090 REFP_MOVE_DIST_CORR = MD 34100 REFP_SET_POS = 0
- Reference axis
- In JOG mode, traverse the axis to half the path length between the two zero marks. This path is independent of the pitch of the leadscrew S and the gear ratio n (e.g. S=10 mm/rev, n=1:1 produces a path of 5 mm).
- Calibrate the cam switch so that switching is done at exactly this position (IS "Reference point approach delay" (V380x 1000.7))
- Alternatively, the value of MD34092 REFP_CAM_SHIFT can be changed instead of moving the cam switch.

 **WARNING**

If the reference cam is not calibrated precisely, an incorrect synchronizing pulse (zero mark) may be evaluated. In this case, the control assumes an incorrect machine zero and moves the axes to incorrect positions. Software limit switches act on incorrect positions and are therefore not able to protect the machine.

18.3 Referencing with distancecoded reference markers

18.3.1 General information

Distance-coded reference markers

Measuring systems with distance-coded reference marks consist of two parallel scale tracks:

- Incremental grating
- Reference mark track

The distance between any two consecutive reference markers is defined. This makes it possible to determine the absolute position of the machine axis when two consecutive reference marks are crossed. For example, if the distance between the reference marks is approx. 10 mm, a traverse path of approx. 20 mm is all that is required to reference the machine axis.

Referencing can be performed from any axis position in the positive or negative direction (exception: end of travel range).

18.3.2 Basic parameter assignment

Linear measuring systems.

The following data must be set to parameterize linear measuring systems:

- The absolute offset between the machine zero point and the position of the first reference mark of the linear measuring system:

MD34090 REFP_MOVE_DIST_CORR (reference point/absolute offset)

See also below: Determining the absolute offset

- Orientation of the length measuring system (equidirectional or inverse) relative to the machine system coordinate system:

MD34320 ENC_INVERS (linear measuring system inverse to the machine system)

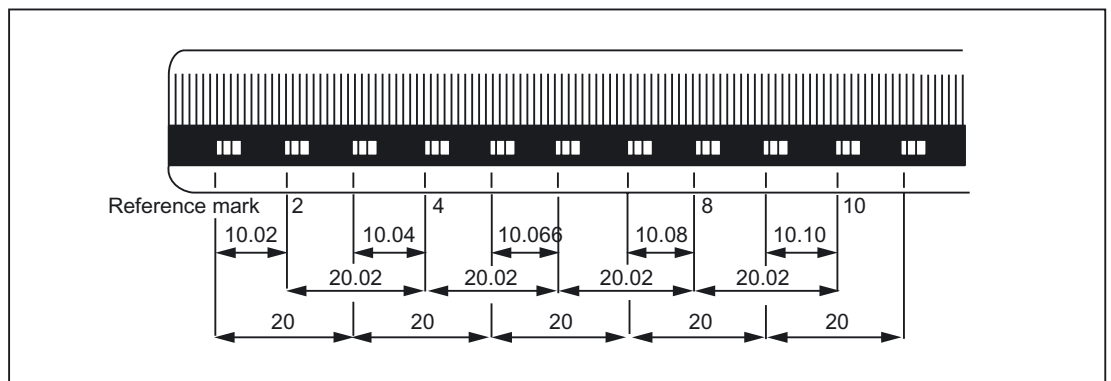


Figure 18-2 DIADUR graduated glass scale with distance-coded reference marks (dimensions in mm for 20 mm scale division)

Rotary measuring system

For rotary measuring systems, the same applies as for linear measuring systems (see above).

Determining the absolute offset

The following procedure is recommended for the determination of the absolute offset between the machine zero point and the position of the first reference mark of a machine axis:

1. Enter the value zero for the absolute offset:
MD34090 REFP_MOVE_DIST_CORR = 0
2. Perform reference point approach.


Note

The reference point should be approached at a point in the machine where the exact position of the machine axis relative to machine zero can be determined easily (using a laser interferometer, for example).

3. Determine the actual position of the machine axis via the operator interface screen.
4. Measure the current position of the machine axis with reference to the machine zero point.
5. Calculate absolute offset and enter in MD34090.

The absolute offset is calculated with respect to the machine coordinate system and depending on the orientation of the measuring system (equidirectional or inverse) as:

| Orientation of the measuring system | Absolute offset |
|-------------------------------------|---|
| equidirectional | Measured position + displayed actual position |
| Opposite direction | Measured position - displayed actual position |

| |
|--|
|  WARNING |
| After determining the absolute offset and the entry in MD34090, the reference point traversing for the machine axis must be carried out once more. |

Referencing methods

Referencing with distance-coded reference marks can be performed in one of two ways:

- Evaluation of **two** consecutive reference marks:
 MD34200 ENC_REFP_MODE (referencing mode) = 3
 Advantage:
 - Short travel path
- Evaluation of **four** consecutive reference marks:
 MD34200 ENC_REFP_MODE = 8
 Advantage:
 - Plausibility check by NC is possible
 - Increase in reliability of referencing result

18.3.3 Chronological sequence

Chronological sequence

Referencing with distance-coded reference marks can be divided into two phases:

- Phase 1: Travel across the reference marks with synchronization
- Phase 2: Traveling to a fixed destination point

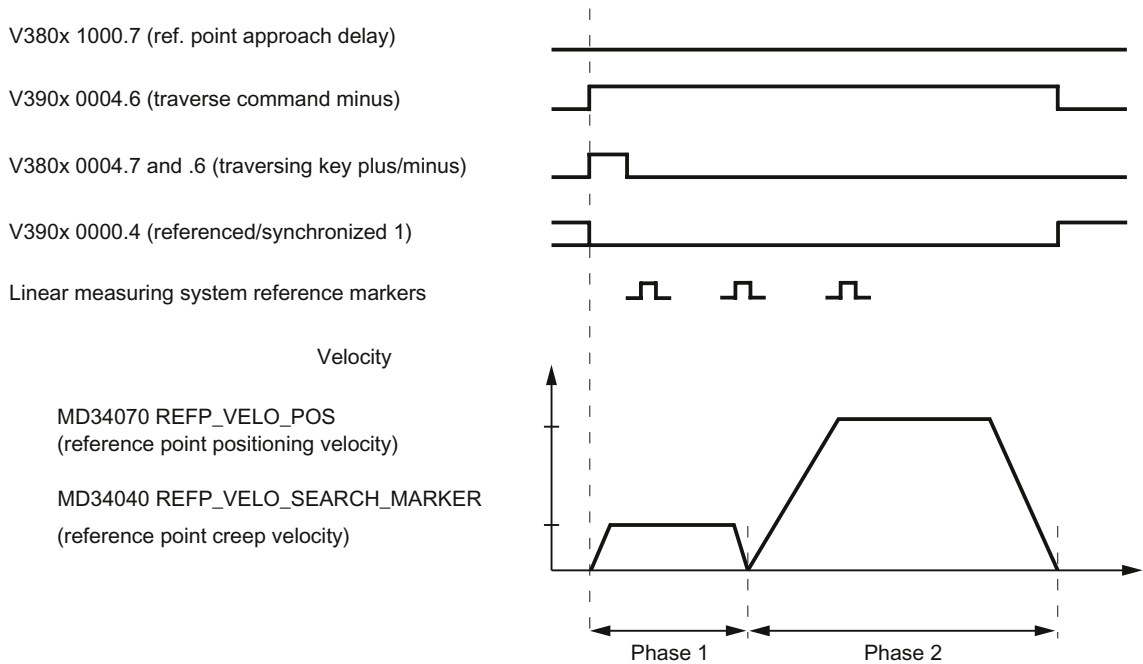


Figure 18-3 Distance-coded reference markers

18.3.4 Phase 1: Travel across the reference marks with synchronization

Phase 1: Start

For information on starting reference point approach, refer to "Axis-specific referencing" and "Channel-specific referencing."

reference cam

In measuring systems with distance-coded reference marks, reference cams are not required for the actual referencing action. For functional reasons, however, a reference cam is required for channel-specific reference point approach and reference point approach from the part program (G74) before the traversing range end of the machine axis.

Phase 1: Sequence

Sequence without touching a reference cam

Once the reference point approaching process is started, the machine axis accelerates to the reference point shutdown speed set by means of parameter assignment:

MD34040 REFP_VELO_SEARCH_MARKER (reference point shutdown speed)

Once the number of reference markers set by means of parameter assignment has been crossed, the machine axis is stopped again and the actual value system of the machine axis is synchronized to the absolute position calculated by the NC.

Sequence when starting from the reference cam

If the machine axis is at the reference cam at the start of the reference point traversing, it accelerates to the parameterized reference point creep velocity against the parameterized reference point approach direction:

MD34040 REFP_VELO_SEARCH_MARKER (reference point shutdown speed)

MD34010 CAM_DIR_IS_MINUS (reference point approach in minus direction)

That ensures that the machine axis does not reach the travel range limit before it has crossed the parameterized number of reference marks.

Once the number of reference markers set by means of parameter assignment has been crossed, the machine axis is stopped again and the actual value system of the machine axis is synchronized to the absolute position calculated by the NC.

Sequence when contact is made with reference cam during referencing

Once the reference point approaching process is started, the machine axis accelerates to the reference point shutdown speed set by means of parameter assignment:

MD34040 REFP_VELO_SEARCH_MARKER (reference point shutdown speed)

Before the machine axis travels over the parameterized number of reference marks, it touches the reference cam. It is then reversed and reference mark search is restarted in the opposite direction.

Once the number of reference markers set by means of parameter assignment has been crossed, the machine axis is stopped again and the actual value system of the machine axis is synchronized to the absolute position calculated by the NC.

Plausibility check of the reference mark distance

An error occurs if, during reference point traversing for two subsequent reference marks, the NC determines a distance greater than twice the parameterized reference mark distance.

MD34300 ENC_REFP_MARKER_DIST (reference marker distance)

The machine axis will then traverse in opposite direction at half the parameterized reference point creep velocity (MD34040) and the search for reference mark is restarted.

If a faulty reference mark distance is detected again, the machine axis is stopped and the reference point traversing is aborted (alarm 20003 "fault in the measuring system").

Abort criterion

If the parameterized number of reference marks is not detected within the parameterized distance, the machine axis is stopped and reference point traversing is aborted.

MD34060 REFP_MAX_MARKER_DIST (max. distance to the reference marker)

Features of phase 1

After phase 1 is successfully completed, the actual value system of the machine axis is synchronized.

18.3.5 Phase 2: Travel to fixed stop**Phase 2: Start**

Phase 2 is automatically started when phase 1 has been completed without an alarm.

Initial situation:

- The machine axis is positioned directly behind the last of the parameterized number of reference marks.
- The actual value system of the machine axis is synchronized.

Phase 2: Sequence

In Phase 2, the machine axis completes reference point approach by traversing to a defined target position (reference point). This action can be suppressed in order to shorten the reference point approach:

MD34330 STOP_AT_ABS_MARKER

| Value | Meaning |
|-------|------------------------------|
| 0 | Travel to target position |
| 1 | No travel to target position |

Travel to target position (normal case)

The machine axis accelerates to the parameterized reference point position velocity and travels to the parameterized target point (reference point):

MD34070 REFP_VELO_POS (reference point positioning velocity)

MD34100 REFP_SET_POS (reference point value)

The machine axis is referenced. To identify this, the NC sets the relevant interface signal: V390x 0000.4 = 1 (referenced/synchronized 1)

No travel to target position

The machine axis is now referenced. To identify this, the NC sets the relevant interface signal:

V390x 0000.4 = 1 (referenced/synchronized 1)

Features of phase 2

Phase 2 will display different characteristics, depending on whether a reference point cam is parameterized for the machine axis.

Machine axis without reference point cam

MD34000 REFP_CAM_IS_ACTIVE (axis with reference point cam) = 0

Properties:

- Feed override active.
- The feed stop (channel-specific and axis-specific) is active.
- NC STOP and NC START are active.

Machine axis with reference point cam

MD34000 REFP_CAM_IS_ACTIVE (axis with reference point cam) = 1

Properties:

- Feedrate override is **not** active.
Machine axis moves internally when feedrate override = 100%.
If a feedrate override of 0% is specified, an abort occurs.
- The feed stop (channel-specific and axis-specific) is active.
- NC-STOP and NC-START are **not** active.
- If the parameterized number of reference marks is not detected within the parameterized distance after the exit of the reference cam, the machine axis will be stopped.

MD34060 REFP_MAX_MARKER_DIST (max. distance to the reference marker)

Special features of rotary measuring systems

On rotary distancecoded measuring systems, the absolute position can only be determined uniquely within one revolution. Depending on the mechanical mounting of the encoder, the overtravel of the absolute position in the hardware does not always coincide with the traversing range of the rotary axis.

Special features of modulo rotary axes

With modulo rotary axes, the reference point position is mapped on the parameterized modulo range:

MD30330 MODULO_RANGE (magnitude of modulo range)

MD30340 MODULO_RANGE_START (starting position of modulo range)

Note

The reference point position is mapped onto the assigned (ghost) modulo range even with axis function "Determination of reference point position rotary, distance-coded encoder within the configured modulo range":

MD30455 MISC_FUNCTION_MASK (axis functions), Bit1 = 1

18.4 Referencing with absolute encoders

18.4.1 General

Requirements

An axis with absolute encoder is referenced automatically when the control is switched on if the system detects that the relevant axis is already calibrated. This transfer of the absolute value takes place without any axis motion, e.g. at POWER ON. Two conditions must be fulfilled before an axis can be automatically referenced:

- The axis has an absolute encoder with which the position control operates
- The Absolute encoder has been calibrated (MD34210 ENC_REFP_STATE = 2)

Adjustment

In the case of axes with absolute encoders, the measurement system is not synchronized by approaching a reference cam. It is calibrated instead. For this purpose, the actual value of the absolute encoder is set once during start-up and transferred to the control.

18.4.2 Operator-assisted calibration

Basic procedure

Move the axis to be calibrated to a defined position to set the resulting actual value.

Chronological sequence

1. Set MD34200 ENC_REFP_MODE and MD34210 ENC_REFP_STATE to 0 and activate by means of a POWER ON. Parameter ENC_REFP_MODE = 0 means that the actual value of the axis is set once.
2. Traverse axis manually in JOG mode to a known position. The direction in which the position is approached must match the direction stored in MD34010 REFP_CAM_DIR_IS_MINUS (0 = positive direction, 1 = negative direction).

Note

This known position must always be approached at a low velocity and from a defined direction so that it is not falsified by any backlash present in the drive train.

3. Enter the actual value corresponding to the approached position in MD34100 REFP_SET_POS. This value may be a specified design value (e.g. fixed stop) or can now be determined with a measuring instrument.
4. Set MD34210 ENC_REFP_STATE to "1" to enable the "calibration" function.
5. The modified machine data become effective after a RESET.
6. Switch to JOG-REF mode.
7. If you press the traversing key already used in step 2, the current offset is entered in MD34090 REFP_MOVE_DIST_CORR and MD34210 ENC_REFP_STATE changes to "2", i.e. the axis is considered to be calibrated. (The screen is updated by pressing the traversing key.)

Note

The axis does not move when the correct traversing key is actuated! The value entered in MD34100 REFP_SET_POS appears in the actual value display of the axis position.

8. Exit JOG-REF operating mode. The calibration operation for this axis is now complete.

18.5 Secondary conditions for absolute encoders

18.5.1 Calibrating absolute encoders

Calibration time

The calibration process determines the offset between the machine zero and the encoder zero and stores it in a non-volatile memory. Normally, calibration need only be performed once, i.e. during first commissioning. The control then knows the value and can calculate the absolute machine position from the encoder absolute value at any time. This status is identified by MD34210 ENC_REFP_STATE=2.

The offset is stored in MD34090 REFP_MOVE_DIST_CORR.

The calibration process must be repeated in the following situations:

- After mounting/removal or replacement of encoder or of motor with built-in encoder.
- After change of an existing gear unit between motor (with absolute encoder) and load.
- Generally speaking, every time the mechanical connection between the encoder and load is separated and not reconnected in exactly the same way.

NOTICE

The control may not always recognize the need for recalibration. If it detects such a need, it sets MD34210 to 0 or 1. The following is detected: changeover to another gear speed with a different gear ratio between the encoder and load.

In all other cases, the user must overwrite MD34210.

Data backup

When machine data is backed up, the status of MD34210 ENC_REFP_STATE is also saved. By loading this data set, the axis is automatically deemed calibrated!

 **WARNING**

If the data set has been taken from another machine (e.g. series startup), calibration must still be carried out after loading and activating the data.

18.6 Data lists

18.6.1 Machine data

| Number | Identifier | Name |
|------------------|-------------------------------|--|
| Channel-specific | | |
| 20700 | REFP_NC_START_LOCK | NC-Start disable without reference point |
| Axis-specific | | |
| 30200 | NUM_ENCS | Number of encoders |
| 30240 | ENC_TYP[0] | Actual value encoder type |
| 30330 | MODULO_RANGE | Magnitude of modulo range |
| 30340 | MODULO_RANGE_START | Starting position of modulo range |
| 31122 | BERO_DELAY_TIME_PLUS[0] | BERO delay time in plus direction |
| 31123 | BERO_DELAY_TIME_MINUS[0] | BERO delay time in minus direction |
| 34000 | REFP_CAM_IS_ACTIVE | Axis with reference cam |
| 34010 | REFP_CAM_DIR_IS_MINUS | Reference point approach in minus direction |
| 34020 | REFP_VELO_SEARCH_CAM | Reference point approach velocity |
| 34030 | REFP_MAX_CAM_DIST | Maximum distance to reference cam |
| 34040 | REFP_VELO_SEARCH_MARKER[0] | Reference point creep speed |
| 34050 | REFP_SEARCH_MARKER_REVERSE[0] | Direction reversal to reference cam |
| 34060 | REFP_MAX_MARKER_DIST[0] | Maximum distance to reference marker; maximum distance to 2 reference markers with distance-coded scales |
| 34070 | REFP_VELO_POS | Reference point positioning velocity |
| 34080 | REFP_MOVE_DIST[0] | Reference point distance/destination point for distancecoded system |
| 34090 | REFP_MOVE_DIST_CORR[0] | Reference point/absolute offset, distancecoded |
| 34092 | REFP_CAM_SHIFT[0] | Electronic reference point cam shift for incremental measuring systems with equidistant zero marks. |
| 34093 | REFP_CAM_MARKER_DIST[0] | Reference cam/reference mark distance |
| 34100 | REFP_SET_POS[0]...[3] | Reference point value |
| 34110 | REFP_CYCLE_NR | Axis sequence for channel-specific referencing |
| 34200 | ENC_REFP_MODE[0] | Referencing mode |
| 34210 | ENC_REFP_STATE[0] | Status of absolute encoder |
| 34220 | ENC_ABS_TURNS_MODULO[0] | Absolute encoder range for rotary encoders |
| 34330 | REFP_STOP_AT_ABS_MARKER[0] | Distancecoded linear measuring system without destination point |
| 36300 | ENC_FREQ_LIMIT[0] | Encoder frequency limit |
| 36302 | ENC_FREQ_LIMIT_LOW[0] | Encoder limit frequency resynchronization |
| 36310 | ENC_ZERO_MONITORING[0] | Zero mark monitoring |

18.6.2 Interface signals

| Number | Bit | Name |
|----------------------------|-----------|--------------------------------|
| Specific to operating mode | | |
| V3000 0001 | .2 | Machine function REF |
| V3100 0001 | .2 | Active machine function REF |
| Channel-specific | | |
| V3200 0001 | .0 | Activate referencing |
| V3300 0001 | .0 | Referencing active |
| V3300 0004 | .2 | All axes referenced |
| Axis-specific | | |
| V380x 0000 | .5 | Position measuring system 1 |
| V380x 0004 | .6 and .7 | Traversing key minus/plus |
| V380x 1000 | .7 | Reference point approach delay |
| V390x 0000 | .4 | Referenced, synchronizing 1 |
| V390x 0004 | .6 and .7 | Traverse command minus/plus |

Rotary Axes (R2)

19.1 General

Properties of a rotary axis

Rotary axes are always programmed in degrees. They are generally characterized by the fact that they assume the same position after exactly one rotation (modulo 360 degrees). Depending on the application in question, the traversing range of the rotary axis can be limited to less than 360 degrees (e.g. on swiveling axes for tool holders) or may be unlimited (e.g. when the tool or workpiece is rotated).

Definition of a rotary axis

An axis is declared as rotary axis with the aid of MD30300 IS_ROT_AX = 1.

Note

The geometry axes (X, Y, Z) **cannot** be used as rotary axes or spindle. MD20050 AXCONF_GEOAX_ASSIGN_TAB (assignment of geometry axis - channel axis) defines these geometry axes.

Axis addresses, axis identifiers, direction

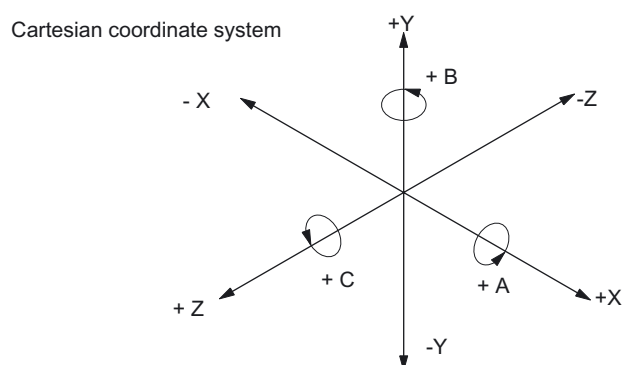


Figure 19-1 Identification of axes and positive direction of movement for rotary axes

For axes / rotary axes, the extended addressing (e.g. C2=) or an axis name can also be defined via configuration in the following machine data:

MD10000 AXCONF_MACHAX_NAME_TAB or

MD20080 AXCONF_CHANAX_NAME_TAB.

Units of measurement

The following units of measurement apply as standard to inputs and outputs for rotary axes:

Table 19- 1 Units of measurement for rotary axes

| Physical quantity | Units |
|--------------------------------|-------------------------|
| Angular position | Degrees. |
| Programmed angular speed | Degrees/min |
| MD for angular velocity | rev/min ¹⁾ |
| MD for angular acceleration | rev/sec ^{2 1)} |
| MD for angular jerk limitation | rev/sec ^{3 1)} |

¹⁾ In the case of axis-specific machine data, these units are interpreted by the control as soon as the axis is declared as a rotary axis.

More detailed information on the subject of dimension units can be found in Chapter "Velocities, Setpoint/Actual Value Systems, Closed-Loop Control (G2)"

Feedrate

The programmed feedrate F corresponds to an angular speed [degrees/min] for rotary axes programmed alone in a block.

If rotary axes and linear axes traverse along a common path with G94 or G95, the feed should be interpreted in the unit of measurement of the linear axes [e.g. mm/min, inch/min].

The tangential velocity of the rotary axis in this case refers to the diameter DE (unit diameter DE = 360/p mm, with p = circle constant).

If the diameter is the same as the unit diameter (D=DE), the programmed angular speed in degrees/min and the tangential velocity in mm/min are numerically identical. For the inch unit setting, "inch" applies instead of "mm".

In general, the following applies for tangential velocity:

$$F = F_w * D / D_E$$

with $D_E = 360 / \pi$

F = tangential velocity [mm/min]

F_w = angular speed [degrees/min]

D = diameter acted on by F [mm]

D_E = unit diameter [mm]

π = circle constant PI = 3.14...

JOG velocity for rotary axes

With SD41130 JOG_ROT_AX_SET_VELO (JOG velocity for rotary axes), a jog velocity that is valid for all rotary axes can be set.

If a value of 0 is entered in the setting data, then axis-specific MD32020 JOG_VELO (conventional axis velocity) functions as JOG velocity for the rotary axis.

Software limit switches

The software limit switches and working area limitations are active and are required for swivel axes with a limited operating range. However, in the case of continuously-turning rotary axes with MD30310 ROT_IS_MODULO=1, the software limit switches and working area limitations can be **deactivated** for individual axes.

19.2 Modulo 360 degrees

Term "Modulo 360 degrees"

With respect to a rotary axis, the term "Modulo" refers to mapping of the axis position internally in the control within the range from 0 to 359.999 degrees. With path defaults > 360° (e.g. for incremental programming using G91), the position is mapped in the range of values 0° to < 360° following conversion within the control. The mapping process is applied in JOG and AUTOMATIC mode. Exception: service display.

Machine data settings

Machine data can be used to define programming and positioning (MD30310 ROT_IS_MODULO) as well as the position display (MD30320 DISPLAY_IS_MODULO) individually in modulo 360° for each rotary axis, depending on the particular machine requirements.

Axis is modulo

MD30310 ROT_IS_MODULO = 1:

Activation of this machine data allows the special rotary-axis response to be utilized. The positioning action of the rotary axis for programming (G90, AC, ACP, ACN or DC). A modulo 360° representation is executed within the control after the current work offsets have been taken into account. The resulting target position is then approached **within a revolution**.

The software limit switches and working area limitations **are inactive**, meaning that the operating range is unlimited.

The modulo 360° position display should always be selected for a modulo axis.

MD30320 DISPLAY_IS_MODULO = 1

Modulo position display

MD30320 DISPLAY_IS_MODULO = 1:

For rotary axes, a position display with "modulo 360°" (one revolution) is often required, i.e. with a positive direction of rotation the display is periodically reset within the control to 0.000° after 359.999° is reached, with a negative direction of rotation the positions are also displayed in the range 0° to 359.999°.

MD30320 DISPLAY_IS_MODULO = 0:

Absolute-position display would, in the case of a positive direction of rotation, for example, result in $+360^\circ$ being displayed after one revolution, $+720^\circ$ after two revolutions, etc., in contrast to the modulo 360° display. In this case, the display range is limited by the control in accordance with the linear axes.

19.3 Programming rotary axes

Note

For general information about programming, please see:

Reference:

/BP_/ Operation and Programming

19.3.1 Rotary axis with active modulo conversion

Absolute programming (AC, ACP, ACN, G90)

Example with ACP: $C=ACP(5.33)$, general: **axis name =ACP(value)**

- The value identifies the target position of the rotary axis in a range from 0 to 359.999° . Alarm 16830, "Incorrect modulo position programmed", is output for values with a negative sign or $\geq 360^\circ$.
- ACP (positive) and ACN (negative) define the traversing direction of the rotary axis unambiguously (irrespective of actual position).
- When programming with AC or with G90, the traversing direction depends on the actual position of the rotary axis. If the target position is greater than the actual position, the axis traverses in the positive direction, otherwise it traverses in the negative direction.
- Use of ACP and ACN: With asymmetrical workpieces, it must be possible to define the traversing direction in order to prevent collisions during rotation.

Absolute programming along the shortest path (DC)

Example with DC: $C=DC(25.3)$, general: **axis name =DC(value)**

- The value identifies the target position of the rotary axis in a range from 0 to 359.999° degrees. Alarm 16830, "Incorrect modulo position programmed", is output for values with a negative sign or $\geq 360^\circ$.
- With DC (Direct Control), the rotary axis approaches the programmed absolute position within one revolution along the **shortest path** (traversing movement max. $\pm 180^\circ$).

- The control calculates the direction of rotation and the traverse path according to the current actual position. If the path to be traversed is the same in both directions (180°), the positive direction of rotation receives preference.
- Application example of DC: the rotary table is required to approach the changeover position in the shortest time (and, therefore, via the shortest path) possible.

Note

If DC is programmed with a linear axis, alarm 16800 "DC traversing instruction cannot be used" is output.

Incremental programming (IC, G91)

- The value identifies the rotary axis traversing distance. The value can be negative and $\geq 360^\circ$.
- The value's **sign** unequivocally defines the rotary-axis **traversing direction**.

Example:

```
C=IC(720)           ;C axis traverses to 720° incrementally in the positive direction
C=IC(-180)         ;C axis traverses to 180° incrementally in the negative direction
```

Endless traversing range

As soon as the modulo function is active, no limit is placed on the traversing range (software limit switches are not active). The rotary axis can now be programmed to traverse continuously.

Example:

```
N10 LOOP: C=IC(7200)
N20 GOTOB LOOP
```

19.3.2 Rotary axis without modulo conversion

Absolute dimension programming (AC, G90)

Example of AC: C=AC(-410), general: **axis name =AC (+/-value)**

- The value and its sign uniquely identify the rotary-axis target position. The value can be above $\pm 360^\circ$. The position value is limited by the software limit switch.
- The traversing direction is ascertained by the control according to the signed rotary-axis actual position.
- If ACP or ACN are programmed, alarm 16810 "ACP traversing instruction cannot be used" or alarm 16820 "ACN traversing instruction cannot be used" is output.

Absolute dimension programming along the shortest path (DC)

Example of DC: C=DC(60.3), general: **axis name =DC(value)**

Even if the rotary axis is not defined as a modulo axis, the axis can still be positioned with DC (direct control). The response is the same as on a modulo axis.

Incremental programming (IC, G91)

Example of IC: C=IC(-532.4) , general: **axis name =IC (+/-value)**

When programming with incremental dimensions, the rotary axis traverses across the same path as with the modulo axis. In this case, however, the traversing range is limited by the software limit switches.

Limited traversing range

The traversing range is limited as with linear axes. The range limits are defined by the plus and minus software limit switches.

19.4 Data lists

19.4.1 Machine data

| Number | Identifier | Name |
|-----------------------|-------------------------|---|
| General | | |
| 10000 | AXCONF_MACHAX_NAME_TAB | Machine axis name |
| 10210 | INT_INCR_PER_DEG | Computational resolution for angular positions |
| Channel-specific | | |
| 20050 | AXCONF_GEOAX_ASSIGN_TAB | Assignment between geometry axis and channel axis |
| 20080 | AXCONF_CHANAX_NAME_TAB | Channel axis name |
| Axis/spindle-specific | | |
| 30300 | IS_ROT_AX | Axis is rotary axis |
| 30310 | ROT_IS_MODULO | Modulo conversion for rotary axis |
| 30320 | DISPLAY_IS_MODULO | Actual value display modulo |
| 34220 | ENC_ABS_TURNS_MODULO | Range, rotary absolute encoders |
| 36100 | POS_LIMIT_MINUS | Minus software limit switch |
| 36110 | POS_LIMIT_PLUS | Plus software limit switch |

19.4.2 Setting data

| Number | Identifier | Name |
|---------------|----------------------|-------------------------------|
| General | | |
| 41130 | JOG_ROT_AX_SET_VELO | JOG velocity for rotary axes |
| Axis-specific | | |
| 43430 | WORKAREA_LIMIT_MINUS | Working area limitation minus |
| 43420 | WORKAREA_LIMIT_PLUS | Working area limitation plus |

Spindle (S1)

20.1 Brief description

Spindle functions

Depending on the machine type the following functions are possible for a spindle controlled by the NC:

- Input of a direction of rotation for the spindle (M3, M4)
- Input of a spindle speed (S)
- Spindle stop, without orientation (M5)
- Spindle positioning (SPOS=)
(position-controlled spindle required)
- Gear change (M40 to M45)
- Thread cutting/tapping (G33, G34, G35, G331, G332, G63)
- Revolutional feedrate (G95)
- Constant cutting rate (G96)
- Programmable spindle speed limits (G25, G26, LIMS=)
- Position encoder assembly on the spindle or on the spindle motor
- Spindle monitoring for min. and max. speed.
- Dwell time in spindle revolutions (G4 S)

An "enabled" spindle can be used instead of a controlled spindle. However, a spindle speed (S) is then **not** entered via the program but, for example, manually (gearbox) at the machine. This does not permit programming of speed limits. The following is possible via the program:

- Input of a direction of rotation for the spindle (M3, M4)
- Spindle stop, without orientation (M5)
- Tapping (G63)

If the spindle has a position encoder, the following functions are also available:

- Thread cutting/tapping (G33, G34, G35)
- Revolutional feedrate (G95)

If the spindle is enabled, the setpoint output for the spindle via MD30130 CTRLOUT_TYPE = 0 must be suppressed.

Definition of the spindle

A machine axis is declared a spindle by setting the following machine data:

- MD30300 IS_ROT_AX
- MD30310 ROT_IS_MODULO
- MD30320 DISPLAY_IS_MODULO
- MD35000 SPIND_ASSIGN_TO_MACHAX.

The IS "Spindle/no axis" reports the spindle mode (V390x 0000.0).

20.2 Spindle modes

Spindle modes

The spindle can have the following modes:

- Control mode, see Section "Spindle control mode"
- Oscillation mode, see Section "Spindle oscillation mode"
- Positioning mode, see Section "Spindle positioning mode"
- Axis mode, see Section "Spindle axis mode"
- Tapping without compensating chuck, see also Chapter "Feed (V1)"

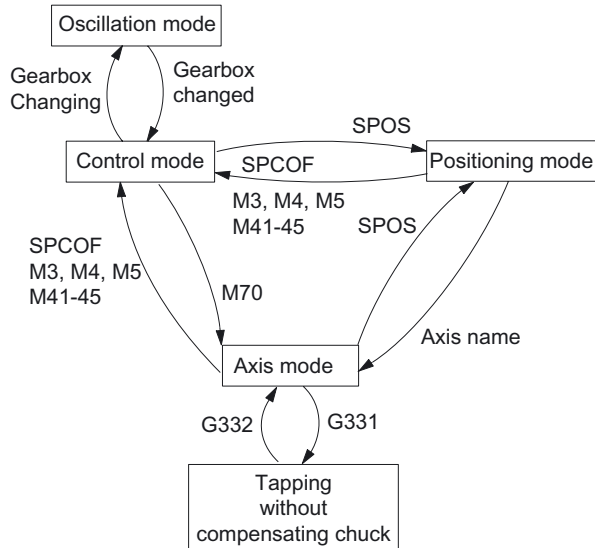


Figure 20-1 Switching between spindle modes

Switching between spindle modes

- Control mode ---> oscillation mode
The spindle changes to oscillation mode if a new gear stage has been specified using automatic gear stage selection (M40) in conjunction with a new S value or by M41 to M45. The spindle only changes to oscillation mode if the new gear stage is not equal to the current actual gear stage.
- Oscillation mode ---> control mode
When the new gear stage is engaged, the IS "Oscillation mode" (V390x 2002.6) is reset and the spindle is switched to control mode with the IS "Gear changed" (V380x 2000.3). The last programmed spindle speed (S value) is reactivated.
- Control mode ---> positioning mode
To stop the spindle from rotation (M3 or M4) with orientation or to reorient it from standstill (M5), SPOS, SPOSA or M19 are used to switch to positioning mode.
- Positioning mode ---> control mode
M3, M4 or M5 are used to change to control mode if the orientation of the spindle is to be terminated. The last programmed spindle speed (S value) is reactivated.
- Positioning mode ---> oscillation mode
If the orientation of the spindle is to be terminated, M41 to M45 can be used to change to oscillation mode. When the gear change is complete, the last programmed spindle speed (S value) and M5 (control mode) are reactivated.
- Positioning mode ---> tapping with compensation chuck
Tapping with compensation chuck (thread interpolation) is activated via G331/G332. SPOS must first be used to set the spindle to position-controlled operation.

20.2.1 Spindle control mode

When control mode?

The spindle is in control mode with the following functions:

- Constant spindle speed S, M3/M4/M5 and G94, G95, G97, G33, G63
- Constant cutting rate G96 S, M3/M4/M5

Requirements

A spindle position actual value sensor is absolutely essential for M2/M4/M5 in conjunction with revolution feedrate (G95, F in mm/rev or inch/rev), constant cutting rate (G96, G97), thread cutting (G33).

Independent spindle reset

MD35040 SPIND_ACTIVE_AFTER_RESET defines the response of the spindle after reset or program end (M2, M30):

- If MD value=0, the spindle is immediately braked to rest at the valid acceleration. The last programmed spindle speed and direction of rotation are deleted.
- If MD value = 1 (independent spindle reset), the last programmed spindle speed (S function) and the last programmed direction of spindle rotation (M3, M4, M5) are retained. If prior to reset or end of program the constant cutting speed (G96) is active, the current spindle speed (in relation to 100% spindle override) is internally accepted as the spindle speed last programmed.

| |
|---|
| NOTICE |
| The spindle can always be stopped with the IS "Delete distance-to-go / Spindle Reset". CAUTION: The program continues at G94! With G95 the axes stop due to the missing feedrates as does the program run if G1, G2, ... is active. |

20.2.2 Spindle oscillation mode

Note

The function for oscillation via the PLC is not available for the G/N plus and pro versions.

Starting oscillation mode

This oscillation movement makes it easy to engage a new gear stage. In principle, the new gear stage can also be engaged without oscillation

The spindle is in oscillation mode if a new gear stage was defined using the automatic gear stage selection (M40) or M41 to M45 (IS "Change gear" (V390x 2000.3) is enabled). The IS "Change gear" is only enabled when a new gear stage is selected that is not the same as the current actual gear stage. The spindle oscillation is started with the IS "Oscillation speed" (V380x 202.5).

If the IS "Oscillation speed" is enabled without defining a new gear stage, the spindle does not change to oscillation mode.

Oscillation is started with the IS "Oscillation speed". The setting of the IS "Oscillation via PLC" (V380x 2002.4) distinguishes between:

- Oscillation via NCK
- Oscillation via PLC

Oscillation time

The oscillation time for oscillation mode can be defined in a machine data for each direction of rotation:

- Oscillation time in M3 direction (referred to as t1 in the following):
MD35440 SPIND_OSCILL_TIME_CW
- Oscillation time in M4 direction (referred to as t2 in the following):
MD35450 SPIND_OSCILL_TIME_CCW

Oscillation via NCK

Phase 1: With the IS "Oscillation speed" (V380x 2002.5) , the spindle motor accelerates to the velocity (with oscillation acceleration) specified in MD35400 SPIND_OSCILL_DES_VELO (oscillation speed). The starting direction is defined by MD35430 SPIND_OSCILL_START_DIR (starting direction during oscillation).

Phase 2: If time t1 (t2) has elapsed, the spindle motor accelerates in the opposite direction to the speed defined in MD35400 SPIND_OSCILL_DES_VELO (oscillation speed). Time t2 (t1) starts.

Phase 3: When time t2 (t1) expires, the spindle motor accelerates in the opposite direction (same direction as phase 1), etc.

Oscillation via PLC

With the IS "Oscillation speed" (V380x 2002.5) , the spindle motor accelerates to the velocity (with oscillation acceleration) specified in MD35400 SPIND_OSCILL_DES_VELO (oscillation speed).

The direction of rotation is determined by IS "Set direction of rotation counterclockwise" or IS "Set direction of rotation clockwise" (V380x 2002.7 or .6).

The oscillation movement and the two times t1 and t2 (for clockwise and counterclockwise rotation) must be simulated on the PLC.

End of oscillation mode

The IS "Gear changed" (V380x 2000.3) informs the NC that the new gear stage (IS "Actual gear stage" (V380x 2000.0 to .2)) applies and oscillation mode is exited. The actual gear stage should correspond to the set gear stage. Oscillation mode is also ended if the IS "oscillation speed" (V380x 2002.5) is still set. The last programmed spindle speed (S function) and spindle rotation (M3, M4 or M5) are active again.

After termination of oscillation mode the spindle returns to control mode.

All gearspecific limit values (min./max. speed etc.) correspond to the set values of the actual gear stage and are deactivated when the spindle stops.

Block change

If the spindle has been changed over to oscillation mode, IS "Change gear" (V390x 2000.3) is set, part program processing is stopped. A new block is not executed. If oscillation mode is terminated using the IS "Gear switched" (V380x 2000.3), the execution of the part program is continued. A new block is executed.

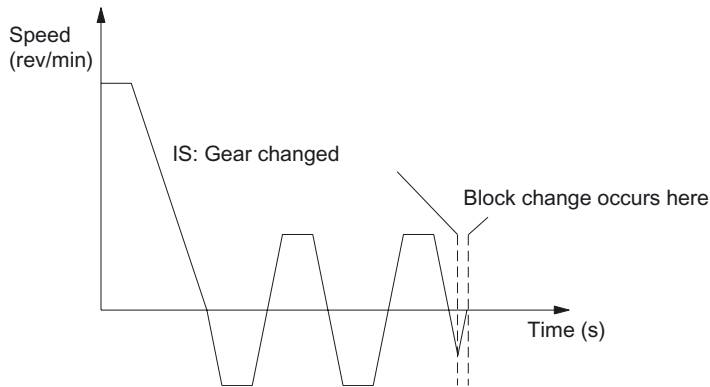


Figure 20-2 Block change following oscillation mode

Special features

- The acceleration is defined by MD35410 SPIND_OSCILL_ACCEL (acceleration during oscillation).
- If the IS "oscillation speed" (V380x 2002.5) is reset, the oscillation stops. However, the spindle **remains** in oscillation mode.
- The IS "Gear changed" should always be used for terminating gear stage change.
- The IS "Reset" (V3000 0000.7) does **not** terminate oscillation mode.
- If an indirect measuring system is used, synchronization is lost. The spindle is re-synchronized the next time the zero mark is crossed.

Reset during gear stage change

The spindle cannot be stopped via IS "Reset" (V3000 0000.7) or IS "NC Stop" (V3200 0007.3) if the spindle is in oscillation mode for gear stage change and the IS "Gear changed" (V380x 2000.3) is not yet available.

In this case, alarm 10640 "Stop not possible during gear change" is displayed if reset is selected. After changing the gear stages, the reset request is performed and the alarm cleared, if this is still present at the interface.

Note

Option for aborting: Set IS "Delete distance-to-go / Spindle Reset" (V380x 0002.2).

20.2.3 Spindle positioning mode

When is positioning mode used?

The spindle positioning mode stops the spindle at the defined position and activates the position control, which remains active until it is deactivated. With the **SPOS =.....** program function, the spindle is in positioning mode (see also Section "Programming").

Block change

The block change is carried out when all functions programmed in the block have reached their end criterion (e.g. axis traverse completed, all auxiliary functions acknowledged by PLC) **and** the spindle has reached its position (IS "Exact stop fine" for spindle (V390x 0000.7)).

Requirements

A spindle position actual value encoder is absolutely essential.

Positioning from rotation

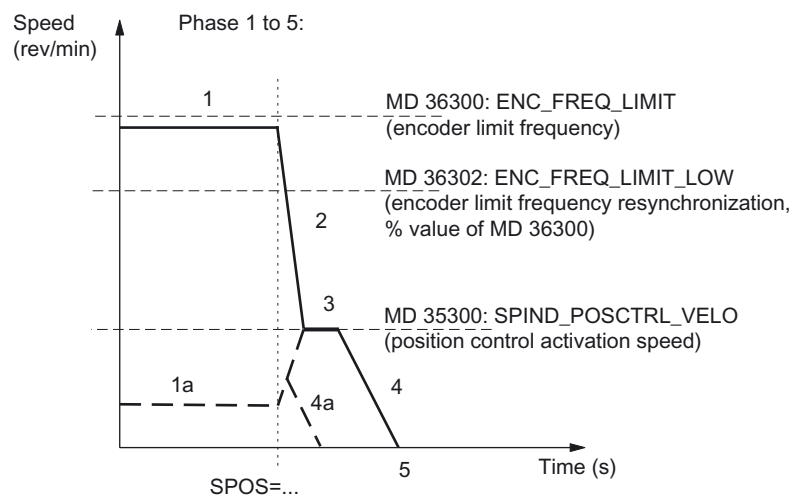


Figure 20-3 Positioning from rotation at different speeds

Sequence

Phase 1: Spindle rotates at a lower speed than the encoder limit frequency. The spindle is synchronized. It is set to control mode. Process continues with Phase 2.

Phase 1a: Spindle rotates at a lower speed than the position control activation speed. The spindle is synchronized. It is set to control mode. The rest of the sequence is possible via 4a.

Phase 1b (not shown): Spindle rotates at a speed higher than the encoder limit frequency. The spindle is not synchronized initially, but is then synchronized when the rotation speed falls below the speed defined by the encoder frequency in MD36302 ENC_FREQ_LIMIT_LOW (% value of MD36300). Sequence continues with Phase 2.

Phase 2: When the SPOS command takes effect, the spindle starts to decelerate with the acceleration stored in MD35200 GEAR_STEP_SPEEDCTRL_ACCEL until it reaches the position control activation speed.

Phase 3: When the position-control activation speed stored in MD35300 SPIND_POSCTRL_VELO is reached:

- The position control is activated
- The distance-to-go (to target position) is calculated, (easier from Phase 1a)
- The acceleration is switched to MD35210 GEAR_STEP_POSCTRL_ACCEL (acceleration in position control mode) (always active below the position control activation speed)

Phase 4: The spindle brakes from the calculated "braking point" with MD35210 GEAR_STEP_POSCTRL_ACCEL to the target position.

Phase 5: The position control remains active and stops the spindle in the programmed position. The IS "Position reached with exact stop fine" (V390x 0000.7) and "... coarse" (V390x 0000.6) are set if the distance between the spindle actual position and the programmed position (spindle setpoint position) is less than the settings for the exact stop fine and coarse limits (defined in MD36010 STOP_LIMIT_FINE and MD36000 STOP_LIMIT_COARSE).

Positioning from standstill, spindle not synchronized

The spindle is not synchronized after the control has been activated. The first movement of the spindle must be positioning (SPOS=...).

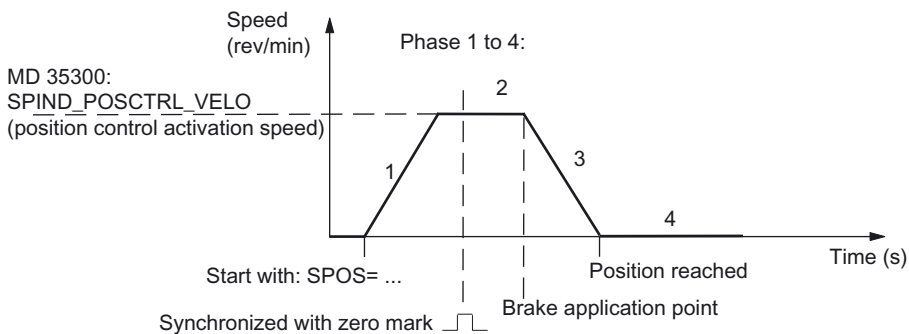


Figure 20-4 Positioning with stopped, non-synchronized spindle

Sequence

Phase 1: Programming SPOS accelerates the spindle with the acceleration in MD35210 GEAR_STEP_POSCTRL_ACCEL (acceleration in position control mode) until the maximum speed entered in MD35300 SPIND_POSCTRL_VELO (position control activation speed) is reached.

The direction of rotation is defined by MD35350 SPIND_POSITIONING_DIR (direction of rotation during positioning from standstill), if no input results from SPOS programming (ACN, ACP, IC). The spindle is synchronized with the next zero mark of the position actual value encoder.

Phase 2: When the spindle is synchronized, the position control is activated. The spindle rotates at the maximum speed stored in MD35300 SPIND_POSCTRL_VELO until the braking start point calculation identifies the point at which the programmed spindle position can be approached accurately with the defined acceleration.

Phase 3: At the brake application point, the spindle is braked down to standstill with the acceleration set in MD35210 GEAR_STEP_POSCTRL_ACCEL (acceleration in position control mode).

Phase 4: The spindle has reached the target point and is stationary. The position control is active and stops the spindle in the programmed position. The IS "Position reached with exact stop fine" (V390x 0000.7) and "... coarse" (V390x 0000.6) are set if the distance between the spindle actual position and the programmed position (spindle setpoint position) is less than the settings for the exact stop fine and coarse limits (MD36010 STOP_LIMIT_FINE and MD36000 STOP_LIMIT_COARSE).

Positioning from standstill, spindle is synchronized

The spindle has already been turned by one spindle revolution with M3 and M4 and was then brought to a standstill with M5.

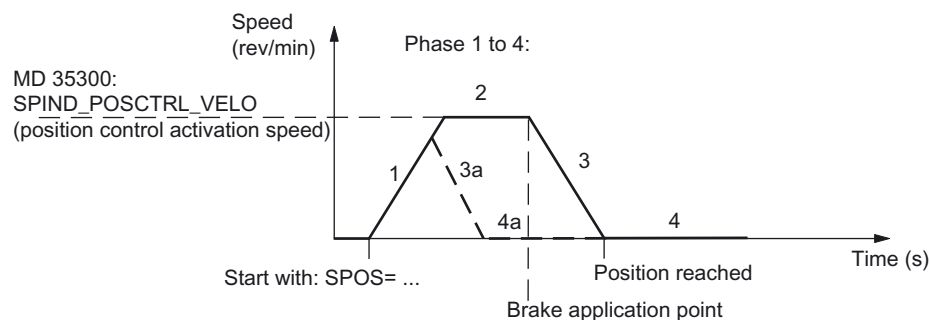


Figure 20-5 Positioning with stationary, synchronized spindle

Sequence

The spindle travels to the programmed end point optimally in terms of time. Depending on the appropriate secondary conditions, the operational sequences in phases 1 - 2 - 3 - 4 or 1 - 3a - 4a are executed.

Phase 1: SPOS will switch the spindle to position control mode. The acceleration from MD35210 GEAR_STEP_POSCTRL_ACCEL (acceleration in the position control mode) is activated. The direction of rotation is determined by the relevant distance-to-go (type of path setting with SPOS).

The speed entered in MD35300 SPIND_POSCTRL_VELO (position control activation speed) is not exceeded. The traversing path to the end point is calculated. If the end point can be accessed immediately from this phase, Phase 3a, 4a continues instead of Phase 2.

Phase 2: Acceleration has been performed up to the speed set in MD35300 SPIND_POSCTRL_VELO (position control activation speed). The brake application point calculation identifies when the programmed spindle position (SPOS=...) can be approached with the acceleration defined in MD35210 GEAR_STEP_POSCTRL_ACCEL.

Phase 3 and Phase 4: The sequence for "Deceleration" and "Position reached" is the same as for non-synchronized spindles.

Spindle reset

The positioning process can be aborted with the IS "Delete distancetogo/spindle reset" (V380x 0002.2). However, the spindle remains in positioning mode.

Notes

- In positioning mode the spindle speed override switch continues to be valid.
- Positioning (SPOS) is cancelled with "Reset" or "NC stop".

20.2.4 Spindle axis mode

When axis mode?

For certain machining tasks on lathes: For face machining with TRANSMIT or peripheral surface machining with TRACYL the spindle must be operated as rotary axis.

Outside of TRANSMIT and TRACYL, you can perform normal rotary axis functions. In this case it is programmed under the address of the rotary axis, e.g. C.

Requirements

The spindle can be switched from spindle mode to axis mode (rotary axis) if the same motor is used for spindle and axis operation.

A position actual value encoder is a mandatory requirement for axis mode.

Activate/deactivate

- Set spindle to position-controlled mode (positioning mode):
N10 SPOS=0
- If the spindle is synchronized, this can also be performed with M70 or SPCON. Traversal as rotary axis is now possible in the program:
N20 G94 G1 C124.4 F4000; feedrate F: 4000 degr./min
- Return to speed-controlled spindle mode is performed with:
N100 M3; or M4, M5, M41 ... M45 or SPCOF

Special features

- The feed override switch is active.
- RESET does not terminate the axis mode per default.
- IS "Spindle / no axis" (V390x 0000.0) is set to zero.
- Axis mode can be activated in all gear stages.

- The gear stage cannot be changed when the axis mode is active.
- In axis mode, the MD of the set of parameters with index zero are effective in order to carry out adaptation in this mode.

20.3 Synchronization

Why synchronize?

The control must be synchronized with the position measurement system on the spindle so that the control knows the exact 0 degree position when switched on. Only a synchronized spindle is capable of thread cutting or positioning.

For axes, this process is referred to as referencing, see Chapter "Reference Point Approach (R1)".

Installation position of the position measurement system

- Directly on the motor in combination with a BERO proximity switch on the spindle (zero mark encoder)
- Directly on the spindle
- Above the measuring gearbox plus BERO switch on the spindle.

Synchronization possibilities

When the spindle is switched on, the spindle can be synchronized as follows:

- The spindle is started with a spindle speed (S function) and a spindle rotation (M3 or M4), and synchronized with the next zero mark of the position measurement system or with the BERO signal. The 0 degree position is shifted by

MD34080 REFP_MOVE_DIST + MD34090 REFP_MOVE_DIST_CORR -
MD34100 REFP_SET_POS.

Note

Only use MD34080 for shifting the 0 degree position. Monitoring with MD34060 REFP_MAX_MARKER_DIST should be set to two spindle revolutions (720 degrees).

- Programming SPOS=... from various states (refer to Section "Spindle positioning mode")
- In JOG mode the spindle is started in speed control mode with the direction keys and synchronizes with the next zero mark of the position measurement system or the BERO signal.

Value acceptance

When synchronizing the spindle, the associated reference point from MD34100 REFP_SET_POS[0] (default value = 0) is transferred and a possible shift of the reference point. These shifts (machine data) act irrespective of the connected measurement system and are described in Chapter "Reference Point Approach (R1)".

Maximum encoder frequency exceeded

If the spindle speed reaches a speed (large S value programmed), which exceeds the maximum encoder limit frequency MD36300 ENC_FREQ_LIMIT (the maximum mechanical speed limit of the encoder must not be exceeded), the synchronization is lost. The spindle continues to rotate, but with reduced functionality.

If a speed is then reached that is below the encoder limit frequency in MD36302 ENC_FREQ_LIMIT_LOW (% value of MD36300), the spindle automatically synchronizes with the next zero mark signal. You can achieve this by programming a lower S value, changing the spindle speed override switch, etc.

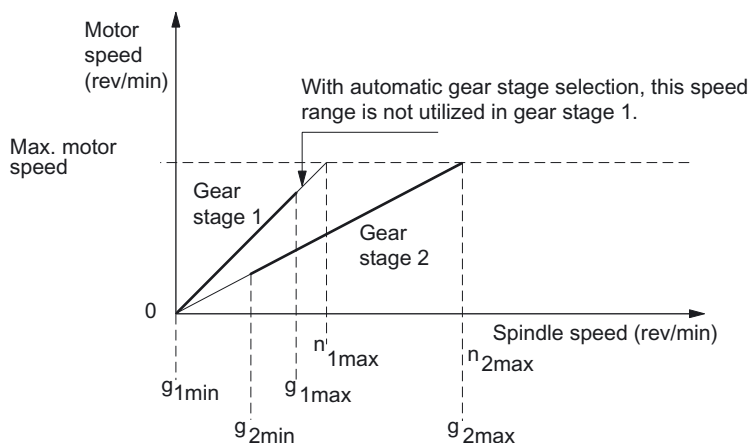
Re-synchronizing

In the following case, however, the position measuring system must be re-synchronized: the position measurement encoder is on the motor, a BERO (distance sensor for synchronization signals) is mounted to the spindle and the gear stage is changed. The synchronization is triggered internally when the spindle is rotating in the new gear stage.

20.4 Gear stage change

Number of gear stages

Five gear stages can be configured for the spindle. If the spindle motor is mounted on the spindle directly (1:1) or with a non-adjustable gear ratio, MD35010 GEAR_STEP_CHANGE_ENABLE (gear stage change is possible) must be set to zero.



| | | |
|------------------|----------------|--|
| Definable by MD: | n_{1max} ... | max. spindle speed of the 1st gear stage |
| | g_{1min} ... | min. spindle speed of the 1st gear stage for autom. gear stage selection |
| | g_{1max} ... | max. spindle speed of the 1st gear stage for autom. gear stage selection |
| | n_{2max} ... | max. spindle speed of the 2nd gear stage |
| | g_{2min} ... | min. spindle speed of the 2nd gear stage for autom. gear stage selection |
| | g_{2max} ... | max. spindle speed of the 2nd gear stage for autom. gear stage selection |

Figure 20-6 Gear stage change with gear stage selection

Defining a gear stage

A gear stage can be defined as follows:

- Permanent definition in the part program (M41 to M45)
- Automatic definition by the programmed spindle speed (M40)

In the case of M40, the spindle must be in control mode for automatic gear stage selection with an S value. The gear stage change is otherwise rejected and alarm 22000 "Gear change not possible" is output.

M41 to M45

The gear stage can be permanently defined in the part program with M41 to M45. If a gear stage is defined by M41 to M45, which is different than the current (actual) gear stage, the IS "Change gear" (V390x 2000.3) and the IS "Set gear stage A" to "...C" (V390x 2000.0 to .2) are set. The programmed spindle speed (S) then refers to this permanently defined gear stage. If a spindle speed exceeding the maximum speed of the permanently defined gear stage is programmed, the speed is limited to the maximum speed of this gear stage and the

IS "Programmed speed too high" (V390x 2001.1) is enabled. If a speed is programmed lower than the minimum speed of this gear stage, the speed is raised to this speed. The IS "Setpoint speed increased" (V390x 2001.2) is then enabled.

M40

M40 in the part program causes the gear stage to be selected automatically by the control. The control checks which gear stage is possible for the programmed spindle speed (S function). If the suggested gear stage is not equal to the current (actual) gear stage, the IS "Change gear" (V390x 2000.3) and the IS "Set gear stage A to C" (V390x 2000.0 to .2) are enabled.

The automatic gear stage selection function initially compares the programmed spindle speed with the minimum and maximum speed of the current gear stage. If the comparison is positive, a new gear stage is not defined. If the comparison is negative, the comparison is performed on each of the gear stages (starting with gear stage 1) until the result is positive. If the comparison in the 5th gear stage is also not positive, no gear stage change is triggered. If necessary the speed is limited to the maximum speed of the current gear stage or increased to the minimum speed of the current gear stage, and the IS "Setpoint speed limited" (V390x 2001.1) or IS "Setpoint speed increased" (V390x 2001.2) is enabled.

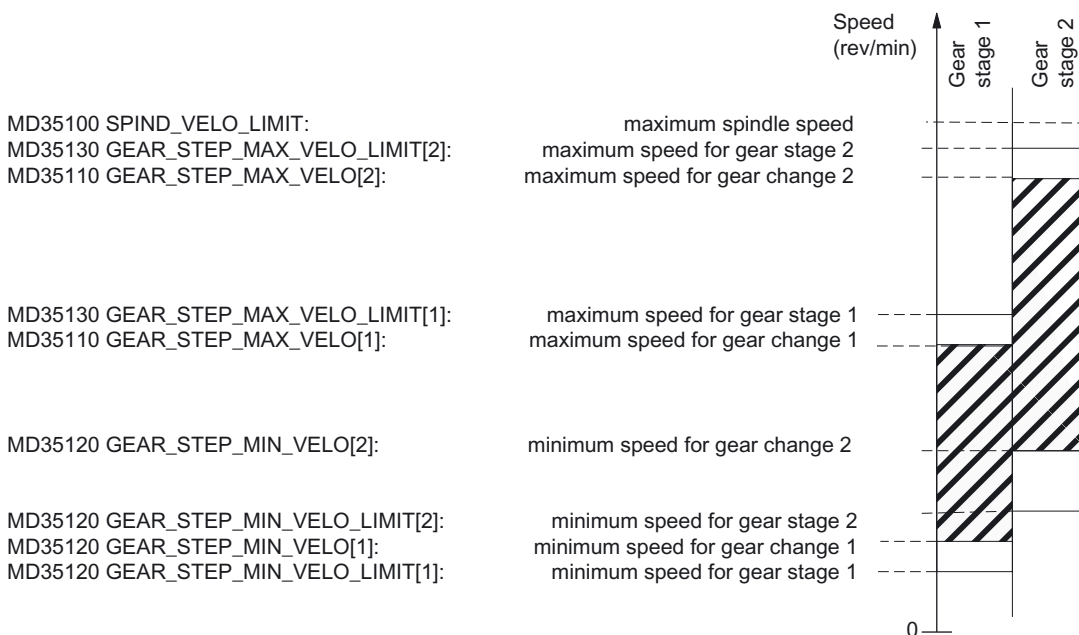


Figure 20-7 Example for speed ranges for automatic gear stage selection (M40)

Gear stage change

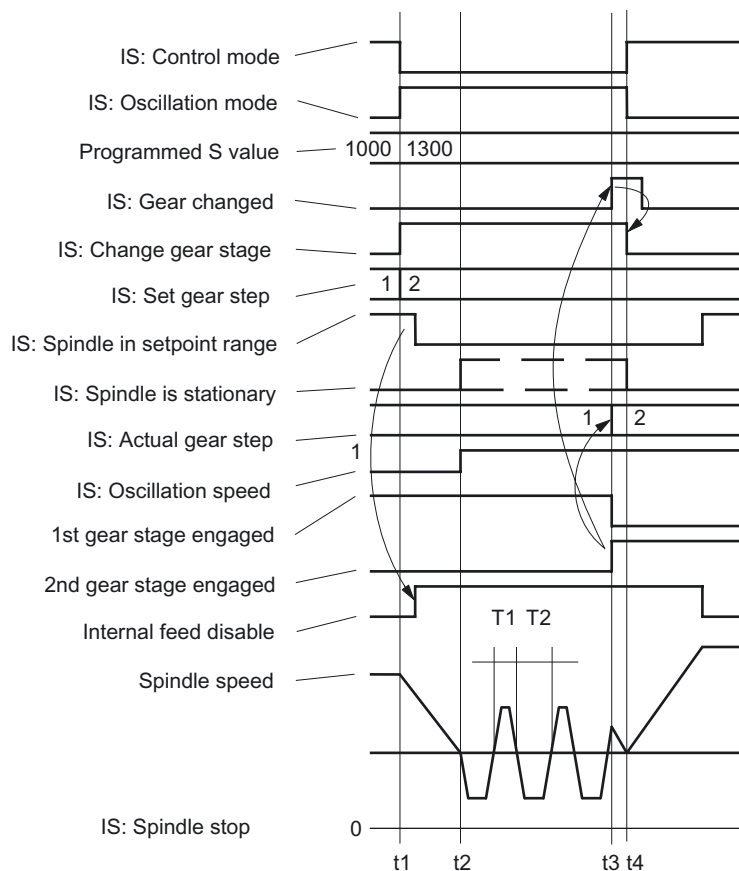
A new gear stage can only be selected when the spindle is stationary. The spindle is stopped internally in the control if a gear stage change is requested. If the new gear stage is preselected by M40 and spindle speed or M41 to M45, the IS "Set gear stage A" to "...C" (V390x 2000.0 to .2) and the IS "Change gear" (V390x 2000.4) are set. At the point when the IS "Oscillation speed" (V380x 2002.5) is enabled, the spindle decelerates to a

stop with the acceleration for oscillation or with the acceleration for speed control / position control.

The next block in the part program **after** the gear stage change via M40 and S value or M41 to M45 is not performed (same effect as if the IS "Read-in disable" (V3200 0006.1) were enabled).

When stationary the spindle (IS "Axis/spindle stationary" (V390x 0001.4)) can be activated with the IS "Oscillation speed" (V380x.2002.5) (see Section "Spindle oscillation mode"). When the new gear stage is engaged, the PLC user sets the IS "Actual gear stage" (V380x 2000.0 to .2) and IS "Gear changed" (V380x 2000.3). The gear stage change is considered completed (spindle mode "Oscillation mode" is deselected) and the spindle is switched to the parameter block of the new actual gear stage. The spindle accelerates at the new gear stage to the spindle speed last programmed (if M3 or M4 are active). The IS "Change gear" (V390x 2000.3) is reset by the NCK, which causes the PLC user to reset the IS "Gear changed" (V380x 2000.3). The next block in the part program can be executed.

Typical time sequence for the gear stage change:



- t1 When S1300 is programmed, the NCK detects a new gear stage (2nd gear stage), enables IS: Change gear and inhibits execution of the next part program block.
- t2 The spindle is stationary and oscillation starts (oscillation via NCK). The Oscillation speed interface signal must be enabled by the time t2.
- t3 The new gear stage is engaged. The PLC user transmits the new (actual) Gear stage to the NCK and sets the IS: gear changed.
- t4 At this point, the NCK cancels the Change gear interface signal, terminates oscillation, enables execution of the next part program block and accelerates the spindle to the new S value (S1300).

Figure 20-8 Gear stage change with stationary spindle

Parameter set

One parameter set each is provided for each of the five gear stages. The appropriate parameter set is activated using the IS "Actual gear stage A" to "...C" (V380x 2000.0 to .2). It is assigned as follows:

| Index n | PLC interface, CBA coding | Data of the data set | Contents |
|---------|---------------------------|-------------------------|--|
| 0 | - | Data for axis mode | Servo gain factor, monitoring functions, speed, acceleration, etc. |
| 1 | 000 001 | Data for 1st gear stage | |
| 2 | 010 | Data for 2nd gear stage | |
| 3 | 011 | Data for 3rd gear stage | |
| 4 | 100 | Data for 4th gear stage | |
| 5 | 101 | Data for 5th gear stage | |

The machine data included in a parameter set are marked specifically in Section "Machine Data". The following machine data is added per gear stage for each parameter set index n (n=1 -> 1st gear stage of the spindle, etc.):

- MD35110 GEAR_STEP_MAX_VELO[n]
- MD35120 GEAR_STEP_MIN_VELO[n]
- MD35130 GEAR_STEP_MAX_VELO_LIMIT[n]
- MD35140 GEAR_STEP_MIN_VELO_LIMIT[n]
- MD35200 GEAR_STEP_SPEEDCTRL_ACCEL[n]
- MD35210 GEAR_STEP_POSCTRL_ACCEL[n]
- MD35310 SPIND_POSIT_DELAY_TIME[n]

20.5 Programming

Functions

The spindle can be set for the following functions:

- G95 Revolutional feedrate
- G96 S... LIMS=... Constant cutting rate in m/min, upper speed limit
- G97 Cancel G96 and freeze last spindle speed
- G33, G331, G332 Thread cutting, tapping
- G4 S ... Dwell time in spindle revolutions

| | |
|------|-----------------------------------|
| M3 | CW spindle rotation |
| M4 | CCW spindle rotation |
| M5 | Spindle stop, without orientation |
| S... | Spindle speed in rpm, e.g. S300 |

| | |
|---------------|---|
| SPOS=... | Spindle positioning, e.g. SPOS=270 -> at position 270 degrees. The block change is only performed when the spindle is in position. |
| SPOS=DC(Pos) | The direction of motion is retained for positioning while in motion and the position approached. When positioning from standstill, the position is approached via the shortest path. |
| SPOS=ACN(Pos) | The position is always approached with negative direction of motion. If necessary, the direction of motion is inverted prior to positioning. |
| SPOS=ACP(Pos) | The position is always approached with positive direction of motion. If necessary, the direction of motion is inverted prior to positioning. |
| SPOS=IC(Pos) | The traversing path is specified. The direction of traversing is obtained from the sign in front of the traversing path. If the spindle is in motion, the direction of traversing is inverted if necessary to allow traversing in the programmed direction. |
| M40 | Automatic gear stage selection for the spindle |
| M41 to M45 | Select gear stage 1 to 5 for the spindle |
| SPCON | Position control on |
| SPCOF | Position control off |
| M70 | Position control on |
| G25 S... | Programmable lower spindle speed limit, e.g. G25 S8 |
| G26 S... | Programmable upper spindle speed limit, e.g. G26 S1200 |
| LIMS=... | Programmable maximum spindle speed for G96 |

Reference:

/BP_/_ Operation and Programming

20.6 Spindle monitoring

Speed ranges

The spindle monitoring functions and the currently active functions (G94, G95, G96, G33, G331, G332, etc.) define the admissible speed ranges of the spindle.

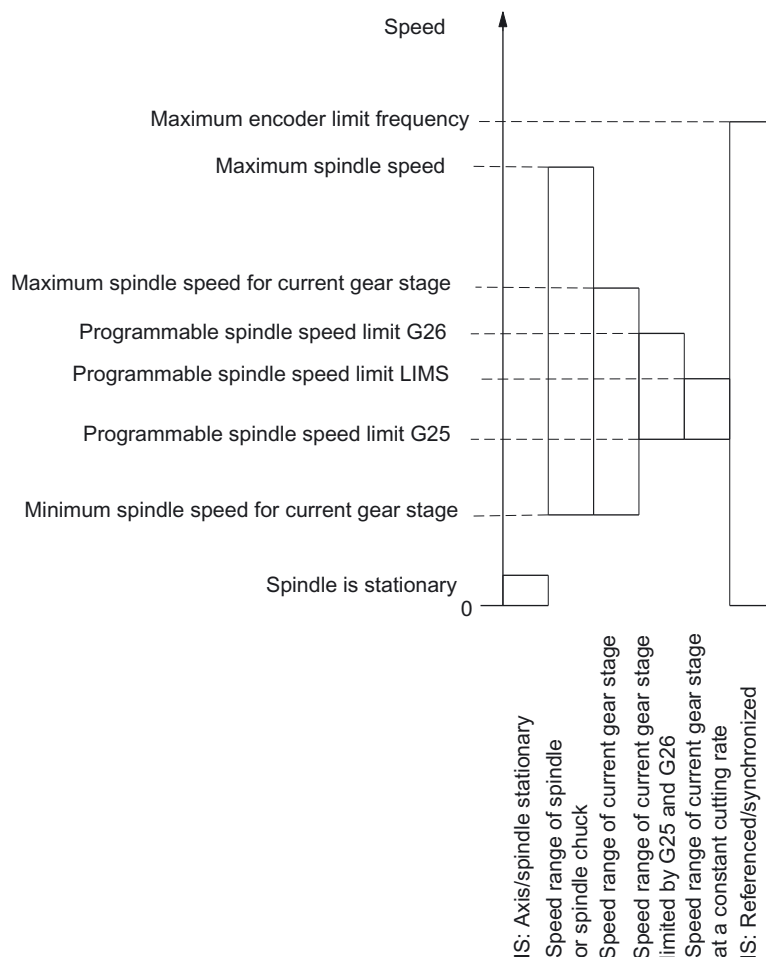


Figure 20-9 Ranges of spindle monitoring functions / speeds

20.6.1 Axis/spindle stationary

Only when the spindle is stationary, i.e. the actual spindle speed is below a value defined in MD36060 STANDSTILL_VELO_TOL, is IS "Axis/spindle stationary" (V390x 0001.4) set. Functions such as tool change, open machine door, path feed can be activated using the PLC user program.

Monitoring is effective in the three spindle modes.

20.6.2 Spindle in setpoint range

The "Spindle in setpoint range" monitor checks whether the programmed spindle speed has been reached, whether the spindle is stationary (IS "Axis/spindle stationary") or whether it is still in the acceleration phase.

In the spindle "control mode", the speed setpoint (programmed speed with spindle override including the active limits) is compared with the actual speed. If the deviation of the actual speed from the speed setpoint is greater than the spindle speed tolerance set in MD35150 SPIND_DES_VELO_TOL:

- IS "Spindle in setpoint range" (V390x 2001.5) is set to zero.
- The next machining block is not enabled if MD35500 SPIND_ON_SPEED_AT_IPO_START is set.

20.6.3 Maximum spindle speed

Maximum spindle speed

A maximum speed is defined for "maximum spindle speed" spindle monitoring, which the spindle may not exceed.

The maximum spindle speed is entered in MD35100 SPIND_VELO_LIMIT.

The control limits an excessive spindle speed setpoint to this value. If the actual spindle speed exceeds the maximum spindle speed despite allowance for the spindle speed tolerance (MD35150 SPIND_DES_VELO_TOL), there is a drive fault and IS "Speed limit exceeded" (V390x 2002.0) is set. Furthermore the alarm 22100 is output and all axes and the spindle are decelerated.

Speed limiting the PLC

The spindle speed can be limited to a specific value via the PLC. This value is in MD35160 SPIND_EXTERN_VELO_UNIT and is activated via IS "Velocity/spindle speed limitation" (V380x 0003.6).

20.6.4 Minimum/maximum speed for gear stage

Max. speed

MD35130 GEAR_STEP_MAX_VELO_LIMIT defines the maximum speed for the gear stage. In the gear stage engaged, this set speed can never be exceeded. When the programmed spindle speed is limited, the IS "Set speed limited" (V390x 2001.1) is enabled.

Minimum speed

MD35140 GEAR_STEP_MIN_VELO_LIMIT defines the minimum speed for the gear stage. It is not possible that the speed falls below this (set) speed if an S value is programmed, which is too small. Then, the IS "Setpoint speed increased" (V390x 2001.2) is enabled.

The minimum gear stage speed is operative only in spindle open loop control mode; the speed of the gear stage can only fall below the minimum limit through:

- Spindle override 0 %
- M5
- S0
- IS "Spindle stop"
- Remove IS "Controller enable"
- IS "Reset"
- IS "Spindle reset"
- IS "Oscillation speed"
- IS "NCSTOP axes and spindle"
- IS "Axis/spindle disable"

20.6.5 Max. encoder limit frequency

| |
|--|
|  WARNING |
| The maximum encoder limit frequency of the actual spindle position encoder is monitored by the control (the limit can be exceeded). It is the responsibility of the machine tool manufacturer to ensure that the configuration of the spindle motor, gearbox, measuring gearbox, encoder and machine data prevents the maximum speed of the actual spindle position encoder from being exceeded. |

Maximum encoder limit frequency exceeded

If the spindle reaches a speed in the open-loop control mode (a high S value has been programmed) which is higher than the max. encoder limit frequency (the max. speed of the encoder may not be exceeded), the synchronization is lost. However, the spindle continues to rotate.

If one of the thread cutting (G33), revolutional feedrate (G95), constant cutting rate (G96, G97) functions is programmed, the spindle speed is reduced automatically so far until the active measuring system is working reliably again.

In the "positioning mode" spindle mode and with position-controlled threads (G331, G332) the max. encoder limit frequency is not exceeded.

If the encoder limit frequency is exceeded, the IS "Referenced/synchronized" V390x 0000.4) is reset for the measurement system and IS "Encoder limit frequency 1 exceeded" (V390x 0000.2) is enabled.

If the maximum encoder limit frequency has been exceeded and the speed subsequently falls below the encoder frequency in MD36302 ENC_FREQ_LIMIT_LOW (% value of MD36300 ENC_FREQ_LIMIT), the spindle is automatically synchronized with the next zero mark or the next BERO signal.

20.6.6 Target point monitoring

Function

During positioning (the spindle is in "positioning mode"), the system monitors the distance from the spindle (with reference to the actual position) to the programmed spindle set position (target point).

Two limit values can be defined as incremental path (starting at the spindle set position) in the following machine data.

- MD36000 STOP_LIMIT_COARSE (exact stop limit coarse)
- MD36010 STOP_LIMIT_FINE (exact stop limit fine)

Regardless of the two limit values, the positioning of the spindle is always as accurate as the connected spindle measurement encoder, the backlash, the transmission ratio, etc.

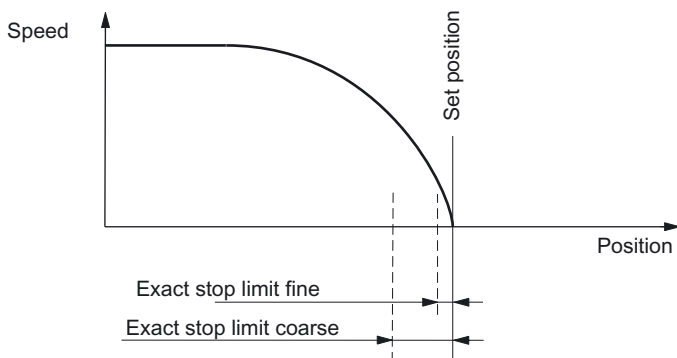


Figure 20-10 Exact stop zones of a spindle for positioning

IS: Position reached with exact stop ...

When the limits MD 36000 and MD 36010 are reached, IS "Position reached with exact stop coarse" (V390x 0000.6) and IS "Position reached with exact stop fine" (V390x 0000.7) are output to the PLC.

Block change with SPOS

If the spindle is being positioned with SPOS, the block change will be dependent on the end point monitoring with the IS "Position reached with exact stop fine". All other functions programmed in the block must have achieved their end criterion (e.g. axes ready, all auxiliary functions acknowledged by the PLC).

20.7 2nd spindle / master spindle

Note

This function is not available for the 802D sl value.

Function

The TRANSMIT and TRACYL kinematic transformation functions are available for milling on lathes. These functions require a second spindle for the driven milling tool. The main spindle is operated as a rotary axis with these functions.

Master spindle

A series of functions is connected to the master spindle, which are only possible for this spindle:

| | |
|---------------------------|---|
| G95 | Rev. feedrate |
| G96, G97 | Constant cutting rate |
| LIMS | Upper speed limit for G96, G97 |
| G33, G34, G35, G331, G332 | Thread cutting, thread interpolation |
| M3, M4, M5, S... | Simple specifications for direction of rotation, stop and speed |

The master spindle is defined via configuration (machine data). Generally it is the main spindle (spindle 1).

A different spindle can be defined as master spindle in the program:

SETMS(n) Spindle n (= 1 or 2) is the master spindle as of now

Switching back can also be performed via:

SETMS Configured master spindle is now master spindle again
or

SETMS (1) Spindle 1 is now master spindle again

The definition of the master spindle changed in the program is only valid until End of program/program abort. Thereafter, the configured master spindle is again active.

Programming via spindle number

Some spindle functions can also be selected via the spindle number:

| | |
|-------------------|---|
| S1=..., S2=... | Spindle speed for spindle 1 or 2 |
| M1=3, M1=4, M1=5 | Specifications for direction of rotation, stop for spindle 1 |
| M2=3, M2=4, M2=5 | Specifications for direction of rotation, stop for spindle 2 |
| M1=40, ..., M1=45 | Gear stages for spindle 1 (if available) |
| M2=40, ..., M2=45 | Gear stages for spindle 2 (if available) |
| SPOS [n] | Position spindle n |
| SPI(n) | Converts the spindle number n into the axis identifier, e.g. "SP1" or "CC" n must be a valid spindle number (1 or 2) The spindle identifiers SPI(n) and Sn are identical in terms of function |
| P_S[n] | Last programmed speed of spindle n |
| \$AA_S[n] | Actual speed of spindle n |
| \$P_SDIR[n] | Last programmed direction of rotation of spindle n |
| \$AC_SDIR[n] | Current direction of rotation of spindle n |

Two spindles available

The following can be interrogated in the program via the system variable:

| | |
|------------------|--|
| \$P_NUM_SPINDLES | Number of configured spindles (in the channel) |
| \$P_MSNUM | Number of programmed master spindle |
| \$AC_MSNUM | Number of active master spindle |

20.8 Analog spindle**Function**

With the "Analog spindle" function, the MCPA option module is used as the setpoint output, and a free DRIVE-CLiQ encoder interface is used as the actual-value input.

For more detailed information see the "adi4.ini" file, which is part of the toolbox and can be found in the "...\\Toolbox 802D_sl\\V01040000\\Special\\ADI4" directory.

For the analog spindle without encoder, MD30240 ENC_TYPE[n] must be set to zero.

20.9 Data lists

20.9.1 Machine data

| Number | Identifier | Name |
|------------------|------------------------------|--|
| Channel-specific | | |
| 20090 | SPIND_DEF_MASTER_SPIND | Master spindle |
| Axis-specific | | |
| 30134 | IS_UNIPOLAR_OUTPUT[0] | Setpoint output is unipolar |
| 30300 | IS_ROT_AX | Rotary axis |
| 30310 | ROT_IS_MODULO | Modulo conversion |
| 30320 | DISPLAY_IS_MODULO | Position display |
| 31050 * | DRIVE_AX_RATIO_DENOM[n] | Denominator load gearbox |
| 31060 * | DRIVE_AX_RATIO_NUMERA[n] | Numerator load gearbox |
| 32200 * | POSCTRL_GAIN [n] | Servo gain factor Kv |
| 32810 * | EQUIV_SPEEDCTRL_TIME [n] | Equivalent time constant speed control circuit for feedforward control |
| 34040 | REFP_VELO_SEARCH_MARKER | Reference point creep speed |
| 34060 | REFP_MAX_MARKER_DIST | Monitoring of zero mark distance |
| 34080 | REFP_MOVE_DIST | Reference point distance/destination point for distancecoded system |
| 34090 | REFP_MOVE_DIST_CORR | Reference point offset/absolute offset, distancecoded |
| 34100 | REFP_SET_POS | Reference point value |
| 34200 | ENC_REFP_MODE | Referencing mode |
| 35000 | SPIND_ASSIGN_TO_MACHAX | Assignment of spindle to machine axis |
| 35010 | GEAR_STEP_CHANGE_ENABLE | Gear stage change possible |
| 35040 | SPIND_ACTIVE_AFTER_RESET | Spindle active after reset |
| 35100 | SPIND_VELO_LIMIT | Maximum spindle speed |
| 35110 * | GEAR_STEP_MAX_VELO[n] | Maximum speed for gear change |
| 35120 * | GEAR_STEP_MIN_VELO[n] | Minimum speed for gear change |
| 35130 * | GEAR_STEP_MAX_VELO_LIMIT[n] | Maximum speed of gear stage |
| 35140 * | GEAR_STEP_MIN_VELO_LIMIT[n] | Minimum speed of gear stage |
| 35150 | SPIND_DES_VELO_TOL | Spindle speed tolerance |
| 35160 | SPIND_EXTERN_VELO_LIMIT | Spindle speed limitation via PLC |
| 35200 * | GEAR_STEP_SPEEDCTRL_ACCEL[n] | Acceleration in speed control mode |
| 35210 * | GEAR_STEP_POSCTRL_ACCEL[n] | Acceleration in position control mode |
| 35300 | SPIND_POSCTRL_VELO | Position control activation speed |
| 35310 | SPIND_POSIT_DELAY_TIME[n] | Positioning delay time |
| 35350 | SPIND_POSITIONING_DIR | Positioning direction of rotation for a nonsynchronized spindle |
| 35400 | SPIND_OSCILL_DES_VELO | Reciprocating speed |
| 35410 | SPIND_OSCILL_ACCEL | Oscillation acceleration |
| 35430 | SPIND_OSCILL_START_DIR | Starting direction during oscillation |

| Number | Identifier | Name |
|---------|-----------------------------|--|
| 35440 | SPIND_OSCILL_TIME_CW | Oscillation time for M3 direction |
| 35450 | SPIND_OSCILL_TIME_CCW | Oscillation time for M4 direction |
| 35500 | SPIND_ON_SPEED_AT_IPO_START | Feed enable with spindle in setpoint range |
| 35510 | SPIND_STOPPED_AT_IPO_START | Feed enable with stationary spindle |
| 36060 | STANDSTILL_VELO_TOL | Threshold velocity "Axis/spindle stationary" |
| 36200 * | AX_VELO_LIMIT [n] | Threshold value for velocity monitoring |
| 36300 | ENC_FREQ_LIMIT | Encoder limit frequency |
| 36302 | ENC_FREQ_LIMIT_LOW | Encoder limit frequency resynchronization |
| 36720 | DRIFT_VALUE | Drift basic value |

The machine data marked with * is contained in the parameter set for a gear stage.

20.9.2 Setting data

| Number | Identifier | Name |
|------------------|---------------------|--------------------------------------|
| General | | |
| 41200 | JOG_SPIND_SET_VELO | JOG velocity for the spindle |
| Spindle-specific | | |
| 43210 | SPIND_MIN_VELO_G25 | Programmable spindle speed limit G25 |
| 43220 | SPIND_MAX_VELO_G26 | Programmable spindle speed limit G26 |
| 43230 | SPIND_MAX_VELO_LIMS | Programmable spindle speed limit G96 |

20.9.3 Interface signals

| Number | Bit | Name |
|---------------|----------|--|
| Axis-specific | | |
| VD30x 0000 | - | M function for the spindle (DINT), axis-specific |
| VD30x 0004 | - | S function for the spindle (REAL), axis-specific |
| VB380x 0000 | - | Feed override |
| V380x 0001 | .7 | Override active |
| V380x 0001 | .5 | Position measuring system 1 |
| V380x 0001 | .3 | Axis/spindle disable |
| V380x 0002 | .2 | Spindle reset/delete distancetogo |
| V380x 0002 | .1 | Controller enable |
| V380x 0003 | .6 | Velocity/spindle speed limitation |
| V380x 2000 | .3 | Gear changed |
| V380x 2000 | .0 to .2 | Actual gear stage A to ...C |
| V380x 2001 | .4 | Resynchronize spindle during positioning 1 (spindle) |
| V380x 2001 | .6 | Invert M3/M4 |

| Number | Bit | Name |
|-------------|----------|--|
| V380x 2002 | .7 | Set direction of rotation counterclockwise |
| V380x 2002 | .6 | Set direction of rotation clockwise |
| V380x 2002 | .5 | Oscillation speed |
| V380x 2002 | .4 | Oscillation via PLC |
| VB380x 2003 | - | Spindle override |
| V390x 0000 | .7 | Position reached with exact stop fine |
| V390x 0000 | .6 | Position reached with exact stop coarse |
| V390x 0000 | .4 | Referenced/synchronized 1 |
| V390x 0000 | .2 | Encoder limit frequency exceeded 1 |
| V390x 0000 | .0 | Spindle / no axis |
| V390x 0001 | .7 | Current controller active |
| V390x 0001 | .6 | Speed control loop active |
| V390x 0001 | .5 | Position controller active |
| V390x 0001 | .4 | Axis/spindle stationary ($n < n_{min}$) |
| V390x 2000 | .3 | Change gear stage |
| V390x 2000 | .0 to .2 | Actual gear stage A to ...C |
| V390x 2001 | .7 | Actual direction of rotation clockwise |
| V390x 2001 | .5 | Spindle in setpoint range |
| V390x 2001 | .2 | Setpoint speed increased |
| V390x 2001 | .1 | Setpoint speed limited |
| V390x 2001 | .0 | Speed limit exceeded |
| V390x 2002 | .7 | Active spindle control mode |
| V390x 2002 | .6 | Active spindle mode oscillation mode |
| V390x 2002 | .5 | Active spindle positioning mode |
| V390x 2002 | .3 | Tapping with compensation chuck active |
| V390x 2002 | .0 | Constant cutting rate active (G96) |

Indexing Axes (T1)

21.1 Brief Description

Note

This function is not available with version T/M value.

Indexing axes in machine tools

In certain applications, the axis is only required to approach specific grid points (e.g. location numbers). It is necessary to approach the defined grid points, the indexing positions, both in AUTOMATIC and set-up mode.

The relevant axes are called "indexing axes". The positions defined on the indexing axes are known as "coded positions" or "indexing positions".

Applications

Indexing axes are used predominantly in connection with specific types of tool magazines such as tool turrets, tool chain magazines or tool cartridge magazines. The coded positions refer to the individual locations of the tools in the magazines. During a tool change, the magazine is positioned at the location containing the tool to be loaded.

Display indexing

The following data can be queried via system variables:

- The number of the current indexing position:
 - When the "exact stop fine" window of the indexing position is reached
 - When half the distance to the next indexing position is crossed
- The programmed indexing position

21.2 Indexing axes

21.2.1 General information

General rules

Indexing axes can be traversed manually in the setup mode types JOG and INC, from a part program with special instructions for "Coded positions" and by the PLC.

Upon reaching the indexing position the following interface signal is given out to the PLC:

V390x 1002.6 (indexing axis in position)

Hirth indexing axes cannot be traversed in JOG mode before reference point approach.

21.2.2 Traversing indexing axes in manual JOG mode

Reference point approach

An indexing axis approaches the reference point in the same way as other axes. The reference point does not have to coincide with an indexing position.

V390x 0000.4 (referenced/synchronized 1) = 1

When **the reference point is reached**, the indexing axis moves only to indexing positions in JOG mode during conventional and incremental traversing.

Exception: No indexing positions are approached when traversing with the handwheel.

V390x 0000.4 (referenced/synchronized 1) = 0

If the axis is not referenced, the indexing positions are not taken into account during manual traversal in JOG!

Continuous traversal in JOG

Pressing a "+" or "-" traversing key causes the indexing axis to move in the same way as with conventional JOG traversing. When the traversing key is released, the indexing axis traverses to the next indexing position in the direction of traversing.

If the operator changes the direction of traversing before the indexing position has been reached, the indexing axis is positioned on the next indexing position in the direction of traversing. The traversing movement must be started in the opposite direction.

For further information on "continuous traversing in jog mode", see:
Manual and Handwheel Travel (H1)

Incremental traversal in JOG mode (INC)

Irrespective of the current increment setting (INC1, ... ,INCvar), the indexing axis always traverses **through one indexing position** in the selected direction when a traversing key "+" or "-" is pressed .

In jog mode, the traversing movement is interrupted when the traversing key is released. The indexing position can be approached by pressing the traversing key again.

Between indexing positions

If an indexing axis is situated between two indexing positions, then it approaches the next-higher indexing position when the "+" traversing key is pressed in JOG-INC mode. Similarly, pressing the "-" traversing key causes the next-lower indexing position to be approached.

Handwheel traversal

When the indexing axis is traversed by means of the handwheel in JOG mode, the indexing positions **are ignored**. As the handwheel is turned, the indexing axis moves to any position depending on the basic system in mm, inches or degrees.

If traversing of the indexing axis with the handwheel is to be interlocked, this can be handled by the PLC user program.

Signal from PLC "Indexing axis in position"

When the indexing axis is traversed in JOG mode, the signal:V390x 1002.6 (indexing axis in position) is output at the PLC interface to indicate that an indexing position has been reached.

This is subject to the condition that the indexing axis has been referenced:
V390x 0000.4 (referenced/synchronized 1) = 1

Alarms in JOG mode

If the indexing axis leaves the traversing range defined in the indexing position table when traversing in JOG mode, alarm 20054 "Incorrect index for indexing axis in JOG" is output.

21.2.3 Traversing indexing axes in AUTOMATIC modes

Traversal to selected positions

An axis defined as an indexing axis can be made to approach **any selected position** from the NC part program in AUTOMATIC mode. This includes positions between the defined indexing positions. These positions are programmed, in the usual way, in the unit of measurement (mm/inches or degrees) for the axis. The general programming instructions used for this purpose (G90, G91, AC, IC, etc.) are described in the Programming Guide.

Traversal to "Coded positions"

Special instructions can also be used in the part program to traverse indexing axes in the following way:

| Instruction | Effect |
|-------------|--|
| CAC | Approach absolute coded position |
| CACP | Approach absolute coded position in positive direction |
| CACN | Approach absolute coded position in negative direction |
| CIC | Approach incremental coded position |
| CDC | Approach coded position along direct (shortest) path |

With absolute positioning, the indexing position to be approached is programmed, and with incremental positioning, the number of indexes to be traversed in the "+" or "-" direction is programmed.

On rotary axes, the indexing position can be approached directly across the shortest path (CDC) or with a defined direction of rotation (CACP, CACN).

V390x 1002.6

If the "Exact stop fine" window is reached and the indexing axis is positioned on an indexing position, IS "Indexing axis in position" (V390x 1002.6) is set, irrespective of how the indexing position was reached.

21.2.4 Traversing of indexing axes by PLC

Indexing axes can also be traversed from the PLC user program.

There are various methods:

Concurrent positioning axes: The indexing position to be approached can be specified by the PLC. See Chapter "Positioning Axes (P2)"

21.3 Parameterization of indexing axes

Definition of the indexing axis

An axis (linear or rotary axis) can be defined as an indexing axis with the aid of the axial machine data MD30500 INDEX_AX_ASSIGN_POS_TAB. The number of the indexing position table (1 or 2) must be entered in the machine data.

Several axes can be assigned to an indexing position table on condition that these indexing axes are of the same type (linear axis, rotary axis, modulo 360° function). Otherwise alarm 4080 "Incorrect configuration for indexing axis in MD [Name]" is output during ramp-up.

Indexing position tables

The axis positions (in mm or degrees) assigned to the indexes must be stored for each indexing axis in the form of a table in machine data. Any value can be entered for the distance between the individual indexing positions.

Note

Up to two indexing position tables are permitted:

MD10910 INDEX_AX_POS_TAB_1 [n]

MD10930 INDEX_AX_POS_TAB_2 [n]

The indexing positions are metric.

Number of entries for each table

Up to 60 positions can be entered in each indexing position table:

[n = 0 ... 59]

The actual number of entries used must be defined with the machine data for table 1 or 2:

MD10900 INDEX_AX_LENGTH_POS_TAB_1

MD10920 INDEX_AX_LENGTH_POS_TAB_2

All positions entered in the table which are higher than the number defined in the machine data are inactive.

Inch/metric switchover

The indexing positions refer to the configured system of units:

- MD10270 POS_TAB_SCALING_SYSTEM = 0: metric
- MD10270 POS_TAB_SCALING_SYSTEM = 1: inch

Note

MD10270 defines the scaling system of the position data for the following machine data:

MD10910 INDEX_AX_POS_TAB_1

MD10930 INDEX_AX_POS_TAB_2

MD10270 affects the following setting data:

SD41500 SW_CAM_MINUS_POS_TAB_1 to

SD41507 SW_CAM_PLUS_POS_TAB_4

MD10270 is not visible in expert mode!

Entry format

- The indexing positions must be entered in the table in ascending order (starting with the negative to the positive traversing range) with no gaps between the entries. Consecutive position values cannot be identical.
- The axis positions must be entered in the basic coordinate system.
- If the indexing axis is defined as a rotary axis with modulo 360° (MD30300 IS_ROT_AX=1 and MD30310 ROT_IS_MODULO=1), the following points must be considered for indexing positions:
 - Indexing positions are allowed in the range from $0^\circ \leq \text{Pos} < 360^\circ$. Positions outside this range generate alarm 4080 on power-up.
 - Since the indexing axis is defined as a continuously rotating rotary axis, indexing position 1 is approached after the highest valid indexing position in the table has been reached and the axis continues to traverse in the positive direction with INC. Similarly, indexing position 1 is followed by the highest valid indexing position in the negative direction with INC.

21.4 Programming of indexing axes**Coded position**

To allow indexing axes to be positioned from the NC part program, special instructions (so-called Coded positions) are provided with which the indexing numbers (e.g. location numbers) are programmed instead of axis positions in mm or degrees.

The following coded position instructions are possible, depending on whether the indexing axis is defined as a linear or rotary axis:

| | |
|-------------------------------|---|
| Indexing axis is linear axis: | CAC (i), CIC (i) |
| Indexing axis is rotary axis: | CAC (i), CIC (i), CACP (i), CACN (i), CDC (i) |

i = coded position; value range 0 ... 59, integer (positive and negative values are possible for CIC)

Absolute POS[B]=CAC(20)

Indexing axis B approaches coded position (index) 20 in absolute mode. The direction of traversing depends on the current actual position.

Absolute in positive direction POS[B]=CACP(10)

Indexing axis B approaches coded position (index) 10 in absolute mode in the positive direction of rotation (only possible with rotary axes).

Absolute in negative direction POS[B]=CACN(10)

Indexing axis B approaches coded position (index) 10 in absolute mode in the negative direction of rotation (only possible with rotary axes).

Direct absolute POS[B]=CDC(50)

Indexing axis B approaches indexing position 50 directly along the shortest path (only possible with rotary axes).

Incremental POS[B]=CIC(-4)

Indexing axis B traverses incrementally by four indexing positions in a negative direction from its current position.

POS[B]=CIC(35)

Indexing axis B traverses incrementally by 35 indexing positions in a positive direction from its current position.

The sign determines the approach direction.

Note

On modulo rotary axes, the indexing positions are divided in factors of 360° and approached directly.

Between indexing positions

If an indexing axis is situated between two indexing positions, the program commands have the following effect in AUTOMATIC mode:

| Program command | Effect |
|------------------|---|
| POS [B]=CIC (1) | The next higher indexing position is approached. |
| POS [B]=CIC (-1) | The next lower indexing position is approached. |
| POS [B]=CIC (0) | The indexing axis is not traversed. |

21.5 Starting up indexing axes

Procedure

The procedure for starting up indexing axes is identical to normal NC axes (linear and rotary axes).

Rotary axis

If the indexing axis is defined as a rotary axis (MD30300 IS_ROT_AX = 1) with modulo 360° conversion (MD30310 ROT_IS_MODULO = 1), indexing positions are also approached with modulo 360°. Only positions within the range from 0° to 359.999° can then be entered in the indexing position table. Otherwise alarm 4080 "Incorrect configuration for indexing axis in MD [Name]" is output during power-up.

The position display can be set to modulo 360° as follows:
 MD30320 DISPLAY_IS_MODULO = 1

Special machine data

The following machine data must be set in addition:

| General machine data | |
|-----------------------------------|---|
| MD10900 INDEX_AX_LENGTH_POS_TAB_1 | Number of positions for indexing axis table 1 |
| MD10920 INDEX_AX_LENGTH_POS_TAB_2 | Number of positions for indexing axis table 2 |
| MD10910 INDEX_AX_POS_TAB_1 [n] | Indexing position table 1 |
| MD10930 INDEX_AX_POS_TAB_2 [n] | Indexing position table 2 |

| Axial machine data | |
|---------------------------------|---|
| MD30500 INDEX_AX_ASSIGN_POS_TAB | Axis is indexing axis (assignment of indexing position table 1 or 2, or 3 for equidistant indexing) |

Machine data examples

The assignment of the above machine data is described in the following paragraphs using two examples.

Example of indexing axis as rotary axis

Tool turret with eight locations. The tool turret is defined as a continuously rotating rotary axis. The distances between the eight turret locations are constant. The first turret location is located at position 0°:

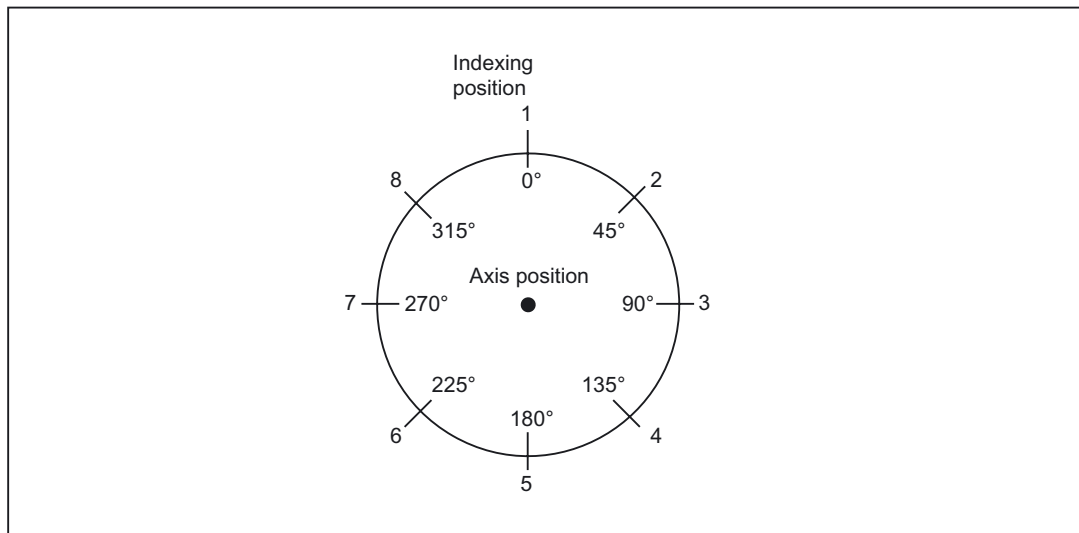


Figure 21-1 Example: Tool turret with eight locations

The indexing positions for the tool turret are entered in indexing position table 1.

| | |
|-------------------------------------|------------------------------|
| MD10910 INDEX_AX_POS_TAB_1[0] = 0 | 1. indexing position at 0° |
| MD10910 INDEX_AX_POS_TAB_1[1] = 45 | 2. indexing position at 45° |
| MD10910 INDEX_AX_POS_TAB_1[2] = 90 | 3. indexing position at 90° |
| MD10910 INDEX_AX_POS_TAB_1[3] = 135 | 4. indexing position at 135° |
| MD10910 INDEX_AX_POS_TAB_1[4] = 180 | 5. indexing position at 180° |
| MD10910 INDEX_AX_POS_TAB_1[5] = 225 | 6. indexing position at 225° |
| MD10910 INDEX_AX_POS_TAB_1[6] = 270 | 7. indexing position at 270° |
| MD10910 INDEX_AX_POS_TAB_1[7] = 315 | 8. indexing position at 315° |

Other machine data:

| | |
|---|---|
| MD10900 INDEX_AX_LENGTH_POS_TAB_1= 8 | Eight indexing positions in table 1 |
| MD30500 INDEX_AX_ASSIGN_POS_TAB [AX5] = 1 | Axis 5 is defined as indexing axis, indexing positions in table 1 |
| MD30300 IS_ROT_AX [AX5] = 1 | Axis 5 is rotary axis |
| MD30310 ROT_IS_MODULO [AX5] = 1 | Modulo conversion is activated |

Example of indexing axis as linear axis

Workholder with ten locations.

The distances between the ten locations are different. The first location is at position -100 mm.

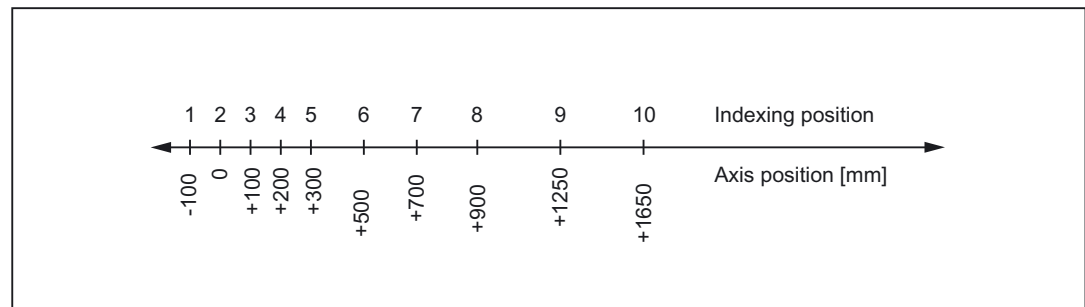


Figure 21-2 Example: Workholder as an indexing axis

The indexing positions for the workholder are entered in table 2:

| | |
|--------------------------------------|------------------------------|
| MD10930 INDEX_AX_POS_TAB_2[0] = -100 | 1. indexing position at -100 |
| MD10930 INDEX_AX_POS_TAB_2[1] = 0 | 2. indexing position at 0 |
| MD10930 INDEX_AX_POS_TAB_2[2] = 100 | 3. indexing position at 100 |
| MD10930 INDEX_AX_POS_TAB_2[3] = 200 | 4. indexing position at 200 |
| MD10930 INDEX_AX_POS_TAB_2[4] = 300 | 5. indexing position at 300 |
| MD10930 INDEX_AX_POS_TAB_2[5] = 500 | 6. indexing position at 500 |
| MD10930 INDEX_AX_POS_TAB_2[6] = 700 | 7. indexing position at 700 |
| MD10930 INDEX_AX_POS_TAB_2[7] = 900 | 8. indexing position at 900 |

| | |
|--------------------------------------|-------------------------------|
| MD10930 INDEX_AX_POS_TAB_2[8] = 1250 | 9. indexing position at 1250 |
| MD10930 INDEX_AX_POS_TAB_2[9] = 1650 | 10. indexing position at 1650 |

Other machine data

| | |
|---|---|
| MD10920 INDEX_AX_LENGTH_POS_TAB_2 = 10 | Ten indexing positions in table 2 |
| MD30500 INDEX_AX_ASSIGN_POS_TAB [AX6] = 2 | Axis 6 is defined as indexing axis, indexing positions in table 2 |

21.6 Special features of indexing axes

Software limit switch

The software limit switches are also effective during traversing movements once the indexing axis has been referenced.

In handwheel traversing with JOG continuous or JOG incremental, the indexing axis stops at the indexing position ahead of the software limit switch.

Axis positions

Since the axis positions stored in the indexing position tables only correspond to the machine positions when the axis is referenced, an NC start must be disabled for as long as the indexing axis is not referenced.

Position display

Positions on indexing axes are displayed in the units of measurement normally used for the axes (mm, inches or degrees).

Abort through RESET

RESET causes the traversing movement on an indexing axis to be aborted and the axis to be stopped. The indexing axis is no longer positioned on an indexing position.

21.7 Data lists

21.7.1 Machine data

| Number | Identifier | Name |
|-----------------------|---------------------------|---|
| General | | |
| 10900 | INDEX_AX_LENGTH_POS_TAB_1 | Number of positions for indexing axis table 1 |
| 10910 | INDEX_AX_POS_TAB_1[n] | Indexing position table 1 |
| 10920 | INDEX_AX_LENGTH_POS_TAB_2 | Number of positions for indexing axis table 2 |
| 10930 | INDEX_AX_POS_TAB_2[n] | Indexing position table 2 |
| Axis/spindle-specific | | |
| 30300 | IS_ROT_AX | Rotary axis |
| 30310 | ROT_IS_MODULO | Modulo conversion for rotary axis |
| 30320 | DISPLAY_IS_MODULO | Position display is modulo 360° |
| 30500 | INDEX_AX_ASSIGN_POS_TAB | Axis is indexing axis |

21.7.2 Interface signals

| Number | Bit | Name |
|---------------|-----|---------------------------|
| Axis-specific | | |
| V390x 0000 | .4 | Referenced/synchronized 1 |
| V390x 1002 | .6 | Indexing axis in position |

Tangential Control (T3)

Note

This function is only available for the T/M pro and G/N pro versions.

22.1 Brief description

Function

The "tangential control" function belongs to the category of NC functions with coupled axes. The function has the following characteristics:

- There are two leading axes (guide axes) which are moved using standard traversing instructions and therefore traverse a path. A following axis also exists and its movement is dependent on this path contour (on the direction of the tangent in the current point on the path).
- The leading axes are defined as geometry axes and the following axis as a rotary axis.
- The coupled axes are axes in the same NC channel.
- The position of the following axis can be the input value for a transformation.
- Tangential control is only active in AUTOMATIC or MDA modes.

Application examples

- Positioning and tangential tracking of a rotatable tool during nibbling.
- Tracking of the workpiece alignment for a belt saw.
- Positioning and tangential tracking of a dressing tool on a grinding wheel.
- Tangential tracking a small cutting wheel when machining glass or paper.
- Tangential feed of a wire for 5-axis welding.

Defining, activating, deactivating the axis coupling

A program instruction defines the tangential axis coupling. Additional activation and deactivation instructions exist in the program.

Deleting an axis coupling

The definition of a tangential axis coupling may be deleted by means of a program instruction. An axis coupling with this following axis can then be defined again.

22.2 Characteristics of "Tangential control" function

Problem definition

Follow-up control for the rotary axis must be implemented so that the axis is always positioned at a specified angle to the tangent on a programmed path of two leading axes.

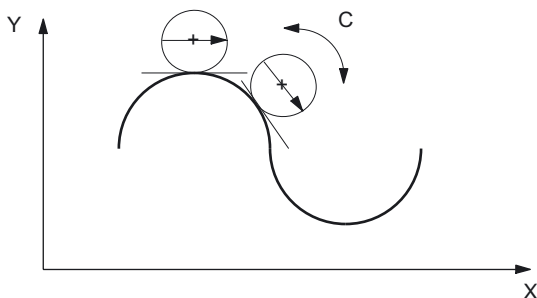


Figure 22-1 Example of a tangential control with an angle of zero degrees to the path tangent

In the figure, X and Y are the leading axes inside which the path is programmed. C is the following axis, the position of which is determined by the control depending on the leading axis path and the desired angle between the tangent and alignment in C.

The tangential control will function only if the leading axes are used as path axes. A leading axis which is programmed as a positioning axis (POS or POSA) does not specify values required for the follow-up control function.

Response to corners in the path

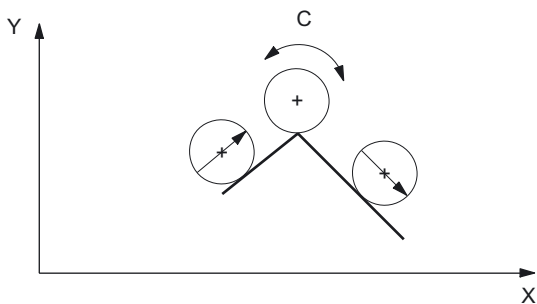


Figure 22-2 Corner in the path contour

Corner in the path contour

If the path specified by the leading axes has a corner, jumps in the direction of the path tangent occur at this point. In this situation, the following responses can be selected for the following axis:

- Without programming TLIFT (see Section: Use of "Tangential control" function): The path velocity is reduced to such an extent that the following axis reaches its new alignment synchronously with the target position of the leading axes.
- If TLIFT has been programmed, an intermediate block is inserted at any corner whose angle is greater than MD37400 EPS_TLIFT_TANG_STEP (tangential angle for corner recognition). In this inserted intermediate block, the rotary axis is moved as fast as possible to the position corresponding to the tangent after the corner. The limit values set for this axis are not violated. At the beginning of the intermediate block, the path velocity of the leading axes is zero.

Information

In blocks with G641 rounding, **no** intermediate block is generated for the following axis with TLIFT. An intermediate block is not required in this case. The compensation of the following axis is performed in conjunction with the rounding operation.

Hidden corner in space

A corner relevant for the tangential follow-up control can be hidden in space. The projection of the contour on the plane defined by the two leading axes is relevant. If there is a hidden corner in space, an intermediate block is inserted before the block causing the tangential jump. This intermediate block moves the following axis to the new position. The block transition is not rounded.

22.3 Use of "Tangential control" function

22.3.1 Overview

Definition, activation

The "Tangential control" function requires the following sequence in the program:

- Assignment of leading axes and definition of following axis with TANG().
- The response at corners is specified, if required program TLIFT().
- Activate the defined coupling with TANGON().

Change response at corners, deactivate, delete definition

If required:

- Change the response at corners (TANG without following TLIFT).
- Deactivation of coupling with TANGOF().
- Deletion of definition with TANGDEL().

Effect on transformations

The position of the rotary axis to which follow-up control is applied can act as the input value for a transformation.

See Kinematic Transformations (M1), Section "TRANSMIT"

Note

The user is recommended to program TLIFT if tangential control is used together with a transformation. TLIFT prevents the follow-up axis from overtraveling and protects against excessive compensating movements.

Explicit programming of following axis in an active coupling

If the following axis is programmed directly, then the positional data is added to the offset angle programmed in the activation instruction TANGON. In this case, positional data with AC, IC, DC, POS for the following axis are permitted.

Reference point approach

Axis coupling is deactivated during reference point approach.

Detailed programming instructions

Reference:

/BPN/ Operation and Programming, Nibbling

22.3.2 Definition of axis coupling: TANG

Programming

The programming is carried out using the predefined subroutine **TANG()**. The following parameters are transferred:

| Parameter | Value |
|-----------------------------------|------------|
| Following axis (rotary axis) | Example: C |
| Leading axis 1 (geometrical axis) | Example: X |
| Leading axis 2 (geometrical axis) | Example: Y |

| Parameter | Value |
|-------------------------------------|---|
| Coupling factor | Default 1 |
| Identifier of the coordinate system | "B"->Basic coordinate system, optional |
| Optimization | "S" = Standard (default) or "P" = Automatic adaptation of the time characteristic for following and leading axis |

The appropriate axis identifiers are used to specify the axes. The coupling factor is normally 1. The coupling factor of 1 can be omitted, the same applies for "B".

Example of abbreviated call: **TANG(C, X, Y)**
 Examples of possible calls: **TANG(C, X, Y,1)**
TANG(C, X, Y,1,"B")
 Example of complete call: **TANG(C, X, Y, 1, "B","S")**

22.3.3 Switching on the axis coupling: TANGON

Programming

The axis coupling is switched on using the predefined subroutine **TANGON()**. The following axis to be tracked is specified. This specification refers to the definition of leading and following axes made beforehand with **TANG**.

An offset angle between the tangent and the position of the following axis can be specified optionally when follow-up is activated. This angle is maintained by the control for as long as the following axis is made to follow. The angle is added to the angle stored in machine data MD37402 **TANG_OFFSET**. If the angle is zero both in **TANGON** and in the MD, the following axis takes the direction of the tangent.

Example without offset angle: **TANGON(C)**

Example with offset angle at 90 degrees: **TANGON(C, 90)**.

22.3.4 Response at corners, activation "with intermediate block": TLIFT

Programming

Following the definition with **TANG()**, the **TLIFT** instruction with the following axis must be written if the corner response "with intermediate block" is required.

TLIFT (C)

The control considers the associated machine data MD37400 **EPS_TLIFT_TANG_STEP** for the tangential following rotary axis C. If the tangential angle jump exceeds the angle (absolute value) of the angle set in the MD, the control recognizes a "corner" and approaches the new position of the follow-up axis via an inserted intermediate block.

22.3.5 Switching off the axis coupling: TANGOF

Programming

The axis coupling is switched off using the predefined subroutine **TANGOF()**. The following axis is hereby specified. This specification refers to the definition of leading and following axes made beforehand with TANG. With reference to the above example, the termination is:

TANGOF(C)

The follow-up control process initiated with TANGON is terminated. Termination of follow-up control initiates a preprocessing stop (preprocessor stop).

Information

The master (leading) axes and following axis, defined using TANG(...) are not cleared by TANGOF.

After RESET/part program end, no tangential control is active (default setting).

22.3.6 Deactivation of corner response "with intermediate block"

Programming

In order to stop generating the intermediate block at corners with active tangential follow-up control, the definition block TANG(...) must be repeated without a following TLIFT().

22.3.7 Deleting the definition of axis coupling: TANGDEL

Programming

An axis coupling definition specified by TANG(...) remains active after TANGOF. This inhibits a plane change or geometry axis switchover. The predefined subroutine **TANGDEL()** may be used to delete the definition of the axis coupling. Other axis couplings with this following axis can then be defined.

Example:

TANGDEL(C): The previous definition of TANG(C,X,Y) is cancelled.

22.3.8 Programming examples

Example with insertion of an intermediate block for positioning of rotary axis

```

N10 TANG (C, X, Y, 1)      ; Definition of the tangential control
N20 TLIFT (C)             ; Activate Insert intermediate block at corners
N30 G1 G641 X0 Y0 F1000
N40 TANGON (C)           ; Activate tangential control
N50 X10
N60 Y10                   ; Before this block is processed, rotary axis C is
                           ; repositioned
                           ; in an intermediate block.
N70 M2

```

Example with plane change for the leading axes

```

N10 TANG(A, X, Y, 1)     ; Definition, following axis is rotary axis A
N20 TANGON(A)           ; Switch on
N30 X10 Y20
.....
N80 TANGOF(A)          ; Switch off
N90 TANGDEL(A)         ; Delete defined coupling of A to X and Y as leading axes
.....
N120 TANG(A, X, Z)      ; New definition, A can be coupled to new leading axes
N130 TANGON(A)         ; Activate new coupling
.....
N200 M2

```

22.4 Limit angle for reversal of path direction

Problem

When the axis moves backwards and forwards along the path, the tangent direction turns abruptly through 180 degrees at the path reversal point. This response is not generally desirable for machining operations (e.g. grinding of a contour). It is far better for the reverse motion to be executed using the same absolute offset angle as the forward motion.

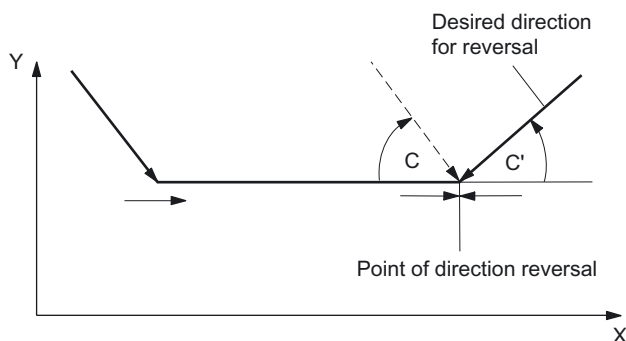


Figure 22-3 Path movement with direction reversal and alignment of follow-on axis

Solution and programming

A minimum and a maximum value for the position of the following axis ("C" in example), in relation to the base coordinate system, are transferred to the control with G25 and G26 (working area limit). These working area limitations are activated with WALIMON and deactivated again with WALIMOF. The working area limitation must be active at the instant of path reversal.

Reference:

/BPN/ Operation and Programming, Nibbling

Effect

If the current offset angle is outside the active working area limit for the following axis, an attempt is made to return to within the permissible working area by means of the negative offset angle.

Programming example for direction reversal, principle

```

N10 TANG (C, X, Y)      ; Definition of the tangential control
N20 TLIFT (C)          ; Activate Insert intermediate block at corners
N30 G1 X10 Y0 F1000
N40 G25 C50            ; Value of the lower working area limit
N50 G26 C70            ; Value of the upper working area limit
N60 TANGON (C, 60)     ; Switch on tangential control, offset angle 60 degrees
N70 WALIMON            ; Switch on working area limitation
N80 X100
N90 X10                ; Before this block is processed, rotary axis C is
                       ; repositioned in
                       ; an intermediate block, C' = -60 degrees.
N200 M2

```

22.5 Data lists

Machine data

| Number | Identifier | Name |
|---------------|---------------------|--|
| Axis-specific | | |
| 37400 | EPS_TLIFT_TANG_STEP | Tangential angle for corner recognition |
| 37402 | TANG_OFFSET | Default angle for tangential follow-up control |

Speed/torque coupling, master-slave (TE3)

23.1 Brief description

Note

This function is under license and is not available with version T/M value.

If the corresponding option is activated without a valid license, alarm 8081 "%1 option(s) that has (have) not been licensed using a license key was (were) set" is output. It will not be possible to operate the machine as normal.

For information on operations relating to "Setting (an) option(s)", please refer to the section titled "Licensing in SINUMERIK 802D sl" in the "Turning, Milling, Grinding, Nibbling" Operating Instructions.

Function

The speed/torque coupling function (master-slave) is used for mechanically-coupled axes that are driven by two separate motors.

Additional applications:

- Compensation of gear backlash and backlash in the gear tooth flank due to mutual tension in the drives
- Coupling/decoupling of **rotating, speed-controlled spindles**
- Dynamic configuration of couplings

Speed/torque coupling (master-slave) is a speed setpoint coupling between a master and a slave axis, involving a torque compensatory controller for even torque distribution.

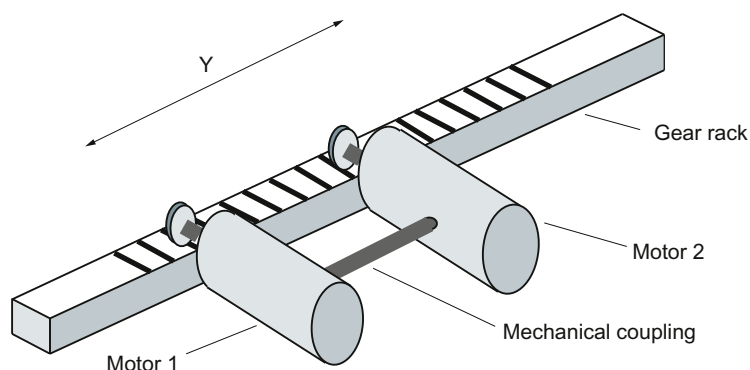


Figure 23-1 Permanent mechanical coupling

23.1 Brief description

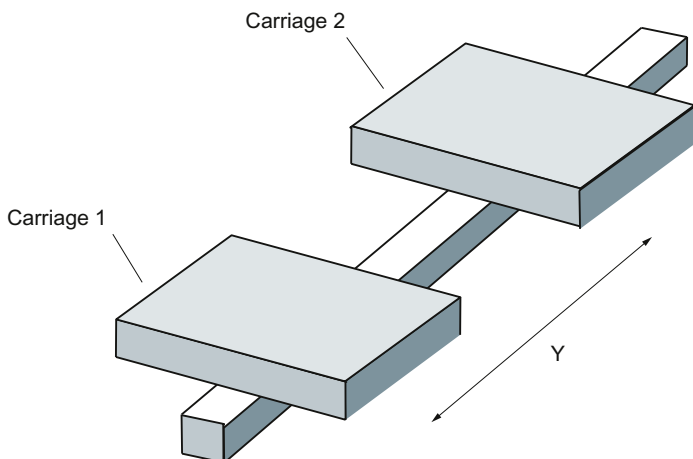


Figure 23-2 Carriages (linear motor) for temporary coupling

| |
|---|
| NOTICE |
| Each slave axis has exactly one master axis and vice versa. |

Characteristics

- If traversing is programmed for a slave axis that has already been linked, an alarm is issued.
- The setpoint position of the coupled slave axis corresponds to the current actual position.
- On request, the coupling is made or released independent of the channel status the next time the axis stops. This allows the coupling status to be changed even during part program processing.
- For brake control, the interface signal "Master-slave coupling status active" should be used.
- If a master axis is simultaneously configured as the slave, an alarm is issued. So cascading is not possible.
- If a coupling is requested and closed, the control activation signals are derived directly from the master axis.
- If the coupling is closed, the slave axis is speed-controlled, meaning that status signal "Position controller active" (V390x 0001.5) is not set.
- A separate item of machine data has been provided for reversing the direction of the slave axis in the coupled state:
MD37274 MS_MOTION_DIR_REVERSE (reverse master-slave traversing direction)

23.2 Coupling diagram

If the coupling is closed, the slave axis is traversed only with the load-side setpoint speed of the master axis. It is therefore only speed-controlled, not position-controlled.

No positional deviation control is implemented between master and slave axes. A torque compensatory controller divides the torque evenly over the master and slave axes.

An additional torque can be used to achieve a tension between the master and slave axis.

If different motors are used, individual weighting factors can be used to adapt the torque distribution.

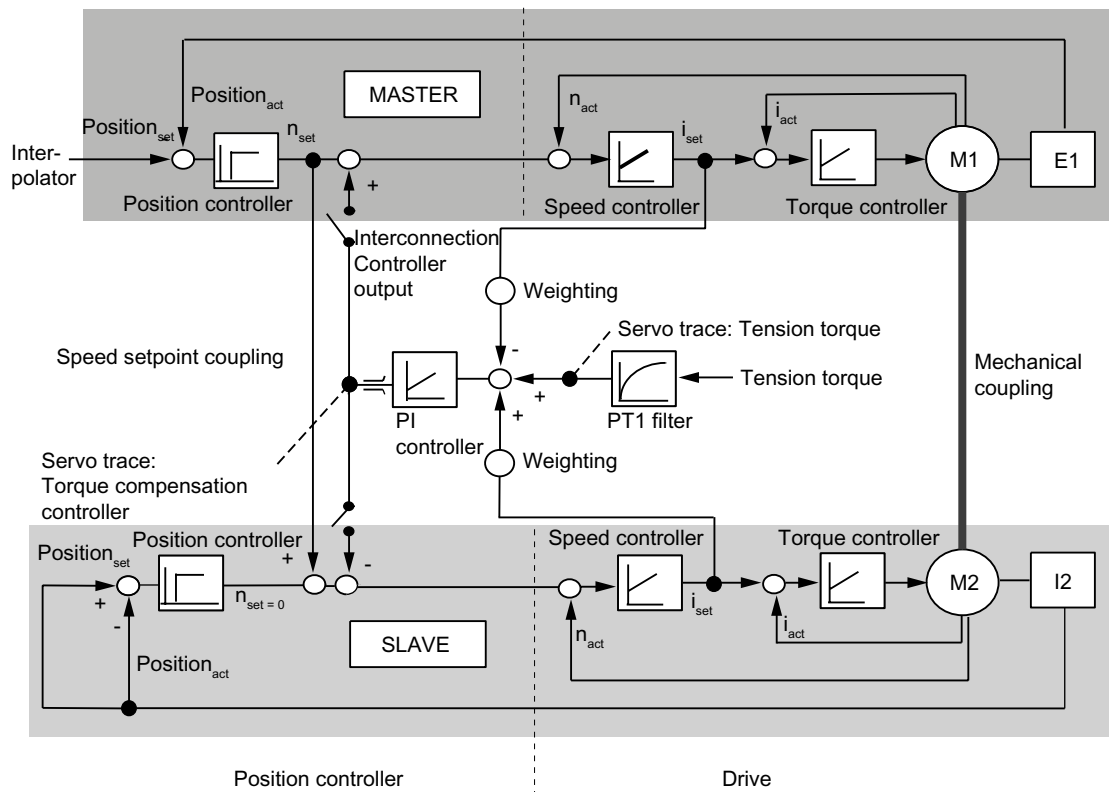


Figure 23-3 Control structure

23.3 Configuring a coupling

Static

A master-slave coupling is configured in the slave axis only.

Each slave axis is assigned one master axis for speed setpoint coupling and one for torque compensation control (default setting).

The assignments done in the following machine data are automatically active in each control start-up.

MD37252 MS_ASSIGN_MASTER_TORQUE_CTR

MD37250 MS_ASSIGN_MASTER_SPEED_CMD

23.4 Torque compensatory controller

A PI controller calculates a load-side additional speed setpoint from the torque difference between the master and slave axes. This is applied as standard to the command speed setpoint in the master and slave axes with different signs in each case.

The torque setpoints used for torque compensation control are smoothed in the drive (r0079). The smoothing time constant is entered in drive parameter p0045. The same value should be set in the master and slave axes.

The gain factor MD37256 MS_TORQUE_CTRL_P_GAIN corresponds to the percentage ratio of the maximum axis velocity MD32000 MAX_AX_VELO to the rated driving torque (p2003) of the slave axis.

The I component is disabled in the default setting.

The integral time MD37258 MS_TORQUE_CTRL_I_TIME is entered in seconds.

The torque compensation controller output is actively limited to the MD37260 MS_MAX_CTRL_VELO value.

If the following settings are made, the torque compensation controller will be inactive:

- MD37254 MS_TORQUE_CTRL_MODE = 3
- MD37256 MS_TORQUE_CTRL_P_GAIN = 0.

The torque distribution can be parameterized via the input variables of the torque compensatory controller. Weighting of the driving torque values:

- Slave axis: MD37268 MS_TORQUE_WEIGHT_SLAVE
- Master axis: 100 - MD37268

If motors with different rated torque values are used, the 50% to 50% standard distribution must be adapted to suit.

A mechanical coupling is absolutely necessary when the torque compensatory controller is used. Otherwise, the drives involved could accelerate from standstill.

Activation/deactivation via the PLC

The torque compensation controller can be activated and deactivated directly via PLC interface signal V380x 5000.4.

For this purpose, the following machine data must be set:
MD37255 MS_TORQUE_CTRL_ACTIVATION = 1

The activated status can be read back via V390x 5000.4.

MD37254 MS_TORQUE_CTRL_MODE is used for configuring the torque distribution.

23.5 Tension torque

By specifying a additional torque MD37264 MS_TENSION_TORQUE, you can achieve tension between the master axis and slave axis when the torque compensation controller is active. The tension torque is entered as a percentage of the rated torque and is active straight away.

The tension torque tension torque is applied via a PT1 filter. Specifying a filter time constant MD37266 MS_TENSION_TORQ_FILTER_TIME > 0 activates the filter.

The tension torque chosen must be high enough to ensure that the resulting torque does not drop below the minimum required tension even during acceleration. To prevent unnecessary heating in the motor, you can reduce the tension torque when the motor is at standstill.

Specifying a tension torque without a mechanical coupling produces axis movement.

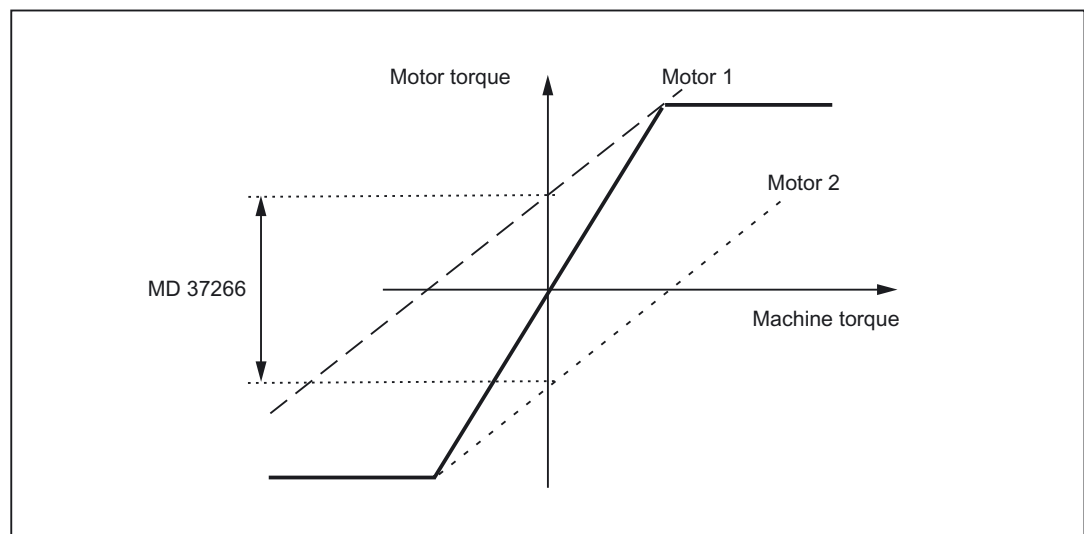


Figure 23-4 Resulting tension torque

23.6 Activating a coupling

The type of activation for a master-slave coupling is defined in the following machine data:

MD37262 MS_COUPLING_ALWAYS_ACTIVE

Depending on the machine configuration, a distinction is made between a permanent and a temporary master-slave coupling.

Only a temporary master-slave coupling (MD37262 = 0) that has been configured can be closed and disconnected using the axial PLC interface signal "Master/slave on" (V380x 5000.7) or within a part program using the following commands:

- MASLON(slave axis1)
- MASLOF(slave axis1)

The setpoint status of the coupling always corresponds to the last specification made.

The current coupling status can be read out in the slave axis via PLC interface signal "Master/slave coupling active" (V390x 5000.7).

In the part program and from the synchronized actions, the current coupling status can be output via the **system variable** of the slave axis \$AA_MASL_STAT.

Spindle coupling

The MASLOFS(Slv1) instruction can be used to disconnect the coupling in the same way as MASLOF and decelerate the slave spindle automatically.

Note

A permanent coupling (MD37262 = 1) does not require explicit activation.

Control system response

The control response to POWER ON, operating mode changes, RESET, block searches, and REPOS is as follows:

- A master-slave coupling activated via the PLC or MASLON instruction is retained after an operating mode change, RESET, or the end of a part program.
- MASLON / MASLOF / MASLOFS becomes effective on block search. Changes in the position of a coupled slave axis and the spindle speed must be calculated separately by the user.

23.7 Response on activation/deactivation

Activating/deactivating during axis standstill

Activation/deactivation is not active until the axis next comes to a standstill. If the specification is changed, the sequence is the same as for axis replacement. The coupling is closed when the axis comes to a standstill. The coupled axes must be in feedback control mode.

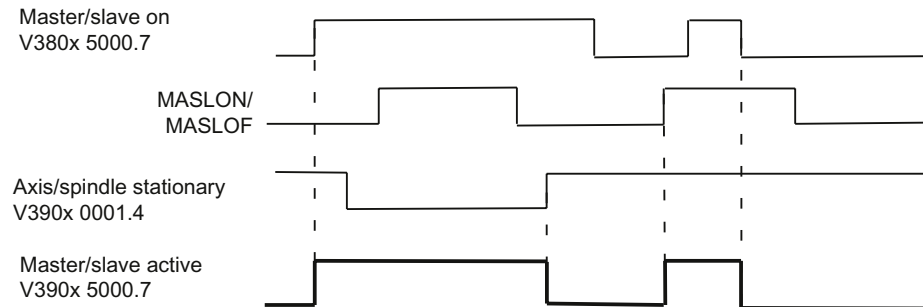


Figure 23-5 Activation procedure

Block stepping is halted for `MASLON` until the coupling has actually been closed. During this time, the associated channel operating message "Master-slave switchover active" is displayed on the user interface.

Activation/deactivation during motion

Activation/deactivation of the coupling in motion has been implemented only for spindles in speed control mode. For axes and spindles in positioning mode, switchover is still carried out when the axis is at a standstill.

Activation during motion

The coupling procedure at different speeds is divided into two phases.

Phase 1

Closure of the coupling is requested with interface signal "Master/slave on" (V380x 5000.7). The slave spindle accelerates or decelerates along the ramp with the dynamic response available to it until it reaches the setpoint speed of the master spindle.

When the setpoint speed is reached, the coupling is closed and IS "Coupling active" (V390x 5000.7) is set.

If the master spindle is accelerated during the coupling process, the first phase is extended according to the existing difference in dynamics between the master and slave spindles.

Phase 2

In the second phase, the differential speed between the master and slave spindle is used to generate the following synchronism signals.

IS "Velocity tolerance coarse" (V390x 5000.3)

IS "Velocity tolerance fine" (V390x 5000.2)

The associated limits are defined via the following machine data:

MD37270 MS_VELO_TOL_COARSE (velocity tolerance coarse)

MD37272 MS_VELO_TOL_FINE (velocity tolerance fine).

Note

The "Velocity tolerance coarse" signal can be used to implement a PLC monitoring function that checks a coupled grouping for loss of speed synchronism. The "Velocity tolerance fine" signal can be used to derive the time taken to close the coupling mechanically and activate the torque compensation controller directly.

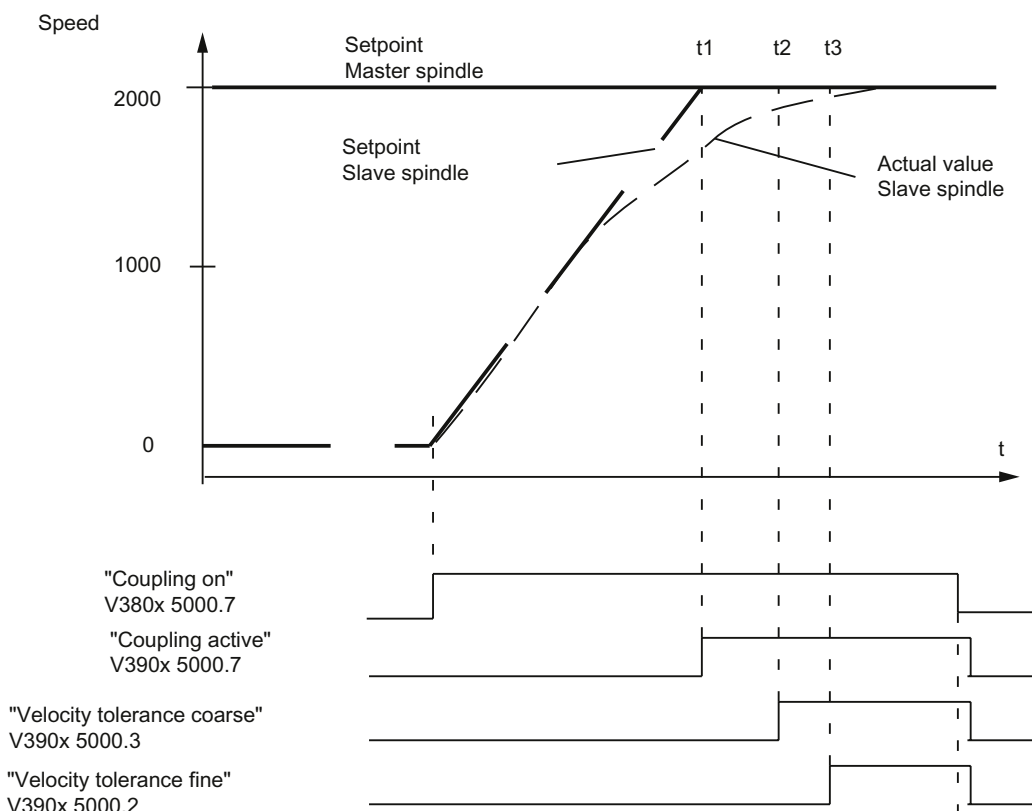


Figure 23-6 Coupling procedure between two spindles with different speeds

Deactivation during motion

An active coupling is disconnected using the MASLOF instruction.

This instruction is executed directly for a spindle in speed control mode. The slave spindle that is rotating at this point in time retains its most recently used speed until the speed is reprogrammed.

You can use the MASLOFS instruction to decelerate slave spindles automatically when disconnecting the coupling. For axes and spindles in positioning mode, the coupling is still only disconnected at standstill.

Note

The implicit preprocessor stop is omitted for MASLON and MASLOF. The missing preprocessor stop means that the \$P system variables of the slave spindle do not supply updated values until reprogrammed.

Coupling characteristics

For spindles in speed control mode, the coupling characteristics of the MASLON, MASLOF, MASLOFS, and MASLDEL instructions, as well as the PLC with IS "Master/slave on" (V380x 5000.7) are defined explicitly via the following machine data:

MD37263 MS_SPIND_COUPLING_MODE

MD37263 = 0

Coupling and disconnection take place only at standstill.

The current coupling state is retained until all axes involved have actually come to a standstill. The MASLOFS and MASLOF instructions are identical; the slave spindle is not decelerated automatically.

MD37263 = 1

Coupling and disconnection takes place immediately and therefore during motion.

During coupling, the slave spindles are automatically accelerated to the current speed of the master spindle.

On disconnection, the slave spindles rotating at this point retain their speeds until the speed is reprogrammed. However, a slave spindle disconnected with MASLOFS decelerates automatically.

23.8 Axial interface signals

When a master/slave coupling is requested, the PLC axis enables "Servo enable" (V380x 0002.1) and "Pulse enable" (V380x 4001.7) of the slave axis are derived directly from the specifications of the master axis. The separate PLC axis enable signals have no effect.

Removing the servo enable in the master axis also results in an interpolated deceleration of the slave axis that is carried out within the time configured for MD36610. The associated speed and current servo enables for individual axes will only be removed once the time for MD36620 has elapsed.

To continue to ensure the same braking response is carried out, where possible the time set in the following machine data should be the same for all coupled axes:

MD36620 SERVO_DISABLE_DELAY_TIME

The same applies to the following drive parameters:

- p9560 (pulse suppression, shutdown speed)
- p1228 (pulse suppression, delay time)

If the "Current controller active" (V390x 0001.7) or "Speed controller active" (V390x 0001.6) drive status signals are missing in the master or slave axis, the PLC interface signal "Master/slave active" (V390x 5000.7) is reset in the slave axis while it is at a standstill. When the master and slave axes return to closed-loop control mode, IS "Master/slave active" is set on the slave axis.

The PLC activates the torque compensation controller using the following IS: V380x 5000.4. The status of the torque compensation controller can be read in IS "Master/slave comp. contr. active" (V390x 5000.4).

Note

If the coupling is closed, the slave axis is speed-controlled, meaning that status signal "Position controller active" (V390x 0001.5) is not set.

23.9 Axial monitoring functions

Except monitoring the speed setpoint and actual speed, axial monitoring functions like contour and standstill in the slave axis are inactive because of the missing position controller. The position control circuit parameters like gain factor, precontrol, balancing can thus be set differently for master and slave axes without initiating the monitoring functions.

To achieve the same braking response for all coupled axes in the event of a fault, the same alarm reaction is applied to the entire coupling grouping when the coupling is active.

When correcting fault states, repositioning of slave axes on the interrupt point is suppressed.

23.10 Response in conjunction with other functions

Function Generator

To calibrate the speed control circuit for a closed master-slave coupling, a low value should be set in slave axis MD37268 MS_TORQUE_WEIGHT_SLAVE. Traversing of a coupled-motion slave axis is not prevented by the torque compensatory controller in this case.

Reference point approach

If the coupling is closed, only the master axis can be referenced. Referencing of slaves axes is suppressed. The referencing requirement does not have to be explicitly canceled for the slave axis in order to do this. The referencing status of coupled slave axes remains unchanged. The slave axis position is generally not the same as the master axis position. This difference in position is not significant. If the coupling is not closed, each axis can be referenced separately as usual.

Compensation

Position offsets of the slave axis, such as spindle pitch errors, backlash, temperature and sag offsets are computed but not active because there is no position controller.

Correct calculation of the backlash compensation requires that the backlash of the slave axis is always overtraveled by the motion of the master axis in coupled mode. Disconnecting the coupling during an axis reversal error will generate an incorrect actual value for the slave axis.

Dynamic stiffness control

The Kv factor of the master axis is copied to the slave axis for an existing coupling and is thus also active in the slave drive. This is an attempt to achieve the same control response in the drive of the master and slave axis as far as possible. MD32640 must be configured identically in all coupled axes.

Speed precontrol

The feedforward control in the slave axis does not have to be activated explicitly. The current settings of the master axis apply. The speed feedforward value of the master axis is already incorporated in the speed setpoint of the slave axis.

The mechanical situation changes in coupled mode. Axial settings must be adjusted accordingly. All coupled drives should have the same speed control dynamics.

Gantry

If one master-slave relationship is defined on each side of the gantry grouping to increase the gain, only the leading axis or following axis may be operated as a master axis.

Moving to fixed end stop

The travel to fixed stop function can be programmed only in the master axis when a coupling is active and has a different effect on the master and slave axes.

- The programmed value is expressed as a percentage of the reference driving torque of the master axis. The master axis detects when the fixed stop has been reached.
- The programmed value is also active on the slave axis, but refers to the reference driving torque of the slave axis.

If the reference torque values of the master and slave axes are different, they can be brought into line using the following slave axis machine data:

MD37014 FIXED_STOP_TORQUE_FACTOR

Specifying a factor < 1 reduces the programmed clamping torque in the slave axis.

Please note the following constraints:

- Torque distribution between the master and slave axes is not possible during clamping as the torque compensatory controller is deactivated during clamping operations.
- Status changes to the master-slave coupling have no effect during travel to fixed stop. Specification of a new status is only accepted when the fixed stop function has been completed.

Weight counterbalance

The additional torque for the electronic counterweight (MD32460) is computed in the slave axis, irrespective of the coupling status.

Gear stage change with active master-slave coupling

An automatic gear stage change in a coupled slave spindle is not possible and can only be implemented indirectly using the master spindle. The point in time at which the gear stage is changed is then derived from the master spindle. The oscillating motion of the coupled slave spindle is generated implicitly via the oscillating motion of the master spindle.

In contrast to the master spindle, the associated parameter block must be explicitly selected in the coupled slave spindle. To enable the parameter block to be specified, the following machine data must be set to the value 2:

MD35590 PARAMSET_CHANGE_ENABLE (parameter set change possible)

In the event of a gear stage change for the master/slave spindle, the associated parameter set index can be activated by the PLC via the VDI interface.

Hardware and software limit switches

Crossing of hardware and software limit switches is detected in coupled axes; the software limit switch is generally overtraveled on slave axes. The alarm is output on the slave axis, while braking is initiated via the master axis.

The path traveled after the slave software limit switch responds is equal to the distance required by the master axis to brake the coupling.

The master axis controls the movement away from the limit switch, since the coupling cannot be disconnected until the cause of the alarm has been eliminated.

23.11 Supplementary conditions

Note

The master-slave function requires the master and slave axes to be operated on the same 802D sl control system. They must not be PLC axes.

- **Closing and separating** the master-slave coupling is carried out when the axis has stopped.
- **Traversing a slave axis** with the coupling closed is possible only via the master axis.
- When the coupling is closed using the slave axis, the master axis is **braked automatically**. This produces an asymmetric response on closure and separation of the coupling. In contrast to closing, there is no automatic braking on separation.
- If a traversing movement is programmed for a slave axis that has already been coupled, the alarm "14092 Channel %1 Block %2 Axis %3 has the wrong type" appears.
- The setpoint position of the coupled slave axis corresponds to the current actual position.
- On request, the coupling is made or released independent of the channel status the next time the axis stops. This allows the coupling status to be changed even during part program processing.
- PLC interface signal V390x 0001.5 "Position controller active" must no longer be evaluated in the braking control logic of the slave axis. This is no longer set for an active coupling. Instead, interface signal V390x 5000.7 "Master-slave coupling status active" should be used.
- If a master axis is simultaneously configured as a slave axis, the alarm "26031 Axis %1 Configuration error master-slave" appears. So cascading is not possible.
- If a coupling is requested and closed, the control activation signals are derived directly from the master axis.
- The master-slave coupling must be deactivated prior to a gear change or a star-delta (wye-delta) switchover.

Modulo rotary axes

- For the slave axis, the actual value in the "System" menu, "Service Display" softkey, exceeds 360 degrees even if modulo operation has been set for the axis using the following machine data: MD30310 ROT_IS_MODULO
- The automatic basic display and the service display do not show the actual value "Modulo 360°" until the coupling is deactivated.

Spindles

- If a master-slave coupling is activated with spindles, the slave axis is in speed control mode. In this case too, the actual value of the slave axis exceeds 360 degrees in the service display. No modulo calculation is active. However, the value shown in the automatic basic display is "Modulo 360°".
- If the spindles are accelerated at the current limit, this may mean that no adjusting reserves are left over in the coupled state for the torque compensation controller to use in order to distribute the torque between the master and the slave as required.
- The maximum master spindle chuck speed is to be configured as less than or equal to that of the slave spindles: MD35100 SPIND_VELO_LIMIT
- The axial velocity monitoring function should be adapted to the chuck speed: MD36200 AX_VELO_LIMIT

23.12 Examples

23.12.1 Master-slave coupling between AX1=Master and AX2=Slave.

Configuration

1. Machine axis number of master axis for speed setpoint coupling
MD37250 MS_ASSIGN_MASTER_SPEED_CMD[AX2] = 1
2. Master axis with torque distribution identical to master axis with speed setpoint coupling
MD37252 MS_ASSIGN_MASTER_TORQUE_CTRL[AX2] = 0
3. Parameter assignment for the master-slave coupling
MD37253 MS_FUNCTION_MASK = 1
4. Permanent coupling
MD37262 MS_COUPLING_ALWAYS_ACTIVE[AX2] = 1
5. Torque is injected in both the master and slave axes
MD37254 MS_TORQUE_CTRL_MODE[AX2] = 0
6. Torque distribution between the master and slave axes is 50% to 50%
MD37268 MS_TORQUE_WEIGHT_SLAVE[AX2] = 50
7. Parameters of the torque compensatory controller
MD37256 MS_TORQUE_CTRL_P_GAIN[AX2] = 0.5
MD37258 MS_TORQUE_CTRL_I_TIME[AX2] = 5.0

23.12.2 Close coupling via the PLC

This application allows you to close or separate a master-slave coupling between the machine axes AX1=Master axis and AX2=Slave axis during operation.

Preconditions

- One configured master axis
MD37250 MS_ASSIGN_MASTER_SPEED_CMD ≠ 0
- Activation of a master-slave coupling via
MD37262 MS_COUPLING_ALWAYS_ACTIVE=0
- The coupling is open.

Typical sequence of operations

| Action | Effect/comment |
|---------------------------------|---|
| • Approach coupling position | Each axis moves to the coupling position. |
| • Close coupling mechanically | Both axes are mechanically coupled to one another. |
| • Request to close the coupling | PLC interface signal "Master/slave on" V380x 5000.7 is set. |
| • Read back coupling state | When the axis is at a standstill, the coupled slave axis sets PLC interface signal "Master/slave active" V390x 5000.7 and clears "Position controller active" V390x 0001.5. Wait for checkback signal. |
| • Move master-slave grouping | The master axis is moved. |

23.12.3 Close/separate coupling via part program

This application allows you to close or separate a master-slave coupling between the machine axes AX1=Master axis and AX2=Slave via the part program.

Preconditions

- One configured master axis (MD37250)
- Activation of a master-slave coupling via the following machine data:
MD37262 MS_COUPLING_ALWAYS_ACTIVE= 0 (permanent master-slave coupling)
- The coupling is open.

Parts program

```

N10 G0 AX1=0 AX2=0           ; The axes moves to the coupling position.
N20 MASLON (AX2)             ; Close coupling temporarily. Both axes are mechanically
                             coupled to one another.
N30 AX1=100;                 ; Move master-slave grouping. Coupled via the speed
                             setpoint, the slave axis follows the master axis.
N40 MASLOF (AX2)             ; Open the coupling. The axes are mechanically
                             disconnected from one another.
N50 AX1=200 AX2=200;        ; The axes are traversed separately.
N60 M30
    
```

23.12.4 Release the mechanical brake

This application allows implementation of a brake control for machine axes AX1=Master axis and AX2=Slave axis in a master-slave coupling.

Preconditions

- Master-slave coupling is configured.
- Axes are stationary.
- No servo enable signals.

Typical sequence of operations

| Action | Effect/comment |
|-------------------------------|--|
| Request to close the coupling | The following PLC interface signal is set: V380x 5000.7 (master/slave on) |
| Set controller enable | The following PLC IS is set for both axes: V380x 0002.1 (servo enable) |
| Interpreting the feedback | Link the following PLC IS for the master axis using AND: V390x 0001.7 (current controller active) V390x 0001.6 (speed controller active) V390x 0001.5 (position controller active) Link the following PLC IS for the slave axis using AND: V390x 0001.7 (current controller active) V390x 0001.6 (speed controller active) V390x 5000.7 (master/slave active) |
| Release brakes | If the result of the AND operations on the master and slave axes is ≠ 0, the brake may be released. |

23.13 Data lists

23.13.1 Machine data

| Number | Identifier | Description |
|-----------------------|-----------------------------|---|
| Axis/spindle-specific | | |
| 37250 | MS_ASSIGN_MASTER_SPEED_CMD | Master axis for speed setpoint coupling |
| 37252 | MS_ASSIGN_MASTER_TORQUE_CTR | Master axis for torque distribution |
| 37254 | MS_TORQUE_CTRL_MODE | Connection of torque control output |
| 37255 | MS_TORQUE_CTRL_ACTIVATION | Activate torque compensatory control |
| 37256 | MS_TORQUE_CTRL_P_GAIN | Gain factor of torque compensatory controller |
| 37258 | MS_TORQUE_CTRL_I_TIME | Reset time for torque compensatory controller |
| 37260 | MS_MAX_CTRL_VELO | Limitation of torque compensatory control |
| 37262 | MS_COUPLING_ALWAYS_ACTIVE | Permanent master-slave coupling |
| 37263 | MS_SPIND_COUPLING_MODE | Coupling characteristics of a spindle |
| 37264 | MS_TENSION_TORQUE | Master-slave tension torque |
| 37268 | MS_TORQUE_WEIGHT_SLAVE | Torque weighting of the slave axis |
| 37270 | MS_VELO_TOL_COARSE | Master-slave velocity tolerance "coarse" |
| 37272 | MS_VELO_TOL_FINE | Master-slave velocity tolerance "fine" |
| 37274 | MS_MOTION_DIR_REVERSE | Invert master-slave direction of travel |

23.13.2 Interface signals

| Number | .Bit | Name |
|---------------|------|---|
| Axis-specific | | |
| V380x 5000 | .4 | Activate torque compensatory controller |
| V380x 5000 | .7 | Activate master-slave coupling |
| V390x 5000 | .2 | Velocity tolerance "fine" |
| V390x 5000 | .3 | Velocity tolerance "coarse" |
| V390x 5000 | .4 | State of torque compensation controller |
| V390x 5000 | .7 | State of master-slave coupling |

Feed (V1)

24.1 Path feedrate F

Functionality

The feedrate F is the **path velocity** of the tool along the programmed workpiece contour. The individual axis velocities therefore result from the portion of the axis path in the overall distance to be traversed.

The feedrate F is effective for the interpolation types G1, G2, G3, CIP, and CT and is retained in a program until a new F word is written.

Reference:

/BP_/ Operation and Programming

Dimension units for F: G94, G95

The dimension unit for the F word is determined by G functions:

- G94 F as feedrate in mm/min or inch/min
- G95 F as feedrate in mm/rev of the spindle or inch/rev
(only meaningful when the spindle is running)

The inch dimension system applies with G700 or system setting "inch" with MD10240 SCALING_SYSTEM_IS_METRIC=0.

Dimension units for F with G96, G97

For **lathes** the group with G94, G95 has been extended by the G96, G97 functions for the **constant cutting rate** (ON/OFF). These functions also influence the S word.

With activated G96 function, the spindle speed is adapted to the currently machined workpiece diameter (transverse axis) such that a programmed cutting rate S remains constant on the tool edge (spindle speed times diameter = constant).

The S word is evaluated as the cutting rate as of the block with G96. G96 is modally effective until cancellation by another G function of the group (G94, G95, G97).

The feedrate F is always evaluated in the unit of dimension of mm/rotation or inch/rotation (as for G95).

Maximum tool path velocity

The maximum path velocity is obtained from the maximum velocities of the relevant axes (MD32000 MAX_AX_VELO) and their proportion of the path. The maximum velocity of an axis stored in the machine data cannot be exceeded.

CFC feedrate override for circles

When machining circular contours using milling tools and the active tool radius compensation (G41/G42), the feedrate at the milling cutter center must be adjusted if the programmed F value is intended to be active at the circular contour. If the **CFC** feedrate override is active, inside and outside circle machining is detected automatically.

The feedrate override can be switched-off using **CFTCP**.

Reference:

/BP_/ Operation and Programming

Interface signals

If the revolutionary feedrate is active, IS "Revolutional feedrate" (V3300 0001.2) is set.

If the G96/G332 function is active, the IS "Constant cutting rate active" (V390x 2002.0) is set for the spindle.

Alarms

- If no F word is programmed at G1, G2, G3, ..., alarm 10860 is issued. An axis movement is not possible. However, please note: SD42110 DEFAULT_FEED!
- If F0 is programmed, alarm 14800 is issued.
- If G95 is active and the spindle is stationary, an axis movement is not possible. No alarm is issued.

Notes

- If the "Dry run feedrate" function is activated and the program is started, the feedrates programmed in combination with G1, G2, G3, CIP, CT are replaced by the feedrate value stored in SD42100 DRY_RUN_FEED, see Section "Program processing with dry run feedrate (DRY)(K1)".
- The velocity of the traversing movement of an axis in the JOG mode is determined by the machine data/setting data. Please refer to Chapter "Manual and Handwheel Travel (H1)" for a detailed description of the velocities, including the possible "Rapid traverse override".

24.1.1 Feedrate with G33, G34, G35 (thread cutting)

Note

The thread cutting function with G33, G34, and G35 is not available for versions G/N plus and pro.

Types of thread cutting

G33 - thread with constant pitch

G34 - thread with (linearly) increasing pitch

G35 - thread with (linearly) decreasing pitch

Axis velocity

With respect to G33, G34, or G35 threads, the axis velocity for the thread length results from the set spindle speed and the programmed pitch. However, the maximum axis velocity defined in MD32000 MAX_AX_VELO cannot be exceeded.

The feedrate F is not relevant. It is, however, kept in the memory.

The axis velocity, e.g. for a cylinder thread, results from the set spindle speed (S) and the programmed pitch (K):

$$F_z \text{ [mm/min]} = \text{speed } S \text{ [rev/min]} * \text{pitch } K \text{ [mm/rev]}$$

Note

For G34 and G35 the pitch change in mm/rev² is programmed under the F address.

Reference:

/BP_/ Operation and Programming

Minimum spindle speed

In order to ensure smooth rotation at low speeds, the spindle speed is not permitted to fall below a minimum level. This speed can be set in SD43210 SPIND_MIN_VELO_G25 (minimum spindle speed) and for each gear stage with MD35140 GEAR_STEP_MIN_VELO_LIMIT (minimum speed for gear stage change). The minimum spindle speed can be changed in the part program with G25.

NC stop, single block

NC stop and single block are only active after completion of thread chaining.

Information

- The spindle speed override switch should remain unchanged during thread machining (tapping).
- The feedrate override switch is irrelevant in a block with G33, G34, G35.

Programmable runin and runout path: DITS, DITE

The run-in and run-out path is to be traversed in addition to the required thread. The starting and braking of the axis (both axes in case of a tapered thread) are performed in these areas. This path depends on the pitch, spindle speed, and the axis dynamics (configuration).

If the available path for run-in or run-out is limited, it may be necessary to reduce the spindle speed so that this path is sufficient. In this case, the run-in and run-out paths can be specified separately in the program to achieve favorable cutting values and short machining times or to simplify the handling of this issue.

If no values are specified, the values from the setting data (SD) apply. The specifications in the program are written in SD42010 THREAD_RAMP_DISP[0] ... [1].

If this path is not sufficient for traversing at the configured axis acceleration, the axis is overloaded in terms of acceleration. Alarm 22280 ("Programmed run-in path too short") is then issued for the thread run-in. The alarm is purely for information and has no effect on part program execution.

The run-out path acts as an approximate distance at the end of the thread. This achieves a smooth change in the axis movement when retracting.

Programming

DITS= ...: Run-in path of the thread

DITE= ...: Run-out path of the thread

Reference:

/BP_/ Operation and Programming

SD42010

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

With the part program instructions, the setting data SD42010 THREAD_RAMP_DISP[0], ...[1] defines the following acceleration response of the axis during thread cutting ([0]-run-in, [1]-run-out):

- SD42010 = < 0 to -1:
Starting/braking of the feedrate axis at configured acceleration rate. Jerk according to current BRISK/SOFT programming.
- SD42010 = 0:
Abrupt starting/braking of the feedrate axis on thread cutting.
- SD42010 = > 0:
The thread run-up/deceleration distance is specified. To avoid technology alarm 22280, the acceleration limits of the axis must be observed in case of very small run-in and run-out paths.

Note

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

Pitch change F with G34, G35

If you already know the starting and final lead of a thread, you can calculate the pitch change F to be programmed according to the following equation:

$$F = \frac{|K_e^2 - K_a^2|}{2 * L_G} \text{ [mm/rev}^2\text{]}$$

The identifiers have the following meanings:

| | |
|-------|--|
| K_e | Pitch of axis target point coordinate [mm/rev] |
| K_a | Initial pitch (progr. under I and K) [mm/rev] |
| L_G | Thread length in [mm] |

24.1.2 Feedrate for G63 (tapping with compensation chuck)

Note

The tapping function with compensation chuck is not available for versions G/N plus and pro.

Feedrate F

In the case of G63 it is necessary to program a feedrate F. It must be suitable for the selected spindle speed S (programmed or set) and for the pitch of the drill:

Feedrate F [mm/min] = speed S [rev/min] x pitch [mm/rev]

The compensation chuck absorbs possible path differences of the drill axis to a limited extent.

References

/BP_/ Operation and Programming

24.1.3 Feedrate for G331, G332 (tapping without compensation chuck)

Note

The tapping function without compensation chuck is not available for versions G/N plus and pro.

Axis velocity

With respect to G331/G332 tapping, the axis velocity for the thread length results from the effective spindle speed S and the programmed pitch. However, the maximum axis velocity defined in MD32000 MAX_AX_VELO cannot be exceeded.

The feedrate F is not relevant. It is, however, kept in the memory.

Interface signal

If the G331/G332 function is active, the IS "Tapping without compensation chuck active" (V390x 2002.3) is set for the spindle.

Note

The tapping may only be carried out without a compensation chuck if an exact dynamic adjustment of the spindle and the relevant axis has been performed. With G331/G332 the parameter set n (0...5) of the axis becomes effective automatically. This parameter set also applies to the current gear stage of the spindle (M40, M41 to M45 - see also Chapter "Spindle (S1)").

In general, the axis is adjusted to the slower spindle.

References

/BP_/ Operation and Programming

24.1.4 Feedrate for chamfer/rounding: FRC, FRCM

Chamfer/rounding

You can insert the chamfer (CHF or CHR) or rounding (RND) elements into a contour corner. If you wish to round several contour corners sequentially by the same method, use "Modal rounding" (RNDM).

You can program the feedrate for the chamfer/rounding with FRC=... (non-modal) or FRCM=... (modal). If FRC/FRCM is not programmed, the normal feedrate F is applied.

Programming

| | |
|----------|---|
| FRC=... | Non-modal feedrate for chamfer/rounding Value > 0: Feedrate in mm/min (G94) or mm/rev. (G95) |
| FRCM=... | Modal feedrate for chamfer/rounding Value > 0: Feedrate in mm/min (G94) or mm/rev. (G95) Modal feedrate for chamfer/rounding ON Value = 0: Modal feedrate for chamfer/rounding OFF Feedrate F applies to the chamfer/rounding |

Notes

- F, FRC, FRCM are not active when a chamfer is traversed with G0. If the feedrate F is active for chamfer/rounding, it is by default the value from the block which leads away from the corner. Other settings can be configured via machine data MD20201 CHFRND_MODE_MASK.
- A maximum of three blocks without corresponding information may be put between two blocks containing traversing information for chamfer/rounding (axes of the plane). In the case of more blocks without axis information in the plane and existing instructions for inserting chamfer or rounding, an alarm is triggered.

24.2 Rapid traverse G0

Application

The rapid traverse movement G0 is used for rapid positioning of the tool, but not for direct workpiece machining. All axes can be traversed simultaneously. This results in a straight path.

For each axis, the maximum speed (rapid traverse) is defined in machine data MD32000 MAX_AX_VELO. If only one axis traverses, it uses its rapid traverse. If, for example, two axes are traversed simultaneously, the path velocity (resulting velocity) is selected to achieve the maximum possible path velocity under consideration of both axes.

If, for example, two axes have the same maximum velocity and also travel the same path, the path velocity = $1.41 * \text{max. axis velocity}$.

The feedrate F is not relevant for G0. It is, however, kept in the memory.

Rapid traverse override

In the AUTOMATIC operating mode, it can be set through the "Position" operation area -> "Program control" softkey that the feedrate override switch also applies to the rapid traverse. The active function is displayed with ROV in the status line. HMI to PLC sets the IS "Feedrate override for rapid traverse selected" (V1700 0001.3). The PLC user program must place this signal on the IS "Rapid traverse override active" (V3200 0006.6).

24.3 Feedrate control

24.3.1 Overview

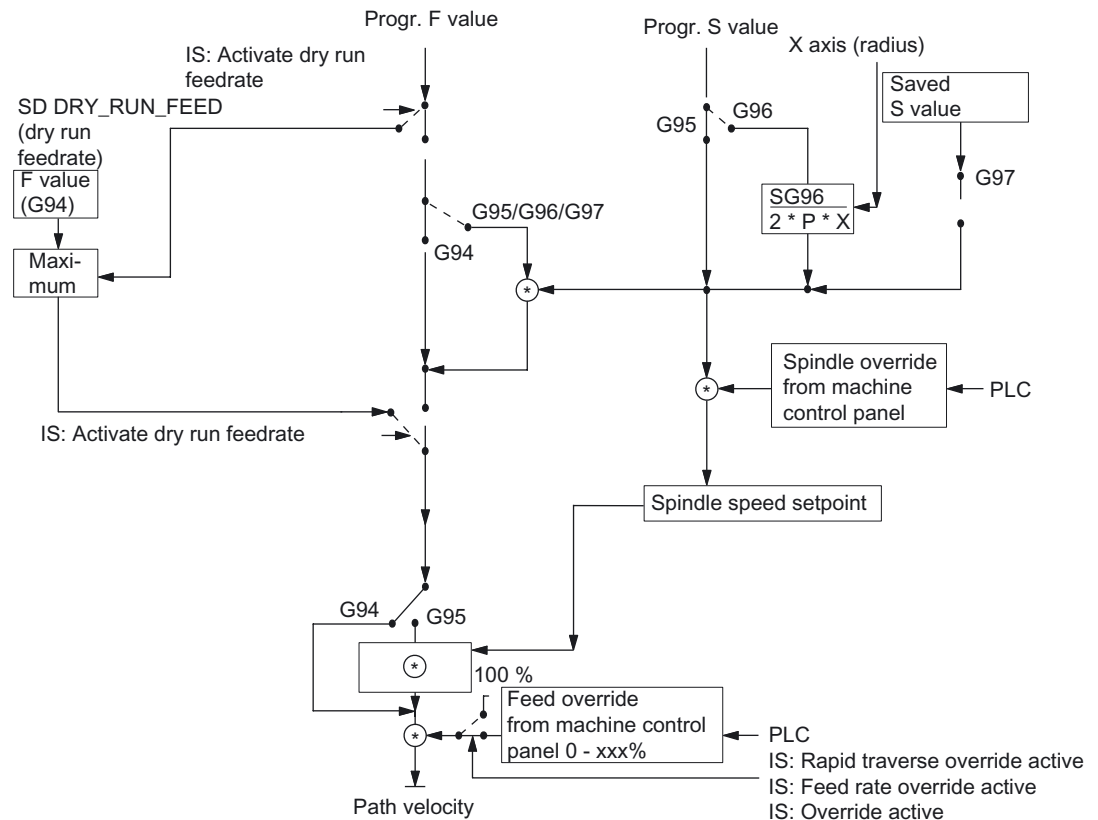


Figure 24-1 Possibilities for programming and controlling the feedrate

24.3.2 Feedrate disable and feedrate/spindle stop

General

The "Feed disable" or "Feed/spindle stop" brings the axes to a standstill. The path contour is maintained (exception: G33 block).

Feed disable

The channel-specific interface signal "Feed disable" (V3200 0006.0) will stop all axes (geometry and special axes) in all operating modes.

This feed disable is **not** effective if G33 is active; it is, however, active with G63, G331, G332.

Feed stop for axes in the WCS

The "Feed stop" interface signals (V3200 1000.3, V32001 004.3, and V3200 1008.3) are used to stop the geometry axes (axes in the WCS) during traversing in the workpiece coordinate system (WCS) in the JOG mode.

Axis-specific feed stop

The axis-specific "Feed stop" interface signal (V380x 0004.3) is used to stop the relevant machine axis.

In automatic mode: If the "Feed stop" is performed for a path axis, all the axes traversed in the current block and all axes participating in the axis group are stopped.

Only the current axis is stopped in JOG mode.

The axis-specific "Feed stop" is active when G33 is active (but: contour deviations = thread error!).

Spindle stop

The "Spindle stop" interface signal (V380x 0004.3) is used to stop the spindle.

"Spindle stop" is active with G33 and G63.

| |
|------------------------------------|
| NOTICE |
| Contour deviations = thread error! |

24.3.3 Feedrate override via a machine control panel

General

The operator can use the feedrate override switch to increase or decrease the path feedrate relative to the programmed feedrate in percent with immediate effect. The feedrates are multiplied by the override values.

An override between 0 and 120% can be programmed for the path feedrate F.

The rapid traverse override switch is used to reduce the traversing velocity when testing a part program.

An override between 0 and 100% can be programmed for the rapid traverse.

The spindle override can be used to modify the spindle speed and the cutting rate (with G96). The override can be between 50 and 120%.

The override is not permitted to exceed the machinespecific acceleration and speed limits or generate a contour error.

The override acts on the **programmed values** before limits (e.g. G26) intervene.

Channel-specific feedrate and rapid traverse override

One enable signal and one byte are provided on the PLC interface for the override factor in percent for feedrate and rapid traverse:

- IS "Feedrate override" (VB3200 0004)
- IS "Feedrate override active" (V3200 0006.7)
- IS "Rapid traverse override" (VB3200 0005)
- IS "Rapid traverse override active" (V3200 0006.6)

The interface for the override (value) is supplied by a machine control panel via the PLC to the NC and it is Gray-coded.

An active feedrate override acts on all path axes. An active rapid traverse override acts on all axes traversing with rapid traverse.

If there is no dedicated rapid traverse override switch, the feedrate override switch can be used. In this case, feedrate overrides above 100% are limited to 100% for rapid traverse override.

The override to be active can be selected via the PLC or operator panel.

If the selection is made using the operator panel (display: ROV), the IS "Feedrate override for rapid traverse selected" (V1700 0001.3) is set and must be transferred by the PLC user program to the IS "Rapid traverse override active" (V3200 0006.6). The value itself is to be transferred by the PLC user program from a machine control panel to the IS "Rapid traverse override" (VB3200 0005).

The channel-specific feedrate and rapid traverse overrides are inactive if G33, G63, G331 and G332 are active.

Axis-specific feedrate override

One enable signal and one byte for the feedrate override factor in percent are available on the PLC interface for each axis:

- IS "Feedrate override" (VB380x 0000)
- IS "Override active" (V380x 0001.7)

If G33, G331, G331, G63 are active, the axis-specific feedrate override has no effect (is internally set to a fixed value of 100%).

Spindle override

One enable signal and one byte for the spindle override factor in percent are available on the PLC interface for each spindle:

- IS "Spindle override" (VB380x 2003)
- IS "Override active" (V380x 0001.7)

The additional signal IS "Feedrate override for spindle valid" (V380x 2001.0) allows the PLC user program to determine that the value of the IS "Feedrate override" (VB380x 0000) should apply.

The spindle override is active with G33, but it should not be actuated for reasons of accuracy; also active with G331, G332. In the case of G63, the spindle override is set to a fixed value of 100%.

Override active

The set override values are effective in all operating modes and machine functions. This applies if the IS "Rapid traverse override active", "Feedrate override active" or "Override active" are set.

An override factor of 0% acts as a feedrate disable.

Override inactive

When the override is inactive (i.e. the above interface signals are set to "0"), the override factor "1" is used internally for all switch positions (except from the 1st position), i.e. the override is **100%**.

Note

The 1st switch position of the Gray-coded interfaces for the value represents a special case. In this case, the override factor of the 1st switch position is also used if the IS "Rapid traverse override active", "Feedrate override active", "Override active" are not set. Thus **0%** is issued as the override value for axes (acts the same as "Feed disable"). The following applies to the spindle if the IS "Override active" is not set: Override value **50%**.

24.4 Data lists**24.4.1 Machine/setting data**

| Number | Identifier | Name |
|-------------------------------|--------------------------|--|
| General machine data | | |
| 10240 | SCALING_SYSTEM_IS_METRIC | Basic system metric |
| Channel-specific machine data | | |
| 20201 | CHFRND_MODE_MASK | Specifications regarding the chamfer/rounding behavior |
| Axis-specific machine data | | |
| 32000 | MAX_AX_VELO | Maximum axis velocity |
| 35100 | SPIND_VELO_LIMIT | Maximum spindle speed |
| Channel-specific setting data | | |
| 42100 | DRY_RUN_FEED | Dry run feedrate |
| 42010 | THREAD_RAMP_DISP | Acceleration behavior of the feedrate axis when thread cutting |
| 42110 | DEFAULT_FEED | Default value for path feed |

24.4.2 Interface signals

| Number | Bit | Name |
|-----------------------|-----|---|
| Channel-specific | | |
| V3200 0000 | .6 | Activate dry run feed |
| V3200 0004 | - | Feed override |
| V3200 0005 | | Rapid traverse override |
| V3200 0006 | .0 | Feed disable |
| V3200 0006 | .6 | Rapid traverse override active |
| V3200 0006 | .7 | Feed rate override active |
| V32001000 | .3 | Feed stop, geometry axis 1 |
| V32001004 | .3 | Feed stop, geometry axis 2 |
| V32001008 | .3 | Feed stop, geometry axis 3 |
| V1700 0000 | .6 | Dry run feed rate selected |
| V1700 0001 | .3 | Feed rate override selected for rapid traverse |
| V3300 0001 | .2 | Revolutional feed rate active |
| Axis/spindle-specific | | |
| VB380x 0000 | - | Feed override |
| VB380x 2003 | - | Spindle override |
| V380x 0001 | .7 | Override active (axis or spindle) |
| V380x 2001 | .0 | Feedrate override for spindle valid |
| V380x 0004 | .3 | Feed stop/spindle stop |
| V390x 2002 | .0 | Constant cutting rate active (spindle) |
| V390x 2002 | .3 | Tapping without compensation chuck active (spindle) |

Tool: Compensation and Monitoring (W1)

25.1 Tool and tool compensation overview

Characteristics

The SINUMERIK 802D sl control is capable of calculating the tool compensation data for different tool types (drill, milling cutter, turning tool, ...).

- Length compensation
- Radius compensation
- Storage of the tool compensation data in the tool offset memory
 - Tool identification with T numbers from 0 to 32000
 - Definition of a tool with a maximum of nine cutting edges (offset blocks) through D number
 - Cutting edge is described by tool parameters:
 - Tool type
 - Geometry: Length/radius
 - Wear: Length/radius
 - Cutting edge position (for turning tools)
- Tool change selectable: Immediately with T command or through M6
- Tool radius compensation
 - Compensation active for all interpolation types: linear and circular
 - Compensation at outer corners selectable: transition circle (G450) or equidistant intersection (G451)
 - Automatic detection of outer/inner corners

Detailed description:

References:

/BP_/ Operation and Programming

25.2 Tool

Select a tool

A tool is selected in the program with the T function. Whether the new tool is immediately loaded with the T function or with M6 depends on the setting in MD22550 TOOL_CHANGE_MODE (new tool offset with M function).

Value range of T

The T function can assume integer values from T0 (no tool) to T32000 (tool with the number 32000).

The following is the maximum that can be stored simultaneously in the control system:

- SINUMERIK 802D sl value: 32 tools
- SINUMERIK 802D sl plus: 64 tools
- SINUMERIK 802D sl pro: 128 tools.

25.3 Tool offset

Tool compensation through D function

A tool can have up to nine cutting edges. The nine tool cutting edges are assigned to the D functions D1 to D9.

The following maximum values for tool compensation blocks can be stored simultaneously in the control system:

- SINUMERIK 802D sl value: 32 data fields (D numbers)
- SINUMERIK 802D sl plus: 64 data fields (D numbers)
- SINUMERIK 802D sl pro: 128 data fields (D numbers)

The tool cutting edge is programmed with D1 (edge 1) to D9 (edge 9). The tool cutting edge always refers to the currently active tool. An active tool cutting edge (D1 to D9) without an active tool (T0) is inactive. Tool cutting edge D0 deselects all tool offsets of the active tool.

Selection of the cutting edge when changing tool

When a new tool (new T number) has been programmed and the old one replaced, the following options are available for selecting the cutting edge:

- The cutting edge number is programmed
- The cutting edge number is not programmed D1 is active automatically.

Activating the tool offset

D1 to D9 activates the tool compensation (offset) for a cutting edge on the active tool. Tool length compensation and tool radius compensation can be activated at different times:

- Tool length compensation (TLC) is performed on the first traversing motion of the axis on which the TLC is to act. This traversing motion must be a linear interpolation (G0, G1).
- Tool radius compensation (TRC) becomes active when G41/G42 is programmed in the active plane (G17, G18 or G19). The selection of tool radius compensation with G41/G42 is only permitted in a program block with G0 (rapid traverse) or G1 (linear interpolation).

Detailed description of the tool compensation (offset) including tool radius compensation:

Reference:

/BP_/ Operation and Programming, Section "Tool and tool offset"

25.4 Tool monitoring

Note

This function is not available with T/M value.

25.4.1 Tool monitoring overview

General

The tool monitoring is activated via the following machine data:

- MD18080 MM_TOOL_MANAGEMENT_MASK: Bit 1=1
- MD20310 TOOL_MANAGEMENT_MASK: Bit 1=1

The "Tool monitoring" function works without active tool management and allows the following monitoring types of the active cutting edge of the active tool for SINUMERIK 802D sl:

- Monitoring of the **tool life**
- Monitoring of the **workpiece count**

The above mentioned monitoring types can be activated simultaneously for a tool.

Monitoring counter

Monitoring counters exist for each monitoring type. The monitoring counters count from a set value >0 down to zero. When a counter has decremented to a value of ≤ 0 , the limit value is reached. An appropriate alarm message is issued and an interface signal output.

Another interface signal is output first if a tool cutting edge has reached its set prewarning limit.

System variable for type and condition of the monitoring

- \$TC_TP8[t] - Status of the tool with the number t:
 - Bit 0 =1/0: Tool is **active**/ spare tool
 - Bit 1 =1/0: Tool is **enabled**/ not enabled
 - Bit 2 =1/0: Tool is **disabled**/ not disabled
 - Bit 3: Reserved
 - Bit 4 =1/0: **Prewarning limit reached**/ not reached
- \$TC_TP9[t] - Type of the monitoring function for the tool with the number t:
 - = 0: No monitoring
 - = 1: Service-life-monitored tool
 - = 2: Quantity-monitored tool

These system variables can be read / written in the NC program and can be accessed through the control (HMI). As the "Tool management" function package is not available for SINUMERIK 802D these system parameters must be managed by the operator/programmer.

Reference:

/BP_/ Operation and Programming

System variables for tool monitoring data

Table 25- 1 Tool monitoring data

| Identifier | Description | Data type | Default value |
|-----------------|--|-----------|---------------|
| \$TC_MOP1[t,d] | Warning limit for tool life in minutes | REAL | 0.0 |
| \$TC_MOP2[t,d] | Residual tool life in minutes | REAL | 0.0 |
| \$TC_MOP3[t,d] | Warning limit for count | INT | 0 |
| \$TC_MOP4[t,d] | Residual unit quantity | INT | 0 |
| ... | ... | | |
| \$TC_MOP11[t,d] | Tool life setpoint | REAL | 0.0 |
| \$TC_MOP13[t,d] | Unit quantity setpoint | REAL | 0.0 |

t for tool number T, d for D number

System variables for active tool

The following can be read in the NC program via system variables:

- \$P_TOOLNO: Number of the active tool T
- \$P_TOOL: Active D number of the active tool

Interface signals

Some monitoring conditions are made available to the PLC:

- "Tool pre-warning limit reached" (V5300 0000.0)
- "Tool limit reached" (V5300 0000.1)
- "T number for tool pre-warning limit" (VD5300 1000)
- "T number for tool limit" (VD5300 1004)

The workpiece counter can be turned off from the PLC:

- "Turn off workpiece counter" (V3200 0013.5)

25.4.2 Tool life monitoring

Tool life monitoring is done for the tool cutting edge that is currently in use (active cutting edge D of the active tool T).

As soon as the path axes traverse (G1, G2, G3, ... but not for G0), the residual tool life (\$TC_MOP2[t,d]) of this tool cutting edge is updated. If the remaining service life of the cutting edge of a tool falls below the value of "Service life prewarning limit" (\$TC_MOP1[t,d]) during machining, the IS "Tool prewarning limit reached" (V5300 0000.0) and the IS "T number for tool prewarning limit" (VD5300 1000) are set.

If the remaining service life ≤ 0 , an alarm is issued and the IS "Tool limit reached" (V5300 000.1) as well as the IS "T number for tool limit" (VD5300 1004) are set. The tool changes to the "disabled" condition and cannot be programmed again until this condition changes. The operator must intervene and ensure that an operational tool for machining is provided.

\$A_MONIFACT system variable

The \$A_MONIFACT system variable (REAL data type) allows the monitoring clock to be run slower or faster. This factor can be set before using the tool, in order to take the different kinds of wear into consideration according to the workpiece material used, for example.

After control system power-up, Reset / End of program, the \$A_MONIFACT factor has the value 1.0. Real time is in effect. Examples for accounting:

- \$A_MONIFACT=1: 1 minute real time = 1 minute tool life which is decremented
- \$A_MONIFACT=0.1: 1 minute real time = 0.1 minute tool life which is decremented
- \$A_MONIFACT=5: 1 minute real time = 5 minute tool life which is decremented

Setpoint update with RESETMON()

The RESETMON(state, t, d, mon) function sets the actual value to the setpoint:

- For all cutting edges or only for a specific cutting edge of a specific tool
- For all monitoring types or only for a specific monitoring type.

Transfer parameters:

| | | |
|-----|-------|--|
| INT | state | Status of the command execution: |
| | = 0 | Successful execution |
| | = -1 | The cutting edge with the specified D number d does not exist. |
| | = -2 | The tool with the specified T number t does not exist. |
| | = -3 | The specified tool t does not have a defined monitoring function. |
| | = -4 | The monitoring function is not activated, i.e. the command is not executed. |
| INT | t | Internal T number: |
| | = 0 | For all tools |
| | <> 0 | For this tool (t < 0: Absolute value generation t) |
| INT | d | <i>Optional:</i> D number of the tool with the number t: |
| | > 0 | For this D number |
| | = 0 | All cutting edges of the tool t |
| INT | mon | <i>Optional:</i> bit-encoded parameters for the monitoring type (values like \$TC_TP9): |
| | = 1 | Tool life |
| | = 2 | Quantity |
| | = 0 | All actual values of the monitoring functions active for tool t are set to their setpoints. |

Note

Please note:

- RESETMON() is not active if the interface signal "Program test active" is set.
- The variable for the **state** feedback must be defined at the beginning of the program using a DEF statement: DEF INT state

You can also define a different name for the variable (instead of state, with a maximum of 15 characters, beginning with 2 letters). The variable is only available in the program in which it was defined.

The same applies for **mon**. If a value is required at all it can also be directly transferred as a number (1 or 2).

25.4.3 Workpiece count monitoring

The workpiece count of the active cutting edge of the active tool is monitored.

The workpiece count monitoring records all tool cutting edges used to produce a workpiece. If the workpiece count is changed by an operator input, the monitoring data of all the cutting edges that have become active since the last workpiece count are adjusted.

Updating the workpiece count by operator input or SETPIECE()

The workpiece count can be updated by an operator input (HMI) or in the NC program through the SETPIECE() function.

With the **SETPIECE** function, the user can refresh the workpiece count monitoring data of the tools that were involved in the machining process. All tools with the D numbers that have become active since the last time SETPIECE was activated are included in the update. If a tool is active at the time SETPIECE is called, this is counted as well.

As soon as a block with path axis movements is executed after SETPIECE() this tool is also considered for the subsequent SETPIECE call.

SETPIECE(x)

x = 1... 32000 The number of workpieces that have been produced since the last execution of the SETPIECE function. The counter status for the remaining part quantity (\$TC_MOP4[t,d]) is reduced by this value.

x = 0 Deleting all counts for the remaining workpiece count (\$TC_MOP4[t,d]) for the tools/D number that have been involved in machining since that point of time. Alternatively, deletion via operator input (HMI) is recommended.

Example

```

N10 G0 X100
N20 ...
N30 T1
N40 M6 D2
N50 SETPIECE(2)           ; $TC_MOP4[1,2 ] (T1,D2) is decremented by 2
N60 X... Y...
N100 T2
N110 M6 D1
N120 SETPIECE(4)         ; $TC_MOP4[2,1 ] (T2,D1) and $TC_MOP4[1,2 ] are
                          ; decremented by 4
N130 X... Y...
N200 T3
N210 M6 D2
N220 SETPIECE(6)         ; $TC_MOP4[3,2 ] (T3,D2) and $TC_MOP4[2,1 ] (T2,D1) and
                          ; $TC_MOP4[1,2 ] decremented by 6
N230 X... Y...
N300 SETPIECE(0)         ; Deletes all above $TC_MOP4[t,d ]
N400 M2

```

Note

The SETPIECE() function is not active during the block search.



Tip

Direct writing of \$TC_MOP4[t,d] is recommended only in simple cases. A subsequent block with the STOPRE command is required.

Setpoint refreshing

The setpoint update, i.e. setting the remaining workpiece counters (\$TC_MOP4[t,d]) to the workpiece count setpoint (\$TC_MOP13[t,d]) is typically performed via operator input (HMI). It can, however, also be performed through the RESETMON (state, t, d, ,mon) function as already described for the tool life monitoring.

Example:

```
DEF INT state ; At program start, define the variable for status feedback
...
N100 RESETMON(state,12,1,2) ; Setpoint updating of the unit counter for T12, D1
...
```

25.4.4 Examples of the service life monitoring

Activating the tool monitoring when the "Tool monitoring" option is present:

```
MD18080 MM_TOOL_MANAGEMENT_MASK = 0x02
MD20310 TOOL_MANAGEMENT_MASK= 0x02
```

Service life monitoring for tool 2, cutting edge 1 in the NC program

```
$TC_TP9[2,1]=1 ; Activation of the service life monitoring
$TC_MOP1[2,1]=100 ; Prewarning limit in minutes
$TC_MOP2[2,1]=245 ; Residual service life in minutes
$TC_MOP11[2,1]=800 ; Set service life in minutes
```

Service life monitoring for the active tool with active D number in the NC program

```
$TC_TP9[$P_TOOLNO,$P_TOOL]=1 ; Activation of the service life monitoring
$TC_MOP1[$P_TOOLNO,$P_TOOL]=200 ; Prewarning limit in minutes
$TC_MOP2[$P_TOOLNO,$P_TOOL]=602 ; Residual service life in minutes
$TC_MOP11[$P_TOOLNO,$P_TOOL]=700 ; Set service life in minutes
```


25.5 Special handling of tool compensation

For SINUMERIK 802D sl plus and pro, tool compensation (offset) can be handled as follows.

Influence of setting data

Using specific setting data the operator / programmer can influence the calculation of the length compensation of the used tool:

- SD42940 TOOL_LENGTH_CONST
(allocation of the tool length components to the geometry axes)
- SD42950 TOOL_LENGTH_TYPE
(allocation of the tool length components independent of tool type)

Note

The modified setting data will become effective with the next cutting edge selection.

Tool length and plane change (SD42940 TOOL_LENGTH_CONST)

Value of the setting data equal to 0:

The behavior corresponds to the standard definition: The lengths 1 to 3 in geometry and wear are assigned to the 1st to 3rd axes of the plane according to the active G17 to G19 and according to the tool type. If the active G17 to G19 changes, the axis assignment for the lengths 1 to 3 also changes because abscissa, ordinate and applicate are allocated to different geometry axes.

Reference:

/BP_/ Operation and Programming

Value of the setting data not equal to 0:

The assignment of the tool lengths 1 to 3 in geometry and wear to the geometry axes is performed according to the SD value and is **not** changed if the machining plane (G17 to G19) changes.

The assignment of the tool lengths 1 to 3 to the geometry axes for **turning tools** (tool types 500 to 599) results from the value of the setting data SD42940 in accordance with the following table:

| Plane/value | Length 1 | Length 2 | Length 3 |
|-------------|----------|----------|----------|
| 17 | Y | X | Z |
| 18*) | X | Z | Y |
| 19 | Z | Y | X |
| -17 | X | Y | Z |
| -18 | Z | X | Y |
| -19 | Y | Z | X |

*) Each value not equal to 0 which is not equal to one of the six listed values is evaluated as the value for 18.

25.5 Special handling of tool compensation

With respect to the values with a negative sign the assignment of length 3 is identical, length 1 and 2 are exchanged - compared to the assignment with the corresponding positive values.

The following table shows the assignment of the tool lengths 1 to 3 to the geometry axes for **drills / milling cutters** (tool types 100 to 299):

| Plane/value | Length 1 | Length 2 | Length 3 |
|-------------|----------|----------|----------|
| 17*) | Z | Y | X |
| 18 | Y | X | Z |
| 19 | X | Z | Y |
| -17 | Z | X | Y |
| -18 | Y | Z | X |
| -19 | X | Y | Z |

*) Each value not equal to 0 which is not equal to one of the six listed values is evaluated as the value for 17.

With respect to the values with a negative sign the assignment of length 1 is identical, length 2 and 3 are exchanged - compared to the assignment with the corresponding positive values.

Note

For representation in tables, it is assumed that geometry axes 1 to 3 are named X, Y, Z. The axis order (1st, 2nd and 3rd geometry axis) and not the axis identifier determines the assignment between an offset and an axis.

Length compensation for tool type (SD42950 TOOL_LENGTH_TYPE)

Value of the setting data equal to 0:

The behavior corresponds to the standard definition: The lengths 1 to 3 in geometry and wear are assigned to the actual **tool type** (milling cutter / drill or turning tool).

Reference:

/BP_/ Operation and Programming

Value of the setting data not equal to 0:

The assignment of the tool lengths is always independent of the actual tool type.

Value 1: Length assignment always as for milling tools.

Value 2: Length assignment always as for turning tools.

Notes

- The influence of these two setting data only refers to tool lengths. The tool radius is not affected.
- If SD42940 TOOL_LENGTH_CONST is set not equal to 0 and the value in SD42950 TOOL_LENGTH_TYPE is 1 or 2, the related table for the assigned tool type (milling or turning tool) applies in SD42940.

Example

SD42940 TOOL_LENGTH_CONST =18
SD42950 TOOL_LENGTH_TYPE =2

Explanation:

The active tool with the active D number always behaves as a turning tool in the length compensation (-> SD42950 =2).

The length assignment is performed in all planes G17 to G19 as for G18 (-> SD42940=18):

Length 1 -> X axis

Length 2 -> Z axis

if Y axis exists: Length 3 -> Y axis

The tool radius acts according to the actual tool type and the active plane.

25.6 Data lists**25.6.1 Machine data**

| Number | Identifier | Name |
|------------------|-------------------------|--|
| General | | |
| 18080 ** | MM_TOOL_MANAGEMENT_MASK | Memory reservation for the tool monitoring |
| Channel-specific | | |
| 20310 ** | TOOL_MANAGEMENT_MASK | Activation of the tool monitoring |
| 22360 | TOOL_PARAMETER_DEF_MASK | Definition of tool parameters |
| 22550 | TOOL_CHANGE_MODE | New tool offsets with M function |

** This machine data is not available with 802D sl value.

25.6.2 Interface signals

| Number | Bit | Name |
|------------------|-----|-------------------------------------|
| Channel-specific | | |
| V2500 0008 | .0 | T function 1 change |
| V2500 0010 | .0 | D function 1 change |
| VD2500 2000 | - | T function 1 |
| VD2500 5000 | - | D function 1 |
| V2500 1000 | .6 | M6 |
| V3200 0013 | .5 | Deactivate workpiece counter |
| V5300 0000 | .0 | Tool pre-warning limit reached |
| V5300 0000 | .1 | Tool limit value reached |
| VD5300 1000 | - | T number for tool pre-warning limit |
| VD5300 1004 | - | T number for tool limit value |

Appendix

A.1 List of abbreviations

| | |
|----------|---|
| A | |
| ALM | Active Line Module - infeed module for drives |
| ASCII | American Standard Code for Information Interchange: American coding standard for the exchange of information by means of 7-bit character coding |
| ASUB | Asynchronous subroutine |
| AUXFU | Auxiliary Function |

| | |
|----------|------------------------|
| B | |
| BA | Mode of operation |
| BAG | Mode group |
| BERO | Proximity limit switch |

| | |
|----------|--------------------------------|
| C | |
| CAD | Computer Aided Design |
| CNC | Computerized Numerical Control |
| CPU | Central Processing Unit |

| | |
|------------|---------------------------------|
| D | |
| DAC | Digital-to-Analog Converter |
| DB | Data Block (interface signals) |
| DRIVE-CLiQ | Drive Component Link with IQ |
| DRY | DRY run: Dry run feedrate |
| DW | Data Word |
| DWORD | Double Word (currently 32 bits) |

| | |
|----------|-------------------|
| E | |
| EN | European standard |

| | |
|----------|---|
| H | |
| HMI | Human Machine Interface, SINUMERIK user interface |
| MSD | Main Spindle Drive |
| HW | Hardware |

Appendix

A.1 List of abbreviations

| | |
|-----------------|--|
| HW Config | SIMATIC S7 tool for configuration and parameterization of hardware components within an S7 project |
| HW limit switch | Hardware limit switch |

| | |
|----------|---------------|
| I | |
| IBN | Commissioning |
| INC | Increment |
| IPO | Interpolator |

| | |
|----------|---------------------|
| J | |
| JOG | Jogging: Setup mode |

| | |
|----------|-----------------------------|
| K | |
| Kv | Gain factor of control loop |

| | |
|----------|---------------------|
| L | |
| LR | Position controller |

| | |
|----------|---|
| M | |
| MCP | Machine Control Panel |
| MD | Machine Data |
| MDI | Manual Data Automatic: Manual input |
| MCS | Machine Coordinate System |
| MPF | Main Program File: NC part program (main program) |
| MCP | Machine Control Panel |

| | |
|----------|--------------------------|
| N | |
| NC | Numerical Control |
| NCK | Numerical Control Kernel |
| IS | Interface Signal |
| NV | Work offset |

| | |
|----------|-------------------------------------|
| O | |
| OP | Operator Panel: Operating equipment |

| | |
|----------|--|
| P | |
| PC | Personal Computer |
| PCU | Panel Control Unit - CNC integrated into the operator panel for user interface, system software and soft PLC |

| | |
|----------|--|
| PI | Program Invocation (PI service, PI index → ASUB) |
| PLC | Programmable Logic Control |
| PROFIBUS | Process Field Bus: Serial data bus |
| PRT | Program Test |
| PTP | Point-To-Point |

| | |
|----------|-----------------------------------|
| R | |
| REF | Reference point approach function |
| REPOS | Reposition function |
| ROV | Rapid Override : Input correction |

| | |
|------------------------------|--|
| Switchgear protection | |
| SBL | Single Block |
| SD | Setting Data |
| SK | Softkey |
| SKP | Skip: Function for skipping a part program block |
| PLC | Programmable (logic) controller |
| LEC | Leadscrew error compensation |
| SW | Software |
| SW limit switch | Software limit switches |

| | |
|----------|---|
| D | |
| TRANSMIT | Transform Milling Into Turning: Coordination transformation for milling operations on a lathe |

| | |
|----------|---|
| V | |
| VDI | Internal communication interface between NCK and PLC |
| VDE | Verband Deutscher Elektrotechniker [Association of German Electrical Engineers] |

| | |
|----------|-----------------------------|
| W | |
| WCS | Workpiece Coordinate System |
| TLC | Tool Length Compensation |
| TRC | Tool Radius Compensation |
| WZ | Tool |
| TO | Tool Offset |

A.2 Feedback on the documentation

This document will be continuously improved with regard to its quality and ease of use. Please help us with this task by sending your comments and suggestions for improvement via e-mail or fax to:

E-mail: <mailto:docu.motioncontrol@siemens.com>

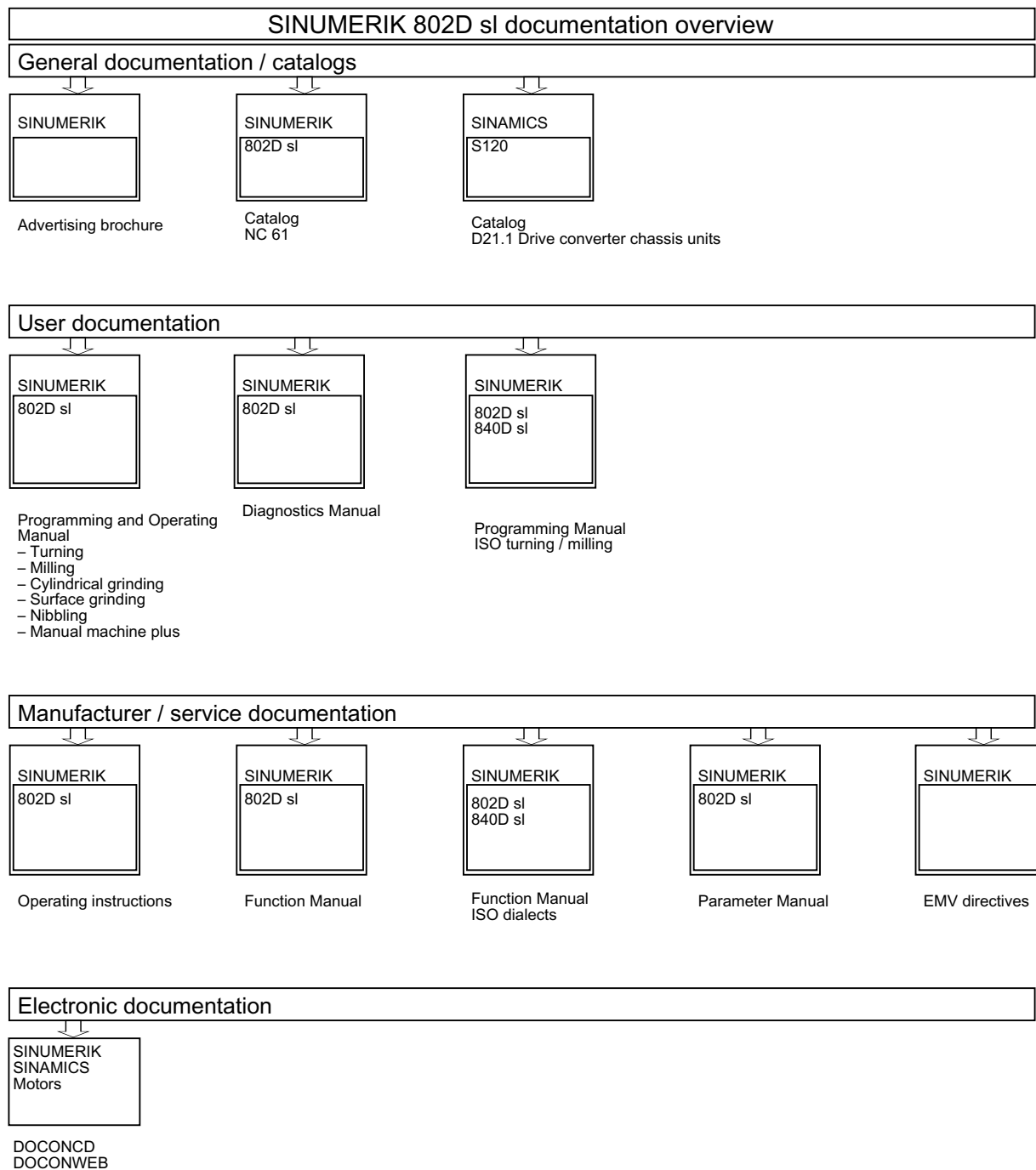
Fax: +49 9131 - 98 2176

Please use the fax form on the back of this page.

| | |
|--|------------------------------------|
| To SIEMENS AG I DT MC MS1 P.O. Box 3180 D-91050 Erlangen / Germany Fax: +49 9131 - 98 2176 (Documentation) | From |
| | Name: |
| | Address of your company/department |
| | Street: |
| | Zip code: City: |
| | Phone: / |
| Fax: / | |

Suggestions and/or corrections

A.3 Overview



Glossary

Absolute dimensions

A destination for an axis movement is defined by a dimension that refers to the origin of the currently active coordinate system. See → Incremental dimension.

Acceleration with jerk limitation

In order to optimize the acceleration response of the machine whilst simultaneously protecting the mechanical components, it is possible to switch over in the machining program between abrupt acceleration and continuous (jerk-free) acceleration.

Address

An address is the identifier for a certain operand or operand range, e.g. input, output, etc.

Alarms

All → messages and alarms are displayed on the operator panel in plain text with date and time and the corresponding symbol for the cancel criterion. Alarms and messages are displayed separately.

1. Alarms and messages in the part program:

Alarms and messages can be displayed in plain text directly from the part program.

2. Alarms and messages from the PLC

Alarms and messages for the machine can be displayed in plain text from the PLC program. No additional function block packages are required for this purpose.

Archive

Reading out of files and/or directories on an **external** memory device.

Asynchronous subroutine

Part program that can be started asynchronously to (independently of) the current program status using an interrupt signal (e.g. "Rapid NC input" signal).

Automatic

Operating mode of the control (block sequence operation according to DIN): Operating mode for NC systems in which a → part program is selected and executed continuously.

Auxiliary functions

Auxiliary functions enable part programs to transfer → parameters to the → PLC, which then trigger reactions defined by the machine manufacturer.

Axes

In accordance with their functional scope, the CNC axes are subdivided into:

- Axes: interpolating path axes
- Auxiliary axes: non-interpolating feed and positioning axes with an axis-specific feedrate. Auxiliary axes are not involved in actual machining, e.g. tool feeder, tool magazine.

Axis identifier

Axes are identified using X, Y and Z as defined in DIN 66217 for a right-handed, right-angled → coordinate system.

Rotary axes rotating around X, Y and Z are identified using A, B and C. Additional axes situated parallel to the specified axes can be designated using other letters.

Axis name

See → Axis identifier

Backlash compensation

Compensation for a mechanical machine backlash, e.g. backlash on reversal for ball screws. Backlash compensation can be entered separately for each axis.

Base axis

Axis whose setpoint or actual value position forms the basis of the calculation of a compensation value.

Basic Coordinate System

Cartesian coordinate system which is mapped by transformation onto the machine coordinate system.

The programmer uses axis names of the basic coordinate system in the → part program. The basic coordinate system exists parallel to the → machine coordinate system if no → transformation is active. The difference between the two coordinate systems lies in the → axis identifiers.

Baud rate

Rate of data transfer (Bit/s).

Blank

Workpiece as it is before it is machined.

Block

"Block" is the term given to any files required for creating and processing programs.

Block search

For debugging purposes or following a program abort, the "Block search" function can be used to select any location in the part program at which the program is to be started or resumed.

Booting

Loading the system program after POWER ON.

C axis

Axis around which the tool spindle describes a controlled rotational and positioning movement.

Channel

A channel is characterized by the fact that it can process a → part program. A channel exclusively controls the axes and spindles assigned to it.

Circular interpolation

The → tool moves on a circle between specified points on the contour at a given feedrate, and the workpiece is thereby machined.

CNC

See → NC

COM

Component of the NC control for the implementation and coordination of communication.

Compensation axis

Axis with a setpoint or actual value modified by the compensation value.

Compensation memory

Data range in the control, in which the tool offset data is stored.

Compensation table

Table containing interpolation points. It provides the compensation values of the compensation axis for selected positions on the basic axis.

Compensation value

Difference between the axis position measured by the encoder and the desired, programmed axis position.

Continuous-path mode

The objective of continuous-path mode is to avoid substantial deceleration of the → path axes at the part program block boundaries and to change to the next block at as close to the same path velocity as possible.

Contour

Contour of the → workpiece.

Contour monitoring

The following error is monitored within a definable tolerance band as a measure of contour accuracy. An unacceptably high following error can cause the drive to become overloaded, for example. In such cases, an alarm is output and the axes are stopped.

Coordinate system

See → Machine coordinate system, → Workpiece coordinate system.

CPU

Central processing unit, see → Programmable logic control.

Cycles

Protected subroutines for implementing a repetitive machining operation on the → workpiece.

Data Block

Data unit of the → NC: Data modules contain data definitions for global user data. These data can be initialized directly when they are defined.

Data word

Two-byte data unit within a → data block.

Diagnostics

The control has both a self-diagnostics program as well as test functions for servicing purposes: status, alarm, and service displays

Dimensions specification, metric and inches

Position and pitch values can be programmed in inches in the machining program. Irrespective of the programmable dimensions (*G70/G71*), the controller is set to a basic system.

Drive

The drive is the unit of the CNC that performs the speed and torque control based on the settings of the NC.

Dynamic feedforward control

Inaccuracies in the → contour due to following errors can be practically eliminated using dynamic, acceleration-dependent feedforward control. This results in excellent machining accuracy even at high → path velocities. Feedforward control can be selected and deselected on an axis-specific basis via the → part program.

Editor

The editor makes it possible to create, edit, extend, join, and import programs/texts/program blocks.

Exact stop

When an exact stop statement is programmed, the position specified in a block is approached exactly and, if necessary, very slowly. To reduce the approach time, → exact stop limits are defined for rapid traverse and feed.

Exact stop limit

When all path axes reach their exact stop limits, the control responds as if it had reached its precise destination point. A block advance of the → part program occurs.

External work offset

Zero point offset specified by the → PLC.

Feed override

The programmed velocity is overridden by the current velocity setting made via the → machine control panel or from the → PLC (0 to 120%). The feedrate can also be corrected by a programmable percentage factor (1-100%) in the machining program.

Fixed machine point

Point that is uniquely defined by the machine tool, e.g. machine reference point.

Fixed-point approach

Machine tools can approach fixed points such as a tool change point, loading point, pallet change point, etc. in a defined way. The coordinates of these points are stored in the control. The control traverses the relevant axes in → rapid traverse, whenever possible.

Frame

A frame is an arithmetic rule that transforms one Cartesian coordinate system into another Cartesian coordinate system. A frame contains the following components: → work offset, → rotation, → scaling, → mirroring.

Geometry

Description of a → workpiece in the → workpiece coordinate system.

Geometry axis

Geometry axes are used to describe a 2- or 3-dimensional area in the workpiece coordinate system.

Ground

Ground is taken as the total of all linked inactive parts of a device which will not become live with a dangerous contact voltage even in the event of a malfunction.

Helical interpolation

The helical interpolation function is ideal for machining internal and external threads using form milling cutters and for milling lubrication grooves.

The helix comprises two movements:

- Circular movement in one plane
- Linear movement perpendicular to this plane

High-level CNC language

The high-level language offers: → user-defined variables, → system variables, → macro techniques.

High-speed digital inputs/outputs

The digital inputs can be used for example to start fast CNC program routines (interrupt routines). High-speed, program-driven switching functions can be initiated via the digital CNC outputs

Identifier

In accordance with DIN 66025, words are supplemented using identifiers (names) for variables (arithmetic variables, system variables, user variables), subroutines, key words, and words with multiple address letters. These supplements have the same meaning as the words with respect to block format. Identifiers must be unique. It is not permissible to use the same identifier for different objects.

Inch measuring system

Measuring system which defines distances in inches and fractions of inches.

Increment

Travel path length specification based on number of increments. The number of increments can be stored as → setting data or be selected by means of a suitably labeled key (i.e. 10, 100, 1000, 10000).

Incremental dimension

Also incremental dimension: A destination for axis traversal is defined by a distance to be covered and a direction referenced to a point already reached. See → Absolute dimension.

Intermediate blocks

Motions with selected → tool offset (G41/G42) may be interrupted by a limited number of intermediate blocks (blocks without axis motions in the offset plane), whereby the tool offset can still be correctly compensated for. The permissible number of intermediate blocks which the control reads ahead can be set in system parameters.

Interpolator

Logic unit of the → NCK that defines intermediate values for the motions to be carried out in individual axes based on information on the end positions specified in the part program.

Interpolatory compensation

Interpolatory compensation is a tool that enables manufacturing-related leadscrew error and measuring system error compensations (LEC).

Interrupt routine

Interrupt routines are special → subroutines that can be started by events (external signals) in the machining process. A part program block which is currently being worked through is interrupted and the position of the axes at the point of interruption is automatically saved.

JOG

Control operating mode (setup mode): In JOG mode, the machine can be set up. Individual axes and spindles can be traversed in JOG mode by means of the direction keys. Additional

functions in JOG mode include: → Reference point approach, → Repos, and → Preset (set actual value).

Keywords

Words with specified notation that have a defined meaning in the programming language for → part programs.

Kv

Servo gain factor, a control variable in a control loop.

Ladder logic

Ladder logic is a graphical programming language (→ STEP 7) for representing electric circuit diagrams.

Leadscrew error compensation

Compensation for the mechanical inaccuracies of a leadscrew participating in the feed. The control uses stored deviation values for the compensation.

Limit speed

Maximum/minimum (spindle) speed: The maximum speed of a spindle can be limited by specifying machine data, the → PLC or → setting data.

Linear axis

In contrast to a rotary axis, a linear axis describes a straight line.

Linear interpolation

The tool travels along a straight line to the destination point while machining the workpiece.

Load memory

The load memory is the permanent memory of the → PLC.

Look Ahead

The **Look Ahead** function is used to achieve an optimal machining speed by looking ahead over an assignable number of traversing blocks.

Machine axes

Physically existent axes on the machine tool.

Machine control panel

An operator panel on a machine tool with operating elements such as keys, rotary switches, etc., and simple indicators such as LEDs. It is used to directly influence the machine tool via the PLC.

Machine Coordinate System

A coordinate system, which is related to the axes of the machine tool.

Machine zero

Fixed point of the machine tool to which all (derived) measuring systems can be traced back.

Macro techniques

Grouping of a set of instructions under a single identifier. The identifier represents the set of consolidated instructions in the program.

Main block

A block prefixed by ":" containing all the parameters required to start execution of a → part program.

Main memory

The main memory is a RAM in the → CPU that the processor accesses when processing the application program.

Main program

The → part program designated by a number or an identifier in which additional main programs, subroutines or → cycles can be called.

MDI

Control operating mode: Manual Data Automatic. In the MDA mode, individual program blocks or block sequences with no reference to a main program or subroutine can be input and executed immediately afterwards through actuation of the NC start key.

Messages

All messages programmed in the part program and → alarms detected by the system are displayed on the operator panel in plain text with date and time and the corresponding symbol for the cancel criterion. Alarms and messages are displayed separately.

Metric measuring system

Standardized system of units: For length, e.g. mm (millimeters), m (meters).

Mirroring

Mirroring reverses the signs of the coordinate values of a contour with respect to an axis. It is possible to mirror with respect to more than one axis at a time.

Mode of operation

An operating concept on a SINUMERIK control. The following modes are defined: → Jog, → MDA, → Automatic.

NC

Numerical Control: Numerical control (NC) includes all components of machine tool control: → NCK, → PLC, HMI, → COM.

Note

A more correct term for the SINUMERIK 802D sl control would be: Computerized Numerical Control

NCK

Numerical Control Kernel: Component of the NC that executes the → part programs and basically coordinates the motion operations for the machine tool.

Network

Thanks to the integrated network adapter, the control system is network-capable. The following connections are possible:

- Peer-to-Peer: Direct connection between control system and PC using a cross-over cable
- Twisted Pair: Incorporation of the control system into an existing local network using a patch cable.

OEM

The scope for implementing individual solutions (OEM applications) for the SINUMERIK 802D sl has been provided for machine manufacturers, who wish to create their own user interface or integrate process-oriented functions in the control.

Overall reset

In the event of an overall reset, the following memories of the → CPU are deleted:

- → Work memory
- Read/write area of → load memory
- → System memory

Override

Manual or programmable control feature, which enables the user to override programmed feedrates or speeds in order to adapt them to a specific workpiece or material.

Part program

Series of instructions to the NC that act in concert to produce a particular → workpiece. Likewise, this term applies to execution of a particular machining operation on a given → raw part.

Part program block

Part of a → part program that is demarcated by a line feed. There are two types: → main blocks and → subblocks.

Path axis

Path axes include all machining axes of the → channel that are controlled by the → interpolator in such a way that they start, accelerate, stop, and reach their endpoint simultaneously.

Path feedrate

Path feed affects → path axes. It represents the geometric sum of the feedrates of the → geometry axes involved.

Path velocity

The maximum programmable path velocity depends on the input resolution. For example, with a resolution of 0.1 mm the maximum programmable path velocity is 1000 m/min.

Peripheral module

I/O modules represent the link between the CPU and the process.

I/O modules are:

- → Digital input/output modules
- → Analog input/output modules

PLC

Programmable Logic Control Component of → NC: Programmable control for processing the control logic of the machine tool.

PLC Programming

The PLC is programmed with STEP 7. STEP 7 is the current programming software of the SIMATIC S7 PLC family and is the successor to STEP 5. As of Version 3.1, the user program can be created with the PLC 802 programming tool in → ladder logic.

Polar coordinates

A coordinate system, which defines the position of a point on a plane in terms of its distance from the origin and the angle formed by the radius vector with a defined axis.

Positioning axis

Axis that performs an auxiliary movement on a machine tool (e.g. tool magazine, pallet transport). Positioning axes are axes that do not interpolate with → path axes.

Program block

Program blocks contain the main program and subroutines of → part programs.

Programmable frames

Programmable → frames enable dynamic definition of new coordinate system output points while the part program is being executed. A distinction is made between absolute definition using a new frame and additive definition with reference to an existing starting point.

Programmable Logic Control

Programmable logic controls (PLC) are electronic controls, the function of which is stored as a program in the control unit. This means that the layout and wiring of the device do not depend on the function of the control. The programmable logic control has the same structure as a computer; it consists of a CPU (central module) with memory, input/output modules and an internal bus system. The peripherals and the programming language are matched to the requirements of the control technology.

Programmable working area limitation

Limitation of the motion space of the tool to a space defined by programmed limitations.

Protection zone

Three-dimensional zone within the → working area into which the tool tip must not pass.

R parameters

Arithmetic parameter that can be set or queried by the programmer of the → part program for any purpose in the program.

Rapid feed

The highest traverse rate of an axis. For example, rapid traverse is used when the tool approaches the → workpiece contour from a resting position or when the tool is retracted from the workpiece contour. The rapid traverse velocity is set on a machine-specific basis using a machine data element.

Rapid stop

A rapid stop is an NC function relating to error states. If an error state is detected, this is displayed by way of an alarm. A rapid stop is one of the responses which can be set for these alarms. As part of this, "0" is immediately specified as the speed setpoint for the machine axis affected, without a braking ramp. This will cause the relevant drives to brake to a standstill as quickly as possible; i.e. at their respective current limits.

Reference point

Machine tool position that the measuring system of the → machine axes references.

Rotary axis

Rotary axes apply a workpiece or tool rotation to a defined angular position.

Rotation

Component of a → frame that defines a rotation of the coordinate system around a particular angle.

Safety Functions

The control is equipped with permanently active monitoring functions that detect faults in the → CNC, the → PLC, and the machine in a timely manner so that damage to the workpiece, tool, or machine is largely prevented. In the event of a fault, the machining operation is interrupted and the drives stopped. The cause of the malfunction is logged and output as an alarm. At the same time, the PLC is notified that a CNC alarm has been triggered.

Scaling

Component of a → frame that implements axis-specific scale modifications.

Serial RS-232-C interface

A serial RS-232-C interface is available on the PCU for the data input/output. Machining programs and manufacturer and user data can be loaded and saved via this interface.

Setting data

Data which communicates the properties of the machine tool to the NC, as defined by the system software.

Softkey

A key whose name appears on an area of the screen. The choice of softkeys displayed is dynamically adapted to the operating situation. The freely assignable function keys (softkeys) are assigned defined functions in the software.

Software limit switch

Software limit switches limit the traversing range of an axis and prevent an abrupt stop of the slide at the hardware limit switch. Two value pairs can be specified for each axis and activated separately by means of the → PLC.

Spline interpolation

With spline interpolation, the control can generate a smooth curve characteristic from only a few specified interpolation points of a set contour.

SRT

Gear ratio

Standard cycles

Standard cycles are provided for machining operations which are frequently repeated:

- Cycles for drilling/milling applications
- For turning technology
- For grinding technology

The available cycles are listed in the <Technology> menu in the "Program" operating area. Once the desired machining cycle has been selected, the parameters required for assigning values are displayed in plain text.

Subblock

Block preceded by "N" containing information for a sequence, e.g. positional data.

Subroutine

Sequence of statements of a → part program that can be called repeatedly with different defining parameters. The subroutine is called from a main program. Every subroutine can be protected against unauthorized read-out and display. → Cycles are a form of subroutines.

Synchronization

Instructions in → part programs for coordination of sequences at certain machining points.

System variables

A variable that exists without any input from the programmer of a → part program. It is defined by a data type and the variable name preceded by the character \$. See → User-defined variable.

Tapping without compensation chuck

This function allows threads to be tapped without a compensation chuck. By using the interpolating method of the spindle as a rotary axis and the drilling axis, threads can be cut to a precise final drilling depth e.g. for blind hole threads (requirement: spindles in axis operation).

Text editor

The control system provides a function for editing Chinese characters in the program editor and in the PLC alarm text editor.

This function is only available in the Chinese language version.

Tool

Active part on the machine tool that implements machining (e.g. turning tool, milling tool, drill, etc.).

Tool nose radius compensation

Contour programming assumes that the tool is pointed. Because this is not actually the case in practice, the curvature radius of the tool used must be communicated to the control which then takes it into account. The curvature center is maintained equidistantly around the contour, offset by the curvature radius.

Tool offset

Consideration of the tool dimensions in calculating the path.

Tool radius compensation

To directly program a desired → workpiece contour, the control must traverse an equidistant path to the programmed contour taking into account the radius of the tool that is being used (G41/G42).

Transformation

Additive or absolute work offset of an axis.

Traversing range

The maximum permissible traversing range for linear axes is ± 9 decades. The absolute value depends on the selected input and position control resolution and the unit of measurement (inch or metric).

User interface

The menus and dialog boxes that the software displays on the monitor to aid the user in operating a program. It features horizontal and vertical softkeys.

User memory

All programs and data, such as part programs, subroutines, comments, tool offsets, and zero offsets/frames, as well as channel and program user data, can be stored in the shared CNC user memory.

User program

User programs for the S7-300 automation systems are created using the programming language STEP 7. The user program has a modular layout and consists of individual blocks.

The basic block types are:

- Code blocks

These blocks contain the STEP 7 commands.

- Data blocks

These blocks contain constants and variables for the STEP 7 program.

User-defined variable

Users can declare their own variables for any purpose in the → part program or data block (global user data). A definition contains a data type specification and the variable name. See → System variable.

Variable definition

A variable definition includes the specification of a data type and a variable name. The variable names can be used to access the value of the variables.

Velocity control

In order to achieve an acceptable traverse rate in the case of very slight motions per block, an anticipatory evaluation over several blocks (→ Look Ahead) can be specified.

Work offset

Specifies a new reference point for a coordinate system through reference to an existing zero point and a → frame.

1. Adjustable

A configurable number of settable work offsets are available for each CNC axis. The offsets - which are selected by means of G functions - take effect alternately.

2. Programmable

Work offsets can be programmed for all path and positioning axes using the TRANS instruction.

Working area

Three-dimensional zone into which the tool tip can be moved on account of the physical design of the machine tool. See → Protection zone.

Working area limitation

With the aid of the working area limitation, the traversing range of the axes can be further restricted in addition to the limit switches. One value pair per axis may be used to describe the protected working area.

Workpiece

Part to be made/machined by the machine tool.

Workpiece contour

Set contour of the → workpiece to be created or machined.

Workpiece coordinate system

The workpiece coordinate system has its starting point in the → workpiece zero-point. In machining operations programmed in the workpiece coordinate system, the dimensions and directions refer to this system.

Workpiece zero

The workpiece zero is the starting point for the → workpiece coordinate system. It is defined in terms of distances to the → machine zero.

Index

A

- Acceleration, 135
- Acceleration characteristic, 266
- Acceleration profiles, 67
 - Abrupt acceleration changes, 67
 - Acceleration with jerk limitation, 67
- Actual-value processing, 123
- Actual-value resolution, 123
- Alternate interface, 264
- Angularity error compensation, 203
- Applications, 355
- Approaching a fixed point
 - in JOG, 142
 - With G75, 142
- ASUB
 - Initializing, 33
 - Start, 178
- Auxiliary function output
 - Block change, 152
 - Block search, 155
- Auxiliary functions, 151
- Axis monitoring functions
 - Actual velocity, 47
 - Clamping, 45
 - Position monitoring, 42
 - Speed setpoint, 46
 - Standstill, 44
- Axisrelated jerk limitation, 68

B

- Backlash compensation, 197
- Basic system
 - Conversion, 112
 - manual changeover, 114
- Block search, 155, 185, 267

C

- Channel, 159
- Channel status, 169
- Clamping protection zone, 283
- Closed-loop control, 127
- Commands MEAS, MEAW, 241
- Compensation table, 200

- Computational resolution, 111
- Concurrent positioning axes, 295
- Continuous travel, 137
- Contour violation, 52
- Coupled motion, 279
- Cyclic signal exchange, 22

D

- D functions, 410
- Direction of rotation reversal, 374
- Display resolution, 111
- Drive variables, 126
- Dry run feedrate, 185

E

- EMERGENCY STOP
 - Acknowledgment, 251
 - Interface, 250
 - Sequence, 250
- EMERGENCY STOP pushbutton, 249
- Encoder monitoring functions
 - Encoder frequency, 48
 - Zero marks, 49
- Eventcontrolled program sequences
 - Operator panel reset, 173
 - Part program start and part program end, 172
- Exact stop, 58
- Exact-stop criteria, 59

F

- Feed disable, 403
- Feedrate override, 135
- Feedrate override switch, 404
- Feedrate/spindle stop, 403
- Feedrates
 - Axis-specific feedrate override, 405
 - Feed disable, 403
 - Feed override, 404
 - Feedrate control, 402
 - Feedrate/spindle stop, 403
 - Path feedrate F, 395
 - Spindle override, 405
 - Tapping with compensation chuck G63, 400
 - Thread cutting G33, 397

Fixed point positions, 145
Fixed stop, 71
Following block velocity, 65
Following error compensation, 215

G

Gantry
 Referencing and synchronizing, 86
Gantry axes
 Terms, 82
Gear stage change in master/slave, 388
Grooves, 227
Grouping auxiliary functions, 153

H

Hand wheel
 Traversal in JOG, 139
 Traversing the axes, 133
Hardware limit switches, 136

I

Implicit exact stop, 60
INC, 357
Incremental travel, 138
Indexing axes
 Coded position, 360
 Coded positions, 358
 Continuous traversal, 356
 Handwheel, 357
 Incremental traversal (INC), 357
 Parameterization, 358
 Programming, 360
 Reference point approach, 356
 Startup, 361
 Traversal from PLC, 358
Input resolution, 111
Interpolator end, 59
Interpolatory compensation
 Compensation table, 199
 Linear interpolation, 199

J

Jerk limit, 62
Jerk limitation, 62
JOG
 Approaching a fixed point, 142

L

Language command
 SPN, 269, 271
 SPP, 268, 270
Leading axis, 82
Leadscrew error compensation (LEC), 200
Linear axis
 Limit switch monitoring, 50
 Traversing range, 110
 with rotary encoder on motor, 124
LookAhead, 59, 63

M

Master-slave, 377
Measuring accuracy, 242
Measuring system error compensation (LEC), 200
Mode change, 161
Monitoring of the input signal, 265
Motion monitoring functions, 41

O

Offset angle, 371
Operating modes
 Error on operating mode changeover, 161
 Interlocks, 164
 Mode change disable, 161
 Monitoring, 163
Overload factor, 61

P

Parameter sets of the position controller, 128
Part program interruption, 166
Path axes, 57
Path feedrate F, 395
 Alarms, 396
 Feedrate for G63 (tapping with compensation
 chuck), 400
 Feedrate with G33 (thread cutting), 397
 Value range, 110
Path segmentation, 268
Pathrelated jerk limitation, 67
Permanently assigned PLC axis, 296
Physical quantities, 111
PI services, 31
Plausibility check, 310
PLC axis, 295
PLC service display, 242

PLC/NCK interface, 21
 Position control, 128
 Position control loop, 127
 Position controller gain, 44
 Position display, 197
 Positioning window, 44
 Pre-start time, 265
 Probe, 239
 Probe connection, 241
 Probe functional test, 243
 Prog Event, 170
 Program control, 167
 Program mode

- Part program interruption, 166
- Part program selection, 164
- Program status, 168
- RESET command, 167
- Start of part program or part program block, 165

 Program operation, 159
 Program status, 168
 Program test

- Block search, 185
- Processing of certain program sections, 185
- Program processing in singleblock mode, 183
- Skip part program blocks, 188

 Pulse inhibit, 77
 PUNCHACC, 266

R

Rapid traverse override, 135
 Rapid traverse override switch, 404
 Read measurement results in PP, 242
 Reducing jerk, 62
 Reference point approach, 356
 Referencing

- Axis-specific, 302
- Channel-specific, 302
- with incremental measuring system, 303

 Referencing with absolute encoders, 313
 Reset, 252
 Rotary axes

- Absolute programming, 322, 323
- Feedrate, 320
- Incremental programming, 323
- Modulo 360, 321
- Software limit switches, 321
- Traversing range, 110
- Units of measurement, 320

 Rotary axes,

- Axis addresses, 319

 Rotary axis

Definition, 319
 with rotary encoder on motor, 125

S

Sag compensation, 203
 Servo gain factor, 128
 Setpoint output, 117
 Signals from PLC to NCK, 22
 Simulation axes, 117
 Single block mode, 183
 Skip part program blocks, 188
 Software limit switch, 364
 Software limit switches, 136, 321
 Speed control loop, 127
 Spindle

- Gear stage change, 339
- Spindle monitoring, 345
- Synchronization, 337

 Spindle modes, 328
 Spindle override factor, 405
 Spindle speed, 110
 Start positioning axis, 296
 Synchronized axis, 83

T

T function, 156
 Tachogenerator compensation, 47
 Tangential angle, 369
 Tangential control, 279
 TEACH IN, 160
 Tool, 410

- Activating the tool offset, 411
- Select, 410
- Selection of the cutting edge when changing tool, 410
- Tool offset, 410
- Value range of T, 410

 Tool monitoring, 411
 Transverse axes

- Diameter programming, 291
- Geometry axes, 291

 Traversing ranges, 110

V

Velocities, 109
 Velocity reduction, 61
 Velocity reduction according to overload factor, 61

W

Working area limitation, 52, 321

