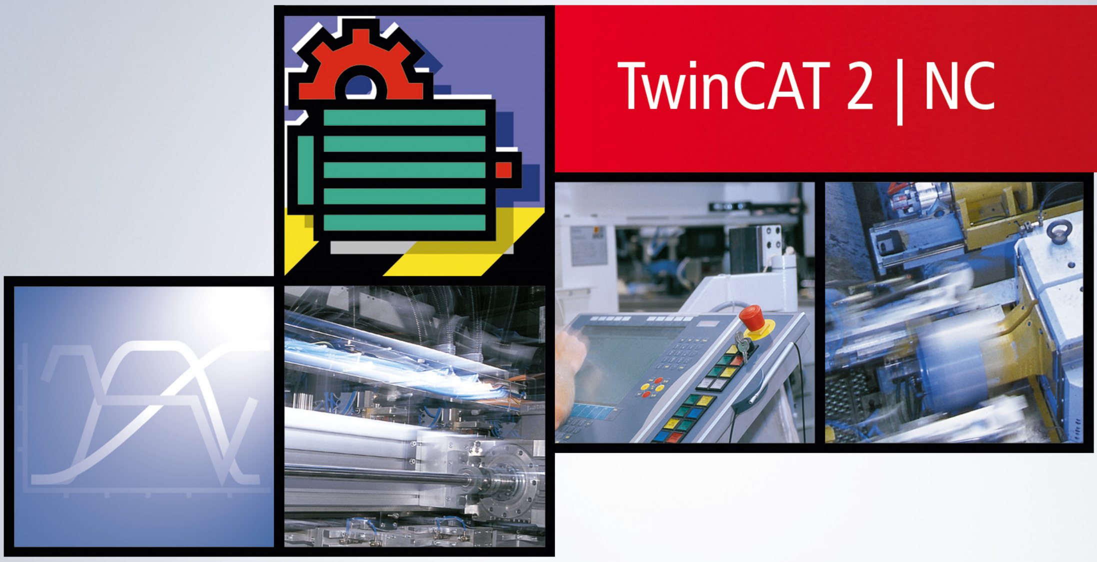
BECKHOFF New Automation Technology

Manual | EN

Numerical Control (NC) Basics

TwinCAT 2



2023-07-20 | Version: 1.2

Table of contents

1 Foreword

1.1 Notes on the documentation

This description is only intended for the use of trained specialists in control and automation engineering who are familiar with applicable national standards.
It is essential that the documentation and the following notes and explanations are followed when installing and commissioning the components.
It is the duty of the technical personnel to use the documentation published at the respective time of each installation and commissioning.

The responsible staff must ensure that the application or use of the products described satisfy all the requirements for safety, including all the relevant laws, regulations, guidelines and standards.

Disclaimer

Trademarks

Patent Pending

Copyright

1.2 For your safety

Safety regulations

Read the following explanations for your safety.
Always observe and follow product-specific safety instructions, which you may find at the appropriate places in this document.

Exclusion of liability

All the components are supplied in particular hardware and software configurations which are appropriate for the application. Modifications to hardware or software configurations other than those described in the documentation are not permitted, and nullify the liability of Beckhoff Automation GmbH & Co. KG.

Personnel qualification

This description is only intended for trained specialists in control, automation, and drive technology who are familiar with the applicable national standards.

Signal words

The signal words used in the documentation are classified below. In order to prevent injury and damage to persons and property, read and follow the safety and warning notices.

Personal injury warnings

⚠ DANGER
Hazard with high risk of death or serious injury.

⚠ WARNING
Hazard with medium risk of death or serious injury.

⚠ CAUTION
There is a low-risk hazard that could result in medium or minor injury.

Warning of damage to property or environment

NOTICE
The environment, equipment, or data may be damaged.

Information on handling the product

This information includes, for example:
recommendations for action, assistance or further information on the product.

1.3 Notes on information security

The products of Beckhoff Automation GmbH & Co. KG (Beckhoff), insofar as they can be accessed online, are equipped with security functions that support the secure operation of plants, systems, machines and networks. Despite the security functions, the creation, implementation and constant updating of a holistic security concept for the operation are necessary to protect the respective plant, system, machine and networks against cyber threats. The products sold by Beckhoff are only part of the overall security concept. The customer is responsible for preventing unauthorized access by third parties to its equipment, systems, machines and networks. The latter should be connected to the corporate network or the Internet only if appropriate protective measures have been set up.

In addition, the recommendations from Beckhoff regarding appropriate protective measures should be observed. Further information regarding information security and industrial security can be found in our https://www.beckhoff.com/secguide.

Beckhoff products and solutions undergo continuous further development. This also applies to security functions. In light of this continuous further development, Beckhoff expressly recommends that the products are kept up to date at all times and that updates are installed for the products once they have been made available. Using outdated or unsupported product versions can increase the risk of cyber threats.

To stay informed about information security for Beckhoff products, subscribe to the RSS feed at https://www.beckhoff.com/secinfo.

2 Overview

TwinCAT NC combines function groups that serve to control and regulate axes or synchronized axis groups. An NC task consists of one or more channels of type PTP channel, FIFO channel or NCI channel, and their subsidiary parts. Immediately after start-up the NC axes generally find themselves in one or more PTP channels. If necessary they are then moved to a different channel through a process of reconfiguration.

Safety functions

You should make yourself aware of the TwinCAT safety functions before commissioning any axes.

See: Safety functions [▶ 10]

Interfaces: System Manager and PLC

The insertion of an NC task, the configuration of the NC channels and the axes, and the use of most of the **functionalities** is described in the NC section of the System Manager, and in the following libraries: the NC Configuration Library, the NC Library and the NCI Interpreter Library.

Hardware I/O

There are input/output [▶ 41] connections for channels, axes, encoders and drives.

NC-PLC-Interface

The NC-PLC interface is a cyclic interface, with which on the one hand the PLC channel and axis data can be read and on the other hand the NC channel and axis parameters can be set.

Axes

TwinCAT supports a variety of axis types: servo axes, rapid/creep axes [▶ 14], stepper motor axes [▶ 21], "low-cost" stepper motor axes [▶ 26], encoder axes and simulation axes.

Axis components

Each axis consists of various elements, depending on the axis type. These elements include the encoder [▶ 31], the drive [▶ 34], the controller [▶ 36] and the setpoint generator [▶ 40], along with the PLC interface [▶ 41]. There are also more complex axes consisting of a number of encoders, switchable controllers or changing setpoint generators.

Axis commissioning

Axis commissioning includes knowledge and use of the NC safety functions, application of the necessary safety precautions, observation of a specific sequence of commissioning steps for the axis elements (encoder, drive and controller) and setting the axis parameters.

See:

- Axis commissioning > Overview [▶ 57]
- Safety functions [▶ 10]

PTP

PTP (Point To Point) axis functionality is a control process for one-dimensional positioning of axes, in particular of servo axes. One-dimensional does not necessarily mean linear. It simply means that one component is interpolated in some specified coordinate system (Cartesian coordinates, polar coordinates).

PTP forms the basis of the whole of TwinCAT NC, because at system startup the axes are normally in PTP mode, and are thus position controlled. The extended TwinCAT NC functionalities are achieved on the basis of the PTP modes by reconfiguration (FIFO, NCI) or by coupling (all slave types).

Slave axes

Axes whose set values (position, velocity, acceleration) are generated in functional dependency on the set values of other axes (master axes, virtual axes, slave axes, simulation axes) - in particular on their position setpoints - are called slave axes. Apart from the coupling and uncoupling, these axes do not generally have any functionality of their own (in particular they do not have start or stop functions) and operate under linear path control in synchronization with the master axis. The master axis can itself be a slave axis.

When a master axis (which is to become a slave axis) is coupled to another master axis, the slave axis retains its own position control.

FIFO

The TwinCAT NC FIFO group provides the facility of supplying externally generated setpoints to a group of axes via the NC, and thus of driving them synchronously. The position setpoints of the axes corresponding to a fixed, although selectable, time period (the FIFO cycle time) are present in the form of a FIFO which is kept topped up by the PLC. The FIFO allows the position of a group of axes to be handled synchronously with reference to time. The FIFO contains a fixed number of specified interpolation points between which the NC position is interpolated and from which the velocity is determined. The axis positions of the FIFO are either read out from a large table calculated off-line in the PLC or generated by a function in the PLC and permanently passed on to the FIFO of the NC by a PLC program in the PLC cycle.

NCI

TwinCAT NC I is the function group for controlling specified synchronized axis movements in three dimensions, according to the conventions of DIN 66025 and various extensions. It consists of interpreter/ PLC function blocks, set preparation task, look ahead, set execution task, cyclic channel interface and cyclic axis interface. On-line access is possible to all levels (interpreter stop, programmed halt, override). See: TwinCAT NCI documentation.

3 Safety functions

Axis commissioning includes:

1. In particular, the knowledge and use of NC security functionalities,
2. taking the necessary safety precautions and
3. compliance with a specific sequence of commissioning steps.

⚠ **DANGER**

Danger to life or risk of serious injury or damage to property due to unintentional movements of the axis

When commissioning axes, there is a movement of them and the mechanics coupled to them, which creates a hazard for people and a risk of damage to the machine. The following safety measures provide guidelines for safe commissioning. The actual measures to be taken depend on the axis and its surroundings.

As a general rule, "Don't take an action whose consequence you can't estimate."

A monitoring function (watchdog) is provided in order to monitor the operation and regular updating of the cyclical interface between the PLC and the NC. In addition, there is task runtime overrun monitoring for each task and position lag monitoring and end position monitoring for the axes. Finally, monitoring facilities are provided on the hardware side.

Make sure that you are aware of further safety procedures for axis commissioning. (See Axis commissioning > General [▶ 57])

Watchdog for cyclic axis interface between PLC and NC

The watchdog (function monitoring) functionality between the cyclic axis interface of the PLC and NC should always be activated. This is the case if any value other than zero is entered for the watchdog. The value specifies the number of sequential task cycles following which the watchdog will trigger if no new information has been transferred between the NC and the PLC. If the watchdog is triggered, the corresponding axis interface (PlcToNc or NcToPlc) is cleared, i.e. zeroed.

If for example the PLC is stopped, or an infinite loop has been programmed within the PLC, or an FPU exception occurs, the active watchdog ensures that the NC axes are halted because the watchdog will cause the enable for the controller and feed to be canceled.

Task time-out monitoring

For purposes of diagnosis and analysis the task time-out monitoring should be activated. This is true both for the SEC task (I/O task of the NC) and for the SPP task of the NC. As regards content this monitoring has no effect, but should there be an occasion where an unexpected task time-out occurs, the response is in the form of a message box and an additional entry in the event display.

Position lag monitoring, end position monitoring, target position control

Right from the start of any operations each axis should be driven with both "position lag monitoring" and "end position monitoring" active. Even if an axis that has not yet been optimized is moving with lag errors that are very large at times, these fundamental monitoring mechanisms should not be switched out. Instead, their parameters should be set correspondingly generously (see TwinCAT System Manager documentation > NC - Configuration > Settings dialog > Axes dialog: Global). Furthermore, there is the possibility to control the target position automatically.

Direction inversion, direction monitoring

There are functionalities, e.g. position compensation on a master axis, which can cause an inversion of the direction of movement. To avoid an unwanted direction of travel, there is a direction-dependent feed enable that stops the axis instantaneously if it travels in the wrong direction.

Maximum velocity

Furthermore there is the possibility to define the maximum allowed travel velocity of an axis in the axis parameters or to limit the output of the drive in percent (see TwinCAT System Manager Documentation > NC - Configuration > Settings Dialog > Axes - Dialog: Global).

For example, it can happen that, by mistake, the direction of actuation of the axis control loop (positive feedback) is altered as a result of changeover of drive or encoder polarity, and the axis, with full logical consistency, drives towards the mechanical end position at maximum output value.

In the following situations the maximum velocity can be exceeded:

- Through position compensation of the master or slave axes.
- Through setting or changing the coupling factor of a slave axis or (indirectly) of the flying saw.
- Through externally generated data in the FIFO or the table slave axes.

Stop

All master axes can be stopped at any time.

Notice **The flying saw is the only slave axis that has a stop function. However, there are situations when a flying saw cannot be stopped.**

Slave axes are to be stopped by converting them online into master axes, which can then be stopped. The FIFO axes can be stopped and the NCI group can be stopped.

Hardware monitoring

It may be that a facility in an emergency situation (emergency stop, watchdog, etc.) must not be allowed under any circumstances, for mechanical or other reasons, to halt abruptly its axes in the next I/O cycle (e.g. to command 0 V suddenly). Such behavior can only be ensured via the drive hardware that is present. To this end most manufacturers offer simple digital circuitry options that ensure that an axis is brought to a halt in a defined way (braking ramp, standstill window for electrical deactivation of the control system and activation of the brakes, etc.).

4 TwinCAT NC Axes

TwinCAT supports a variety of axis types: servo axes, rapid/creep axes, stepper motor axes, encoder axes and simulation axes.

Axis control loop

The axis control loop [▶ 13] consists of velocity pre-control, position controller and a controller limitation.

Slave axes

Axes whose set values (position, velocity, acceleration) are generated in functional dependency on the set values of other axes (master axes, virtual axes, slave axes, simulation axes) - on their set position values - are called **slave axes**. Apart from the coupling and uncoupling, these axes do not generally have any functionality of their own (they do not have start or stop functions) and operate under **linear path control** in synchronization with the master axis (which itself can in turn be a slave axis).

When a master axis (which is to become a slave axis) is coupled to another master axis, the slave axis retains its own position control.

The slave axes of NCI groups make up a special class

High/low speed axes

The TwinCAT high speed/low speed [▶ 14] axis type (two speed) allow the positioning of a high speed/low speed axis. Such a high speed/ low speed axis can be physically made up of a motor with two speeds (switching of the pole pair numbers), or alternatively by a motor that can be driven at two speeds with the help of a frequency converter.

"Low Cost" stepper motor axis with digital control (24V / 2A)

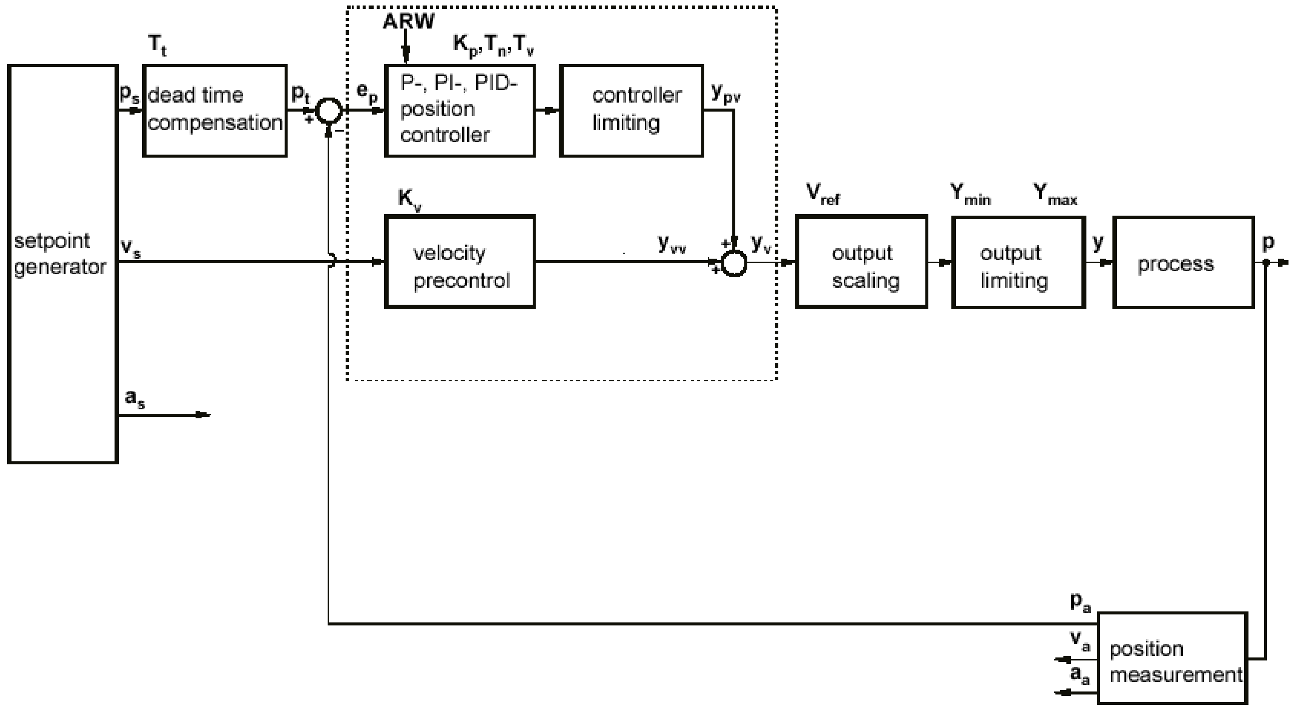
The basic version of the 'low cost stepper motor axis' [▶ 26] is operated without physical encoder (hence simulation encoder). This means that there is no actual physical feedback between set values and actual values, and therefore the axis is not operated with closed-loop control, but only with open-loop control. It is assumed that there is no slippage in the drive, and that the axis can accurately follow the specified step pattern (set value profile). Notwithstanding this constraint imposed for cost reasons, the stepper motor axis can be referenced physically, since the simulation encoder is implemented as an incremental encoder and supports virtually all features.

Functional overview of the individual axis types

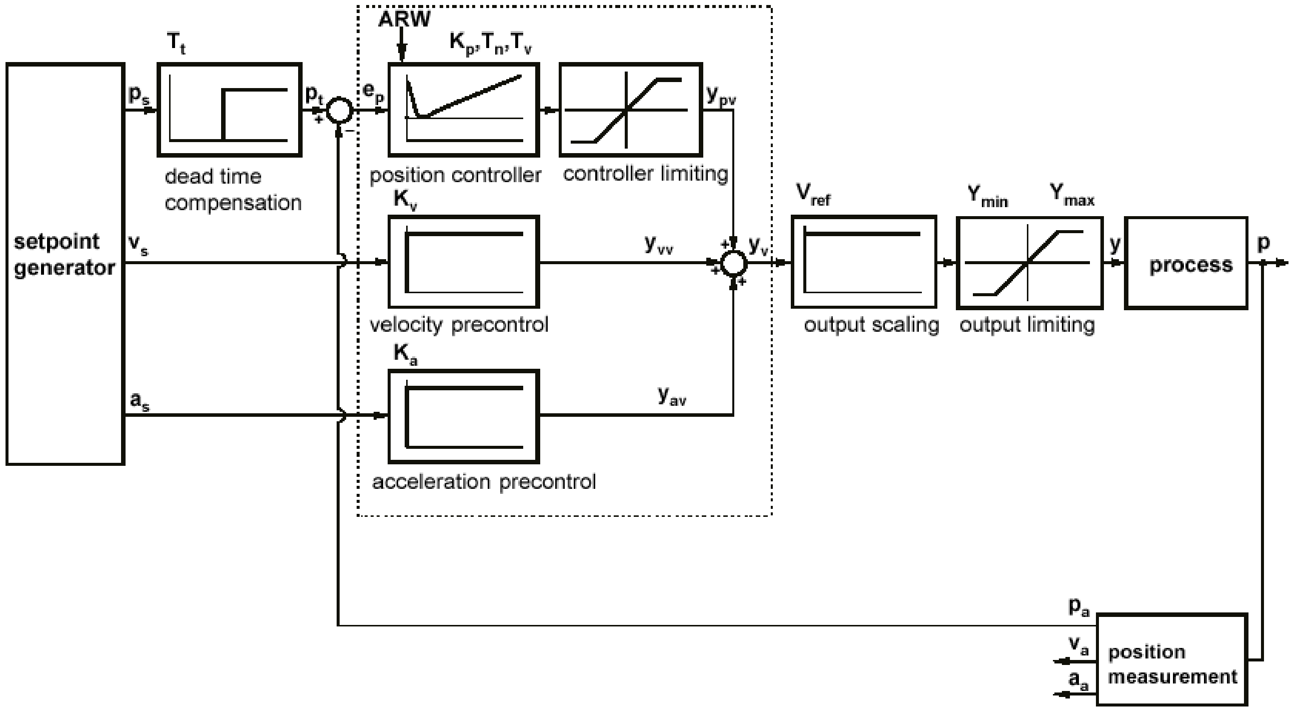
Functionality	Continuous axis (servo axis)	Encoder axis (virtual axis)	High/low speed axis (two speed)	Low cost stepper motor axis (digital I/O's)
TwinCAT PTP (Standard)	√		√	√
TwinCAT PTP (online change of position, endpos, etc.)	√			
TwinCAT Master (possible master for a slave axis)	√	√		
TwinCAT Slave	√			
TwinCAT Camming	√			
TwinCAT Flying Saw	√			
TwinCAT NCI	√			

4.1 Axis control loop

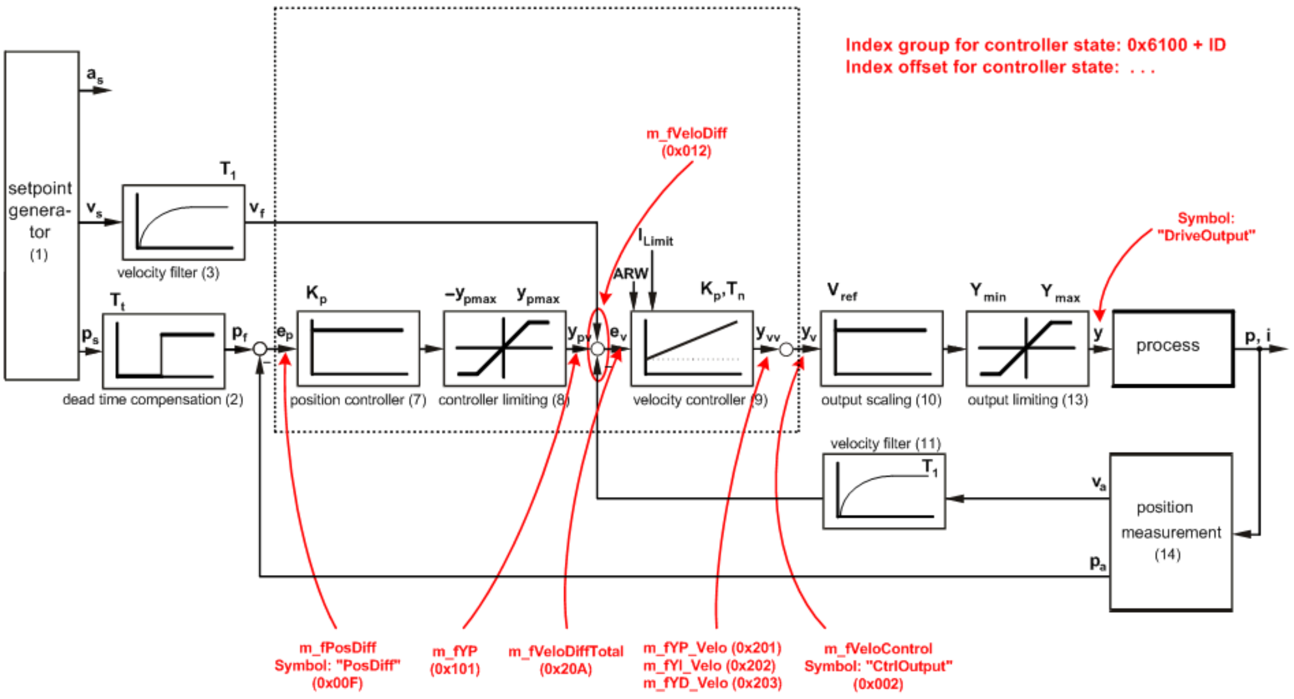
Standard axis control loop (servo drive)



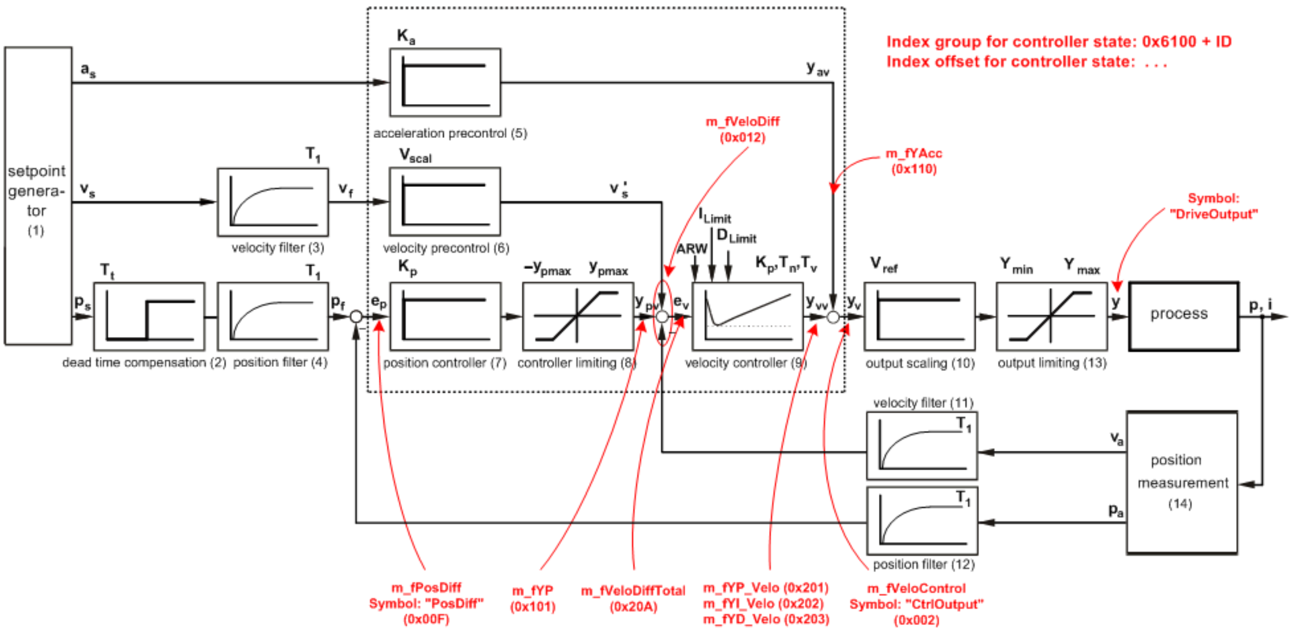
Extended axis control loop (servo drive)



Simplified position and velocity control loop with current/moment interface (servo drive)



Extended position and velocity control loop with current/moment interface (servo drive)



4.2 High/low speed axes

The TwinCAT **high speed/low speed** axis type (two speed) allow the positioning of a high speed/low speed axis. Such a high speed/ low speed axis can be physically made up of a motor with two speeds (switching of the pole pair numbers), or alternatively by a motor that can be driven at two speeds with the help of a frequency converter.

The typical positioning of such an axis is first of all made at high speed up to a parameterization distance away from the target position (*creep distance in pos. or neg. direction).* From this position you switch to low speed, so that the physical speed (actual speed) is reduced to a slower constant speed. The slow speed is then also switched off at a closer distance from the target *(braking distance)* and the stop brake is then activated after a parameterized time *(delay time for brake incidence)*.

This particular positioning sequence is for the sole purpose of ensuring that the axis reaches its target position in the most accurate and repeatable way.

In case a positioning inaccuracy occurs depending on the last physical direction of travel (typical effect of a backlash), then a *loop movement* is activated. This loop movement has the effect that the target position is always approached from the same direction thus reducing the backlash influence. In case of an axis stop basically the same sequence is run through as for positioning without loop movement. However the priority for an axis stop lies on a shorter braking distance or time and not on the positioning accuracy. There is the separate parameter (*loop movement for stop*) for stopping in the shortest possible distance.

General

There are 2 equal possibilities available for the physical control of the axis, in the form of discrete travel signals.

Use of the 6 bits in ControlByte

bMinusHigh	High speed, negative direction
bMinusLow	Low speed, negative direction
bPlusHigh	High speed, positive direction
bPlusLow	Low speed, positive direction
bBreak	Braking bit
bBreakInv	Inverted braking bit

Use of the 6 bits in ExtControlByte

bDirectionMinus	Negative direction
bDirectionPlus	Positive direction
bVeloLow	Low speed
bVeloHigh	High speed
bBreak	Braking bit
bBreakInv	Inverted braking bit

A master-slave coupling is not possible with high/low speed axis.

An axis start will only be initiated if the distance from the target point is in fact larger than the parameterised braking distance.

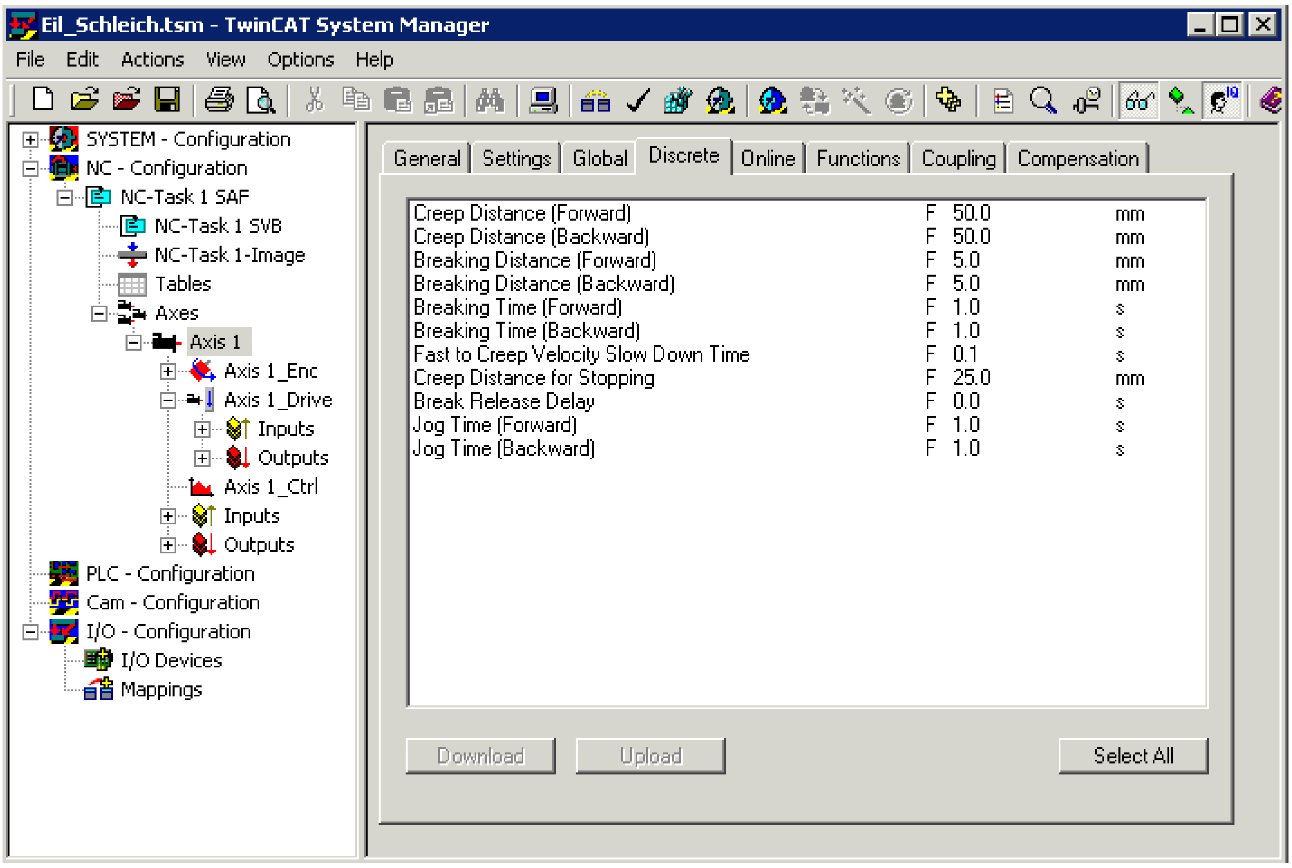
Speed and override

Speed and override [▸ 54]

IO configuration: Drive interface for high/low speed axes NC->IO (12 bytes)

IO configuration: Drive interface for high/low speed axes NC->IO (12 bytes) [▸ 55]

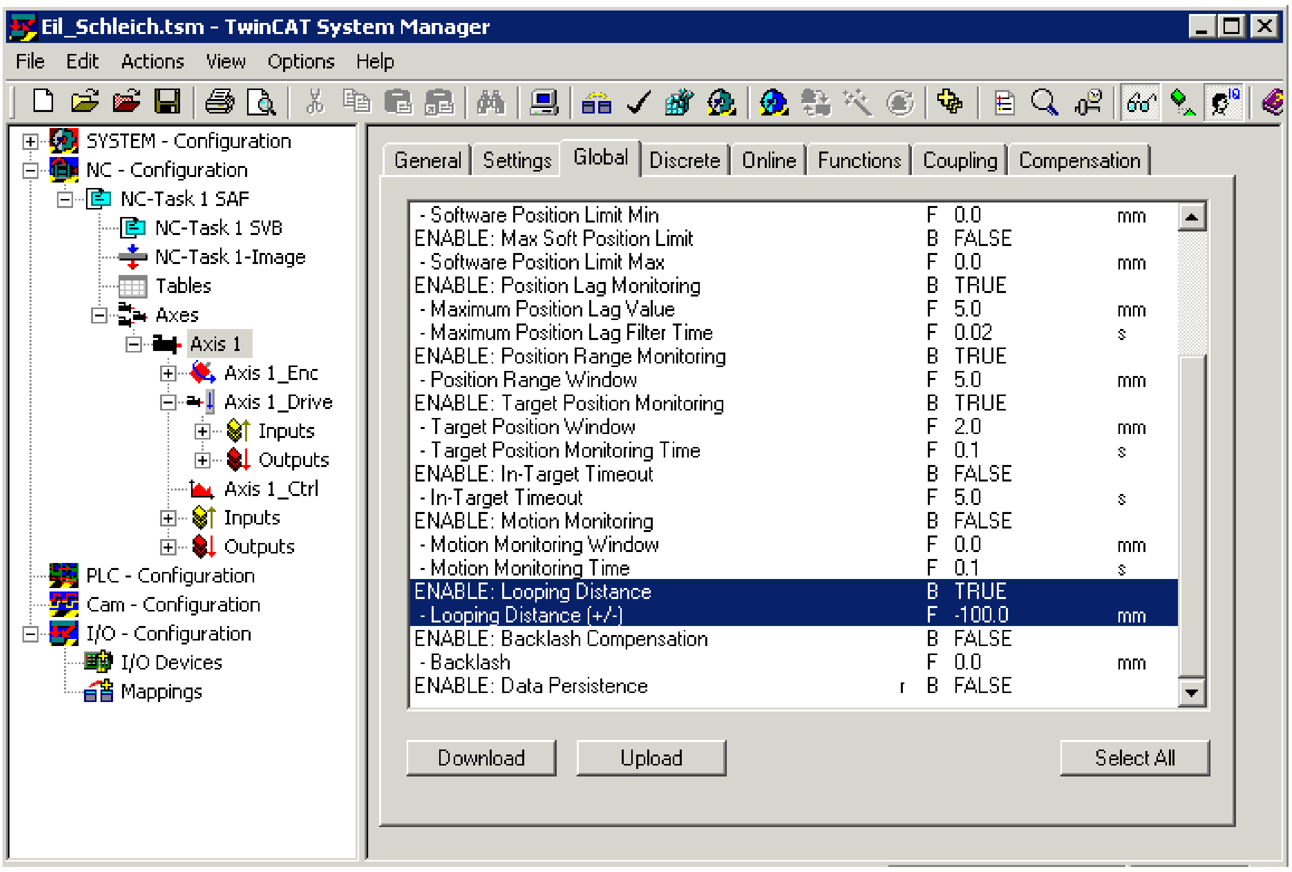
Parameters of the rapid/creep axes



Creep Distance (Forward)	F	50.0	mm
Creep Distance (Backward)	F	50.0	mm
Breaking Distance (Forward)	F	5.0	mm
Breaking Distance (Backward)	F	5.0	mm
Breaking Time (Forward)	F	1.0	s
Breaking Time (Backward)	F	1.0	s
Fast to Creep Velocity Slow Down Time	F	0.1	s
Creep Distance for Stopping	F	25.0	mm
Break Release Delay	F	0.0	s
Jog Time (Forward)	F	1.0	s
Jog Time (Backward)	F	1.0	s

Parameter	Description
Creep distance positive direction	The creep distance in the positive direction gives the distance to the target position, below which the velocity changes from rapid to creep velocity, if the direction of travel is positive. If a looping distance is selected then this distance is based on the movement reversal point. This distance is denoted by **Δp**$_1$ in positioning example 1 [▶ 18].
Creep distance negative direction	The creep distance in the negative direction gives the distance to the target position, below which the velocity changes from rapid to creep velocity, if the direction of travel is negative. If a looping distance is selected then this distance is based on the movement reversal point.
Braking distance positive direction	The braking distance in the positive direction gives the distance to the target position, below which the creep velocity is switched off, if the direction of travel is positive. This distance is denoted by **Δp**$_2$ in positioning example 1 [▶ 18]
Braking distance negative direction	The braking distance in the negative direction gives the distance to the target position, below which the creep velocity is switched off, if the direction of travel is negative.
Brake incidence delay time in pos. direction	This delay time gives the start-up delay of the brakes after switching off of the creep velocity, if the direction of travel is positive. In positioning example 1 [▶ 18], this time is between the times **T**$_4$ and **T**$_5$.
Brake incidence delay time in negative direction	This delay time gives the start-up delay of the brakes after switching off of the creep velocity, if the direction of travel is negative.
Rapid to creep velocity delay time	This waiting time is between the switching off of the rapid velocity and switching on of the creep velocity. In positioning example 1 [▶ 18] , this time is between the times **T**$_2$ and **T**$_3$.

Parameter	Description
Creep distance for stop	The creep distance for stop gives the distance that is covered with the creep velocity after calling up the stop. This creep distance is usually selected shorter than the creep distances in positive and negative direction, since the axis should stop as soon as possible and the exact positioning is not a priority. This distance is denoted by **Δp**$_1$ in positioning example 3 [▸ 20].
Brake release delay	The brake is released immediately the axis is started and after elapse of the brake release delay either the rapid or creep velocity is activated depending on the displacement. In positioning example 1 [▸ 18] , this time is between the times **T**$_0$ and **T**$_1$.
Pulse time in positive direction	This parameter is not evaluated and therefore has no effect.
Pulse time in negative direction	This parameter is not evaluated and therefore has no effect.



```
- Software Position Limit Min          F  0.0      mm
ENABLE: Max Soft Position Limit        B  FALSE
- Software Position Limit Max          F  0.0      mm
ENABLE: Position Lag Monitoring        B  TRUE
- Maximum Position Lag Value           F  5.0      mm
- Maximum Position Lag Filter Time     F  0.02     s
ENABLE: Position Range Monitoring      B  TRUE
- Position Range Window                F  5.0      mm
ENABLE: Target Position Monitoring     B  TRUE
- Target Position Window               F  2.0      mm
- Target Position Monitoring Time      F  0.1      s
ENABLE: In-Target Timeout              B  FALSE
- In-Target Timeout                    F  5.0      s
ENABLE: Motion Monitoring              B  FALSE
- Motion Monitoring Window             F  0.0      mm
- Motion Monitoring Time               F  0.1      s
ENABLE: Looping Distance               B  TRUE
- Looping Distance (+/-)               F  -100.0   mm
ENABLE: Backlash Compensation          B  FALSE
- Backlash                             F  0.0      mm
ENABLE: Data Persistence             r B  FALSE
```

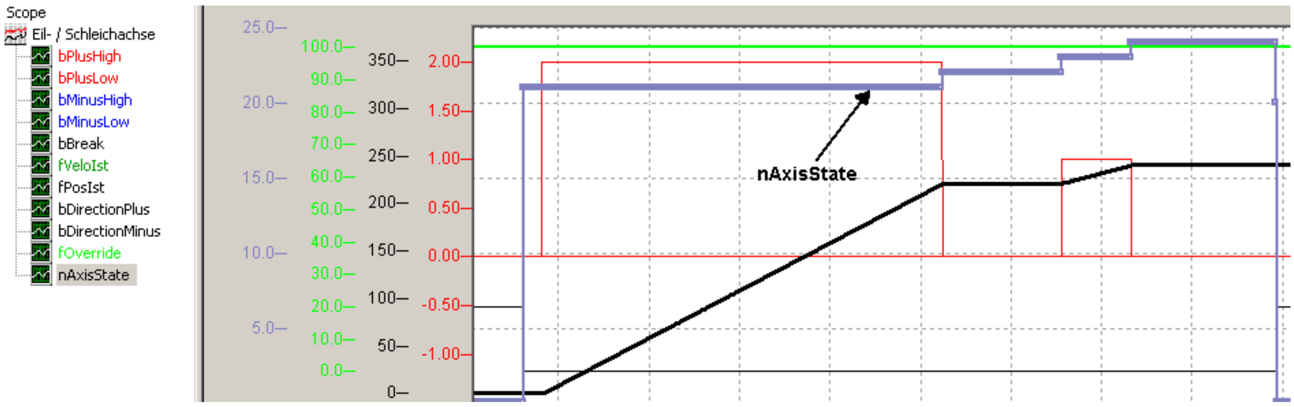
Parameter	Description
OPERATION MODE: looping distance	The looping distance can be activated with this flag. The looping distance is for approaching the target position always from the same direction. In the case of a positive (negative) looping distance a target position in the positive (negative) direction is increased by the amount of the looping distance and the target then approached from the opposite direction. Consequently the target position in case of a positive looping distance is always approached with negative velocity and a negative looping distance with positive velocity.
Looping distance (+ / -)	The looping distance gives the distance, by which the target position is exceeded if necessary so that it is possible to move to the target position from the required direction. This distance is denoted by **creep distance** (looping distance) in positioning example 2 [▸ 19]

Axis movement state (nAxisState in cyclic interface):

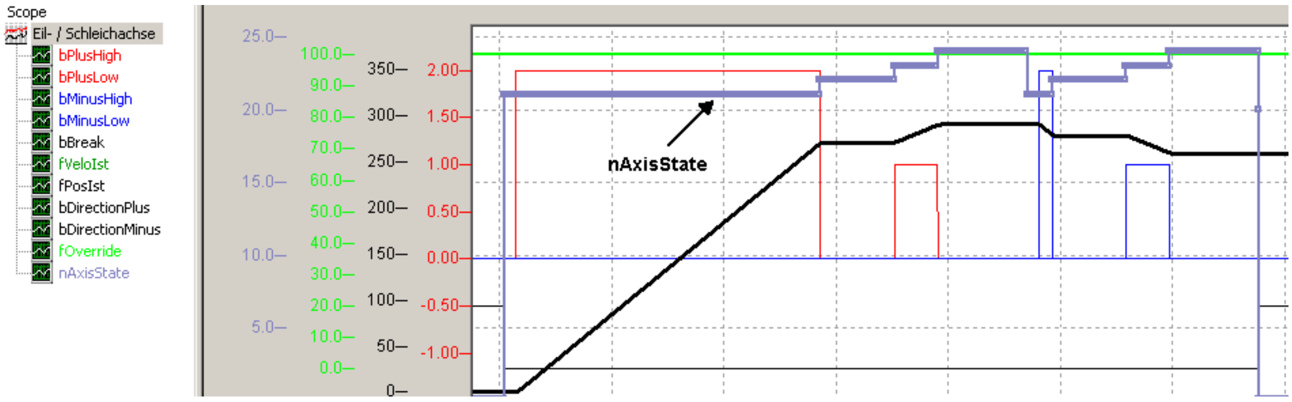
Axis movement state (nAxisState in cyclic interface):

nAxisState	Description
0	Set value generator not active
20	Axis stopped
21	Main travel phase: High or low speed travel in relation to the start speed and override
22	Braking phase: High-to-low speed delay time active
23	Braking phase: Low speed travel
24	Braking phase: Delay time for brake incidence active

1) without looping distance



2) with looping distance



Positioning examples:

Positioning examples

1) Positioning A → B, without loop movement

Positioning A → B, without loop movement

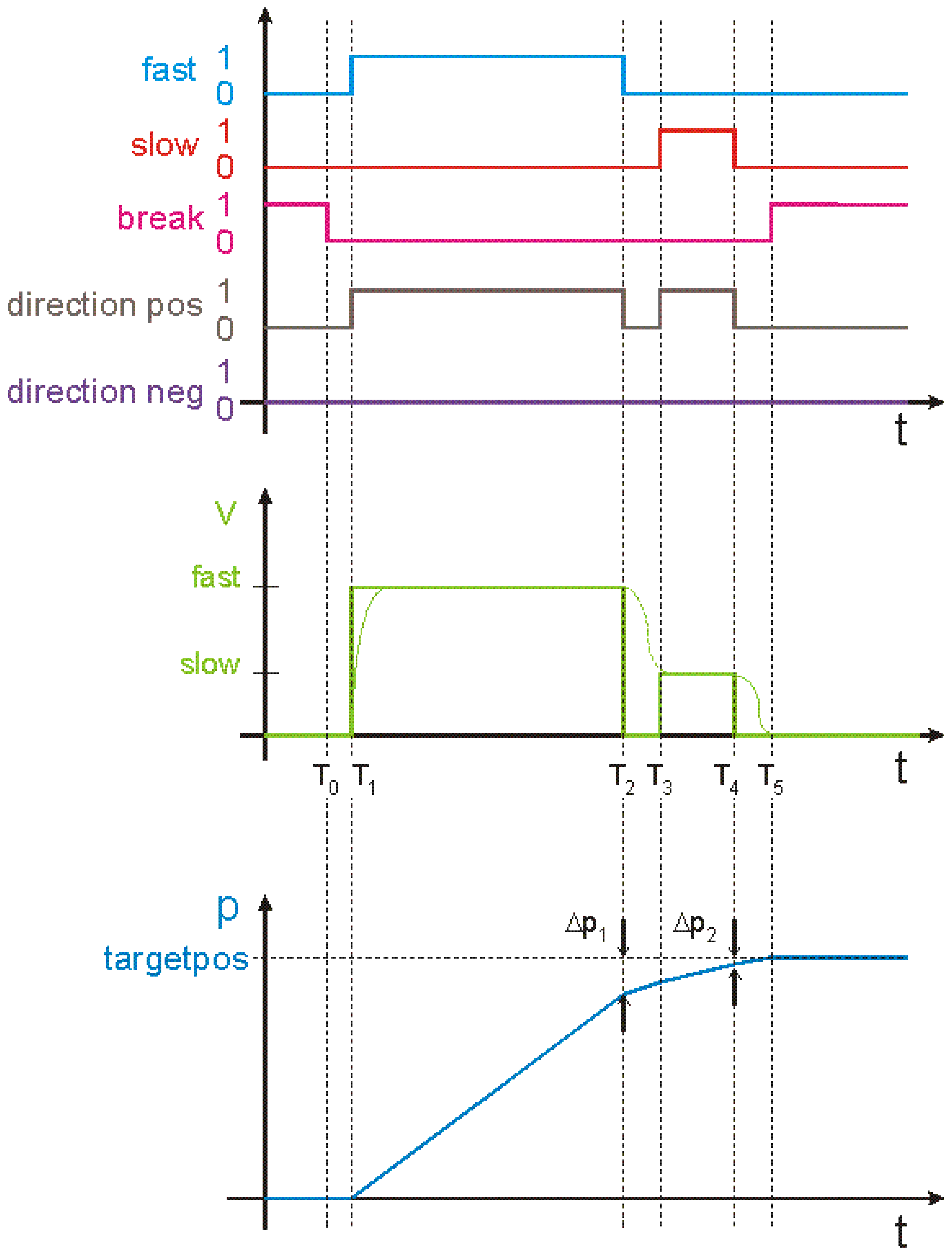


Fig. 1: TcNcTwoSpeed_Positioning1

2) Positioning A → B, with loop movement > 0.0

2) Positioning A → B, with loop movement > 0.0

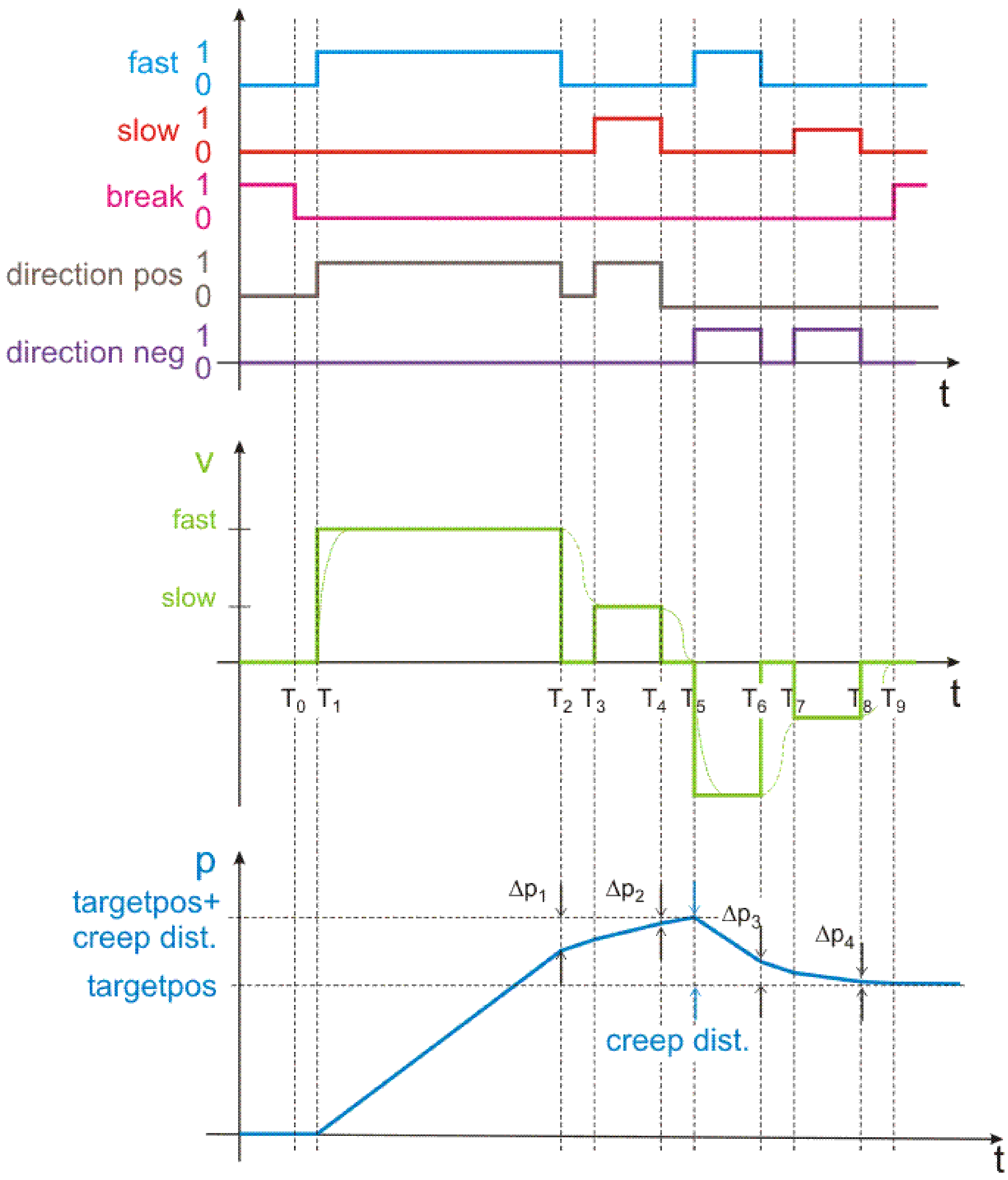
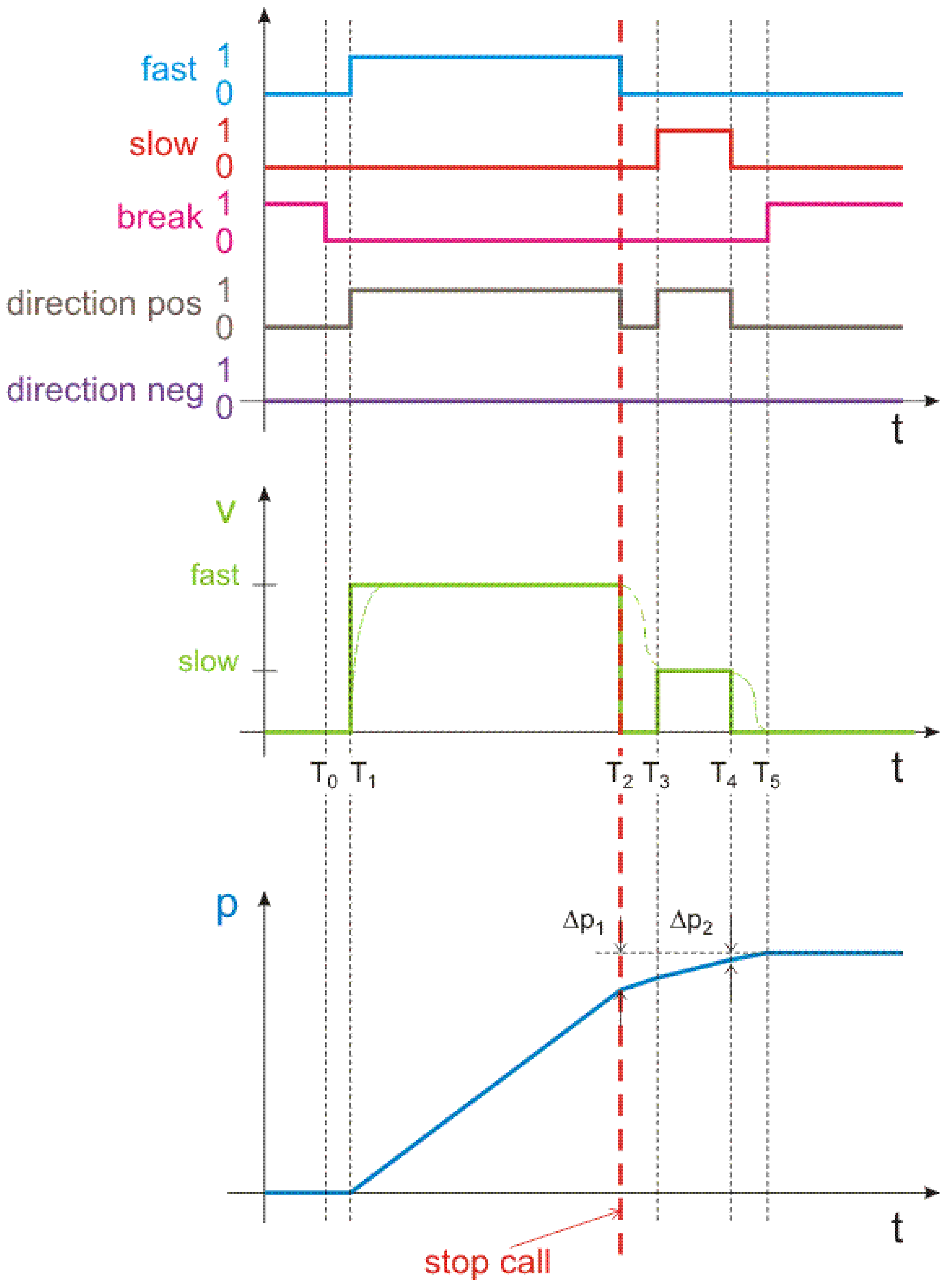


Fig. 2: TcNcTwoSpeed_Positioning2

3) Stop call up in case of active positioning



4.3 Stepper motor axes

The basic version of the 'stepper motor axis' is operated without physical encoder. This means that there is no physical feedback of the actual position. Due to the operating principle of the stepper motor the number of steps made is exactly the same as the pulses output and hence the step counter can be used as a pseudo encoder. When doing so, please be aware that this relationship is lost in case of motor overloading. The diagram below shows the relationship between torque and drift angle.

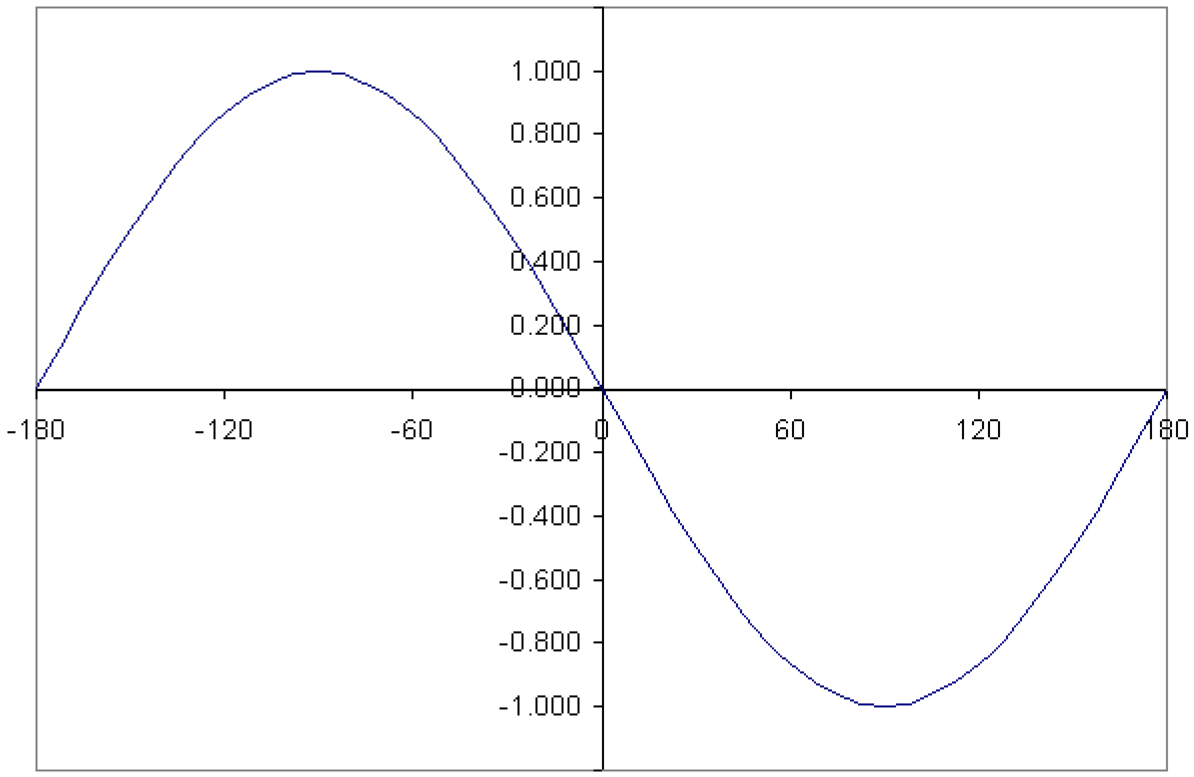


Diagram 1: Dependency of relative torque (M/Mmax) on drift angle.

The angle details relate to a complete pulse cycle. The area displayed represents 3.6 degrees on the motor shaft for a motor with 200 pulses at full stepping operation.

If the motor has to provide torque when stopped or moving, then the rotor is pushed away from the torque-free ideal position by a corresponding angle. This corresponds to the relationship of a non-linear spring in the drive train. Apart from the resulting position deviation of the driven machine part this is unproblematic if the upper and lower vertex are not reached. In this case the motor torque drops, and the remaining load torque continues to push the motor. The motor is then not able to return to the correct angel and runs through a complete polling cycle. Only then is there a chance that the magnetic fields from rotor and stator interlock solidly. However, this only leads to a stable relationship if the load moment now is significantly below the motor torque. In this case the motor has then moved by complete cycle and must be re-referenced. However, this situation is not recognizable without appropriate sensors.

A totally different situation occurs if the load torque does not go down almost immediately due to the polling cycle shift. This is usually the case if the load torque results from the acceleration. Now the motor does not engage at the next stable point with a polling cycle shift, but rather runs through these points. It loses more and more speed in comparison with the distinct stator field and runs through each further pulse cycle without creating a net torque. This means that the motor coasts down due to excessively high noise development. At the end the travel profile runs through an area of decreasing pulse frequencies. Depending on the load moment the motor is rapidly accelerated just before the end of the profile and follow the rest of the profile. Finally, it has not made most of the profile and hence the route. This is also not recognizable without appropriate sensors.

When designing a stepper motor particular attention is to be given to fact that the speed has a strong influence on the available torque. The diagram below shows the relationship for a typical motor.

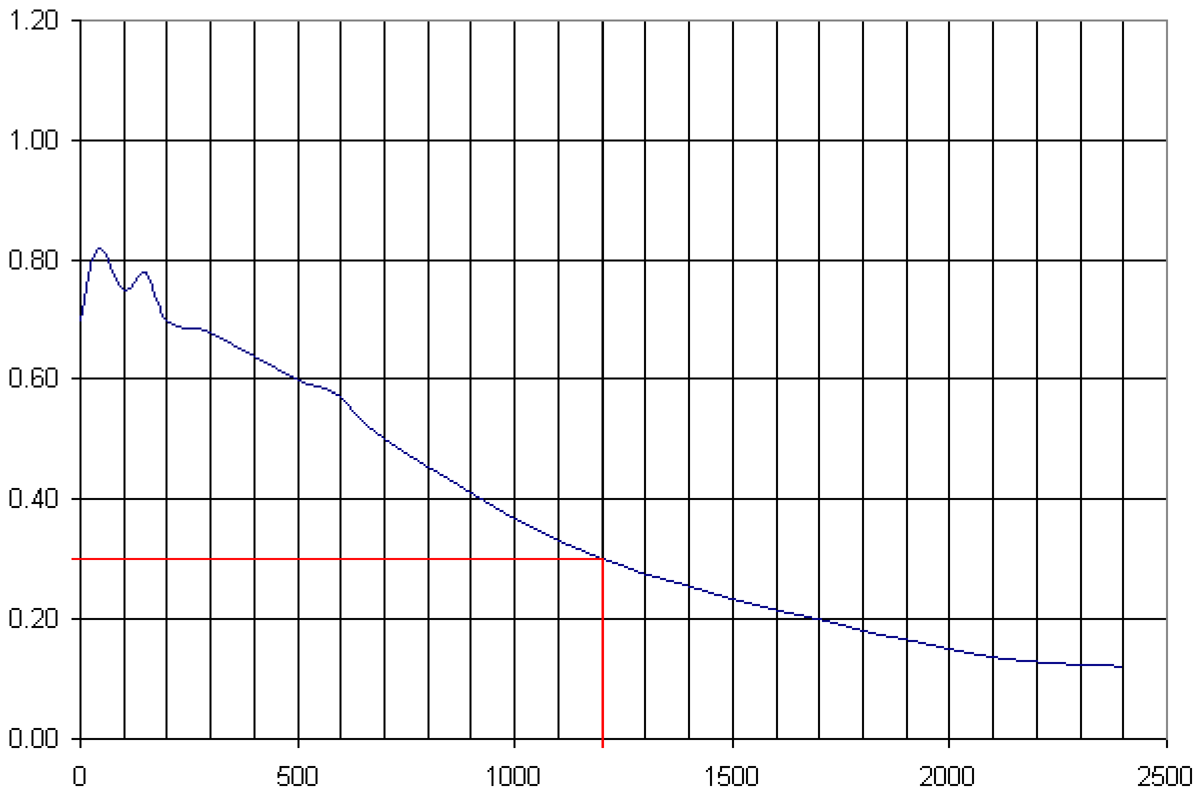


Diagram 2: Dependency of torque (Nm) on speed (RPM) for a typical stepper motor.

Apart from the load torque the motor also has to achieve the torque from the processes that result from the dynamics of the travel profile. If the total torque as shown in the example marked in red is 0.3 Nm then this motor may be run with a max speed of 1200 RPM. In practice however it is essential to plan in a suitable torque reserve.

Configuration of the stepper motor axis

Axis type: 'Continuous axis (incl. SERCOS)'

Encoder type: 'Encoder on KL5051/KL2502-30K/KL2521'

Controller type: e.g. 'position controller P'

Drive type: 'Drive on KL4XXX/KL2502-30K/KL2521'

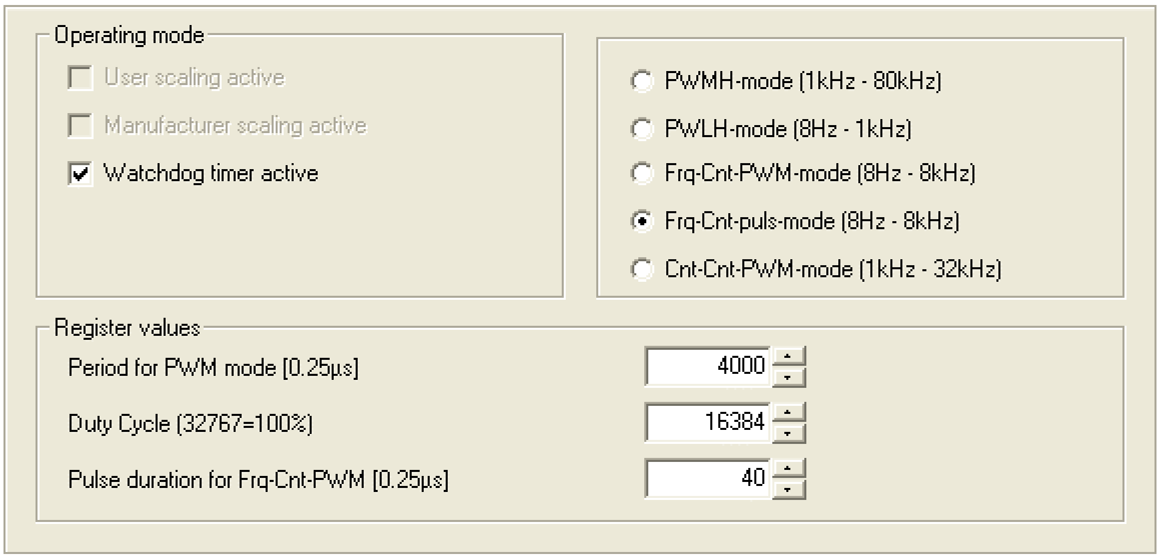
Referencing

As usual, the digital referencing cam signal must be mirrored into the axis interface to the NC (PlcToNc axis structure) via the PLC. The usual procedure for referencing an axis applies, but the hardware feature "latching of a position", initiated by a sync impulse, is not available. Instead, the falling edge is used as local event for determining a reference position during ramp-down from the referencing cam. The accuracy of the referencing process can be increased as required through a reduction in velocity during ramp-down from the cam (the maximum possible precision is one motor step).

In comparison with the axis with analog set value output e.g. via a KL4032 the operation of a stepper motor power section on a KL2502-0010 or -3020 or KL2521 the terminal has to be parameterised depending on the application.

Parameterization of the KL2502-0010 or -3020

These versions of the KL2502 have different hardware from the basic one. The resulting increase of all internal working frequencies makes an expanded work area accessible. However this is not taken into account in the KS2000 software at present.


Operating mode

☐ User scaling active

☐ Manufacturer scaling active

☑ Watchdog timer active

○ PWMH-mode (1kHz - 80kHz)

○ PWLH-mode (8Hz - 1kHz)

○ Frq-Cnt-PWM-mode (8Hz - 8kHz)

● Frq-Cnt-puls-mode (8Hz - 8kHz)

○ Cnt-Cnt-PWM-mode (1kHz - 32kHz)

Register values

Period for PWM mode [0.25µs]	4000
Duty Cycle (32767=100%)	16384
Pulse duration for Frq-Cnt-PWM [0.25µs]	40

Parameterization of the KL2521

The **base frequency 1** is to be set so that the terminal creates the maximum frequency when fully driven for the application concerned. If too low a value is entered here then the axis does not reach the foreseen velocity. If the setting is too high the resolution is worse than it was possible with the high resolution terminal. If this is acceptable then at least limits have to be ensured with by corresponding settings, so that the axis does not travel at too high velocity.

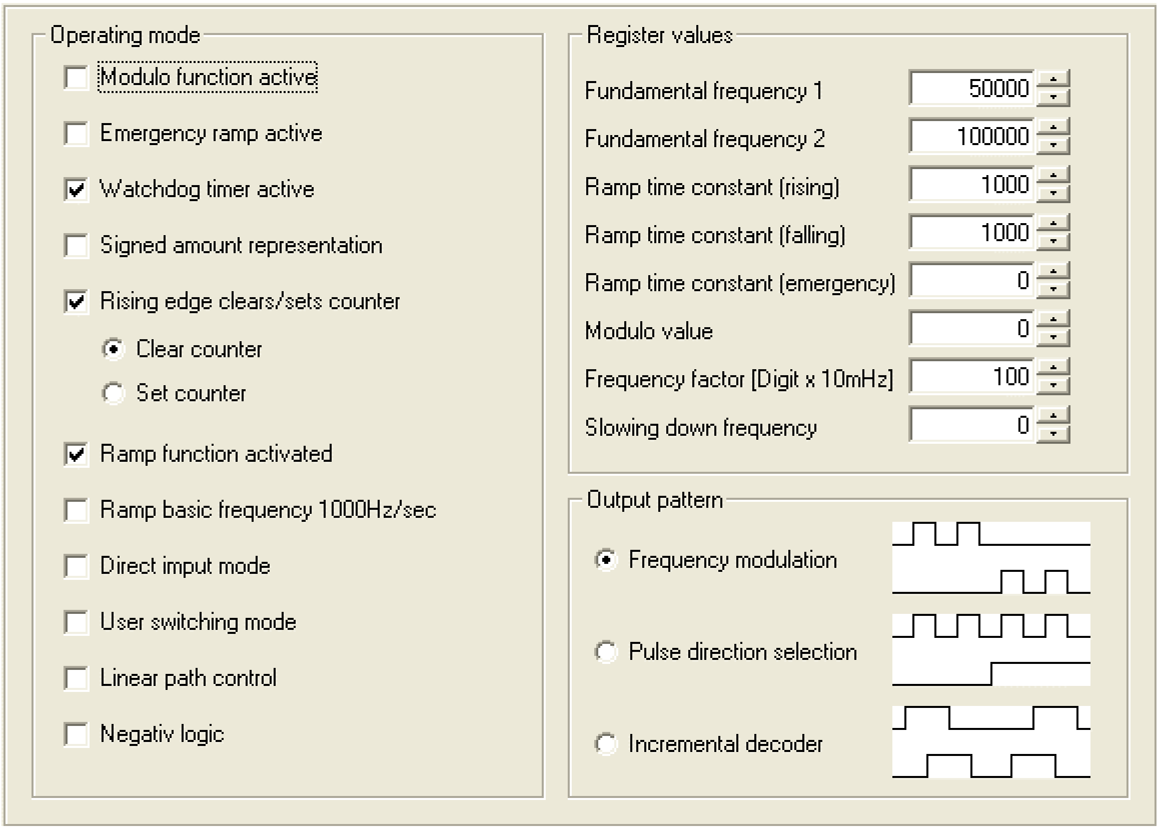
The terminal’s operating mode is to be set as per the signal definition of the power section. In this example, a direction signal and a pulse is expected.

The reference frequency for the example shown in figure 2 is calculated with the formula BF = N * S / 60. Here BF is the frequency in Hertz, N the speed in RPM and S the step number of the motor in the selected operating mode.

Sample calculation:

BF = 1200 [RPM] * 200 [1/R] / 60 [s/min] = 4000 Hz

The following example shows the correct setting for this example.


Operating mode

- [ ] Modulo function active
- [ ] Emergency ramp active
- [x] Watchdog timer active
- [ ] Signed amount representation
- [x] Rising edge clears/sets counter
  - (●) Clear counter
  - ( ) Set counter
- [x] Ramp function activated
- [ ] Ramp basic frequency 1000Hz/sec
- [ ] Direct imput mode
- [ ] User switching mode
- [ ] Linear path control
- [ ] Negativ logic

Register values

Fundamental frequency 1	50000
Fundamental frequency 2	100000
Ramp time constant (rising)	1000
Ramp time constant (falling)	1000
Ramp time constant (emergency)	0
Modulo value	0
Frequency factor (Digit x 10mHz)	100
Slowing down frequency	0

Output pattern

- (●) Frequency modulation
- ( ) Pulse direction selection
- ( ) Incremental decoder

Settings in TwinCAT NC

The scaling factor of the encoder (Sf) is to be calculated from the feed per motor rotation (VU) and the step number (S) of the motor in the selected operating mode:

Sf = VU / S

Use of a KL2521

If a KL2521 is used the reference velocity (Vref) of the drive is to be calculated from the reference frequency (BF) and the scaling factor (Sf) of the encoder. The reference output is to be set to 1.0 when doing so. If the reference frequency of the terminal is set as per the maximum usable frequency then the "maximum allowed velocity" and the "rapid traverse velocity (G0)" in the global axis data can be set to a max of about 90 % of the reference velocity. If a higher reference frequency is selected then a maximum of about 90 % of the velocity may be entered, that would result in case the reference frequency is set exactly.

V_{ref} = BF * Sf

Use of a KL2502-0010 or -3020

If a KL2502-0010 or -3020 is used then the reference velocity of the drive can be entered in two alternative formats. The first one uses an extrapolation of the output frequency on the full value range of the setpoint, even if this frequency cannot be displayed by the terminal. Here the S shows the step number of the motor in the selected operating mode. Here the reference frequency (BF) is the product of the maximum output value and the frequency scaling and results in 32768*8=262144.

V_{ref} = BF * S = 262144 * S

Alternatively the reference velocity of the drive at the max selected velocity can be specified. For this the max corresponding frequency (Fmax) and the step number (S) of the motor are to be calculated in the selected operating mode.

V_{ref} = Fmax * S

The associated relative control factor (A) is to be given as the reference output.

A = F_{max} / BF = F_{max} / 262144

All other axis parameters are to be set in the same way as for a servo axis. Here in particular empirical tests for the setting of the acceleration, deceleration and max jerk are to be carried out with the maximum expected mass to be moved in operation. Here it may not happen that the axis leaves out part of the movement due to overload.

Notice **Stepper motors have a more or less distinctive natural resonance. This can lead to problems in the case of very flat ramps or unfavorably selected target velocities. The axis will then generate significant noise and can then be shifted and lose its correct reference by one or more polling cycles due to superposition of the travel movement with a low frequency jitter movement caused by brief overloading.**

4.4 "Low Cost" stepper motor axes with digital control (24V / 2A)

The basic version of the 'low cost stepper motor axis' is operated without physical encoder (hence simulation encoder). This means that there is no actual physical feedback between set values and actual values, and therefore the axis is not operated with closed-loop control, but only with open-loop control. It is assumed that there is no slippage in the drive, and that the axis can accurately follow the specified step pattern (set value profile). Notwithstanding this constraint imposed for cost reasons, the stepper motor axis can be referenced physically, since the simulation encoder is implemented as an incremental encoder and supports virtually all features.

Configuration of the "low cost" stepper motor axis

Axis type: 'low cost stepper motor axis' (dig. I/O)'

Encoder type: 'simulation encoder'

Controller type: 'stepper motor controller'

Drive type: 'stepper motor drive'

Referencing

As usual, the digital referencing cam signal has to be mirrored into the axis interface to the NC (PlcToNc axis structure) via the PLC. The usual procedure for referencing an axis applies, but the hardware feature "latching of a position", initiated by a sync impulse, is not available. Instead, the falling edge is used as local event for determining a reference position during ramp-down from the referencing cam. The accuracy of the referencing process can be increased as required through a reduction in velocity during ramp-down from the cam (the maximum possible precision is one motor step).

Commissioning instructions

At this point, it is worth mentioning some technical restrictions of stepper motors. Note that all recommended tests have to be carried out under load.

- The maximum permitted limit frequency $f_{v_{max}}$ [kHz] of the stepper motor in motion must not be exceeded, since otherwise the linear range of the drive would be exceeded and the drive limit approached.
- The maximum start and stop frequency $f_{start/stop_{max}}$ [kHz] during starting and stopping of the axis must not be exceeded, since otherwise the stepper motor could "lose steps" during this positioning start and end phase.
- The maximum acceleration and deceleration $\frac{a_{max}}{d_{max}}$ [kHz/s] must not be exceeded during the start and stop phase of the motion, since otherwise steps could be "lost" here too.

- In the event of the stepper motor not being able to cope, the contact to the driving electromagnetic field is interrupted, and the motor comes to a halt (with a chaotic jitter). For commissioning, the motor should therefore be brought up to the desired set velocity (1st test parameter) with slowly and constantly increasing velocity during a first test series with slight ramps (low acceleration and deceleration), in order to ensure that the motor can follow. In a second test series (with fixed set velocity) the ramps should be increased slowly and constantly to the desired acceleration/deceleration (2nd test parameter).

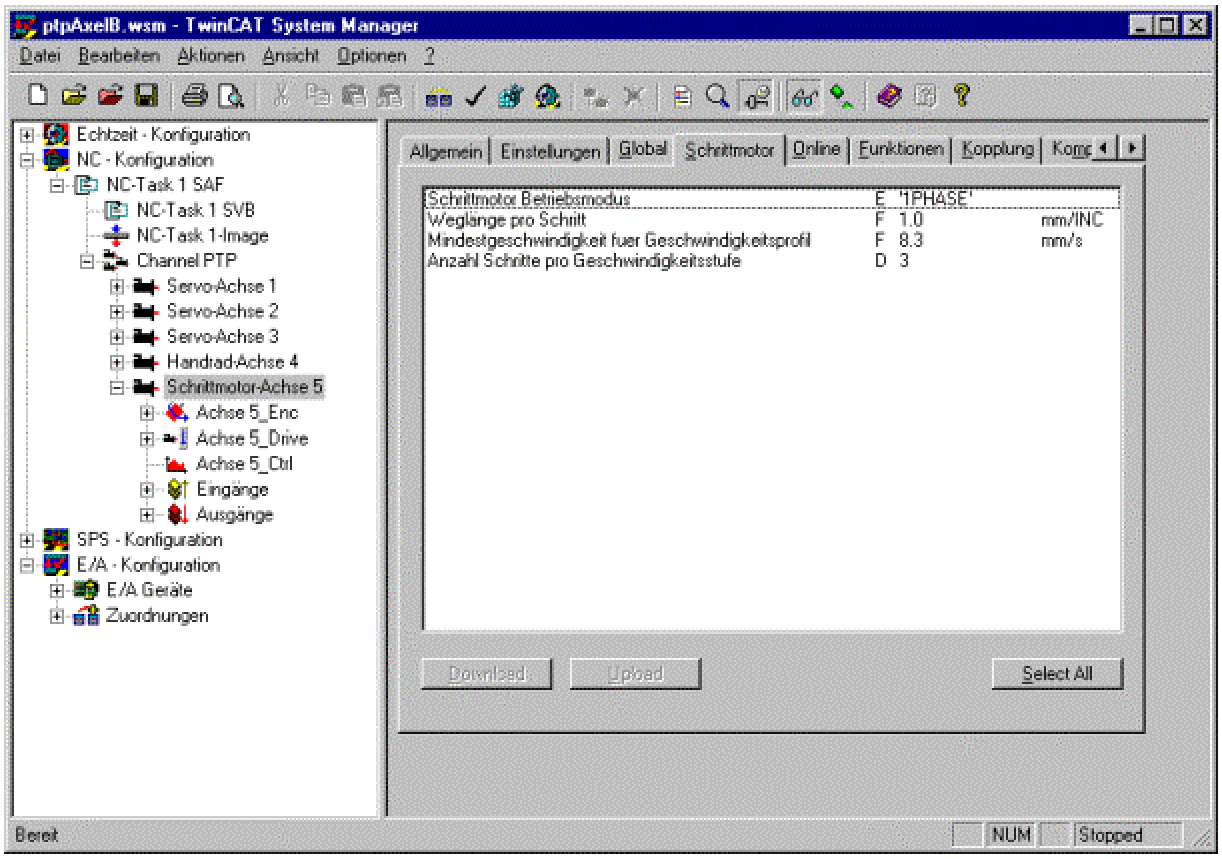
IO configuration of the drive

The link (mapping) of the logical NC axis outputs (drive output structure, 'nCtrl' byte or 'nExtCtrl' byte) with the physical digital IO outputs (24V / 2A) still has to be carried out manually in the TwinCAT System Manager. Individual bits from the 'nCtrl byte' or 'nExtCtrl byte' are linked with the respective digital outputs. In order to select an individual bit from a byte (8 bit), a byte offset in the range between 0 and 7 should be entered. If, for example, the second bit of a byte is to be addressed, a value of 1 has to be entered as byte offset.

Configuration parameters

	Data type	Byte	Bit	Def.-Range	Variable name	Description
1	INT32	0 – 3	__	__	nOutData1	Drive output data 1 (NC→I/O)
2	INT32	4 – 7	__	__	nOutData2	Drive output data 2 (NC→I/O)
3	UINT8	8	__	__	nControlByte	Control byte
3.0	BOOL	8	0	0 / 1	bPhaseA	Phase A
3.1	BOOL	8	1	0 / 1	bPhaseAInv	Phase A inverse
3.2	BOOL	8	2	0 / 1	bPhaseB	Phase B
3.3	BOOL	8	3	0 / 1	bPhaseBInv	Phase B inverse
3.4	BOOL	8	4	0 / 1	__	RESERVE
3.5	BOOL	8	5	0 / 1	__	RESERVE
3.6	BOOL	8	6	0 / 1	bBreakInv	Inverse braking bit: (*0* = ACTIVE, *1* = PASSIVE)
3.7	BOOL	8	7	0 / 1	bBreak	Braking bit (*0* = PASSIVE, *1* = ACTIVE)
4	UINT8	9	__	__	nExtControlByte	Extended control byte
4.0	BOOL	9	0	0 / 1	bFrequency	Frequency (square wave signal)
4.1	BOOL	9	1	0 / 1	bDirectionPlus	Direction positive
4.2	BOOL	9	2	0 / 1	__	RESERVE
4.3	BOOL	9	3	0 / 1	__	RESERVE
4.4	BOOL	9	4	0 / 1	__	RESERVE
4.5	BOOL	9	5	0 / 1	__	RESERVE
4.6	BOOL	9	6	0 / 1	__	RESERVE
4.7	BOOL	9	7	0 / 1	__	RESERVE
5	UINT16	10 -11	__	__	nReserved	Reserved bytes

Stepper motor parameters



```
Schrittmotor Betriebsmodus                        E  '1PHASE'
Weglänge pro Schritt                              F  1.0          mm/INC
Mindestgeschwindigkeit fuer Geschwindigkeitsprofil F  8.3          mm/s
Anzahl Schritte pro Geschwindigkeitsstufe         D  3
```

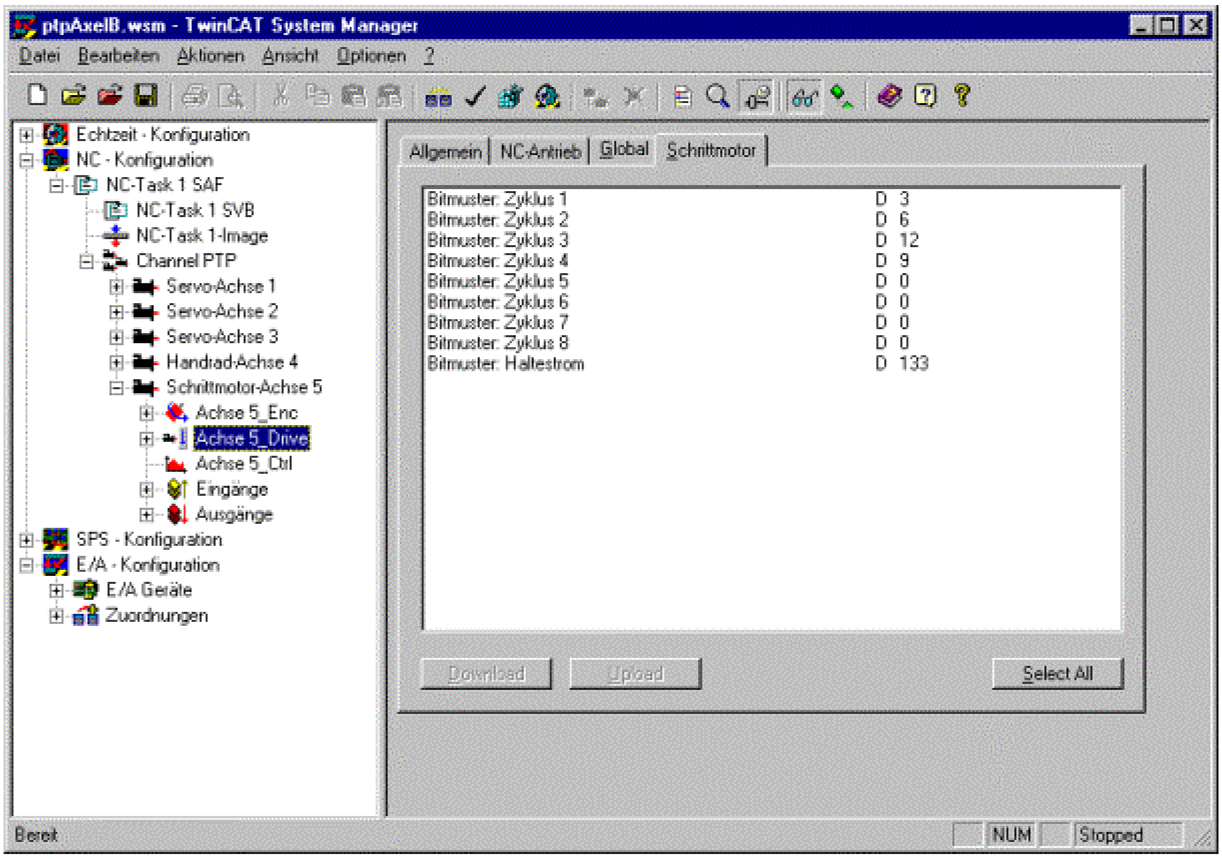
Table 1: Motor parameters

Parameters	**Description**
Stepper motor operating mode	The stepper motor can be operated in different operating modes through the type of control used. The period at which various step pattern combinations are output can thus be set. The step size (full step or half step) and the motor torque can thus be influenced. The available operating modes are described in the "stepper motor operating mode" table below.
Displacement per step dS_{Step}	Physical scaling of a motor step (step change) according to the mechanical conditions such as gearbox, etc. Unit: [*mm/INC*]
Minimum velocity for velocity profile V_{min}	For generating the velocity profile, this minimum velocity Vmin is used as the start and end velocity of the calculated velocity profile, provided the required set velocity can be reached on the travel path. Obviously, due to the time discretisation (cycle time) not every velocity level can be reached. Only a certain number of discrete velocity levels is available (proportional to *1/(n*dT)* with *n=1,2,3...* and *dT* axis cycle time in seconds). The value is rounded, and the next possible velocity level is used. Unit: [*mm/s*]
Number of steps per velocity level >N_{level}	According to the boundary conditions (minimum velocity, target velocity, travel path), a velocity profile with discrete velocity levels (proportional to the step frequency) is calculated. Maintaining of a certain velocity level in the range between *0* to *100* relative to the motor steps can be set via this parameter. The parameter Nlevel can therefore be used for scaling (extending or reducing) the velocity ramp. For the limit case of *Nlevel = 0*, no ramped velocity profile is generated, but the required set velocity is kept constant for the entire travel path. Unit: *1*

Stepper motor operating modes

Table 2: Motor modes

Stepper motor mode	Description
1 PHASE	1-phase excitation (step pattern periodicity: modulo 4) (STANDARD)
2 PHASE	2-phase excitation (step pattern periodicity: modulo 4)
12 PHASE	1-2-phase excitation (step pattern periodicity: modulo 8)
DRIVER	Control via power section (specification of direction and frequency



Bitmuster: Zyklus 1	D	3
Bitmuster: Zyklus 2	D	6
Bitmuster: Zyklus 3	D	12
Bitmuster: Zyklus 4	D	9
Bitmuster: Zyklus 5	D	0
Bitmuster: Zyklus 6	D	0
Bitmuster: Zyklus 7	D	0
Bitmuster: Zyklus 8	D	0
Bitmuster: Haltestrom	D	133

Formula for calculating the discrete travel frequencies

$$f_z(n) = f_g \cdot \frac{1}{n} = \frac{1}{n \cdot T}$$

$n:$ *Subteiler für Grundfrequenz* $n \in [1,2,3,\ldots,100]$

$T:$ *Grundzykluszeit z. B. 2 ms* $[s]$

$f_g:$ *Grundfrequenz z. B. 1000 Hz* $[1/s]$

$f_z(n):$ *Diskrete Schrittmotorfrequenz* $[1/s]$

Formula for calculating the discrete velocity levels

$$v_z(n) = \frac{s}{n \cdot T}$$

$s:$ *Skalierung eines Motorschrittes* $[mm/INC]$

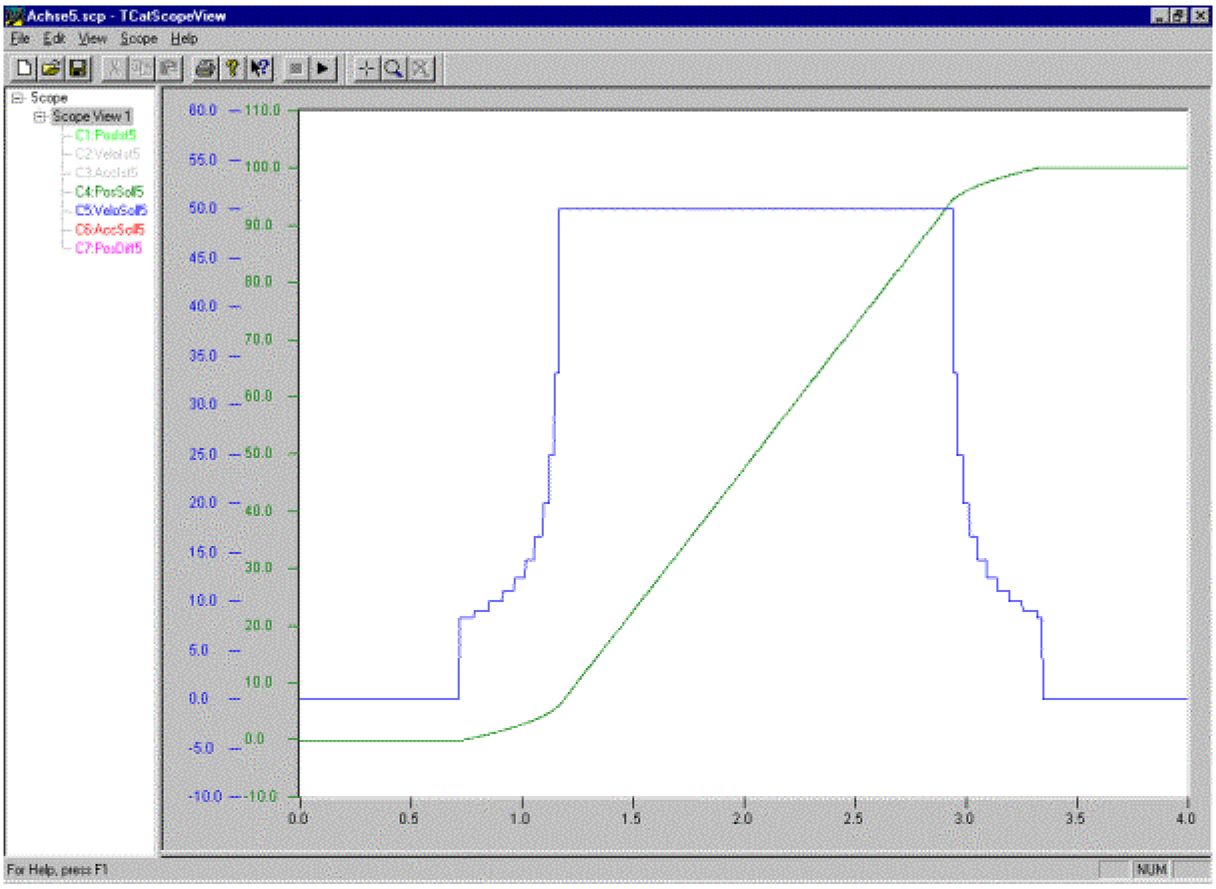
$v_z(n):$ *Diskrete Geschwindigkeit* $[mm/s]$

Stepper motor parameters of the drive

Depending on the type and operating mode of the stepper motor, up to 8 byte masks for the drive output pattern can be parameterized in the "stepper motor" drive tab of the System Manager. Depending on the operating mode of the axis, these patterns are cyclically output during active positioning and are repeated periodically. With the axis at standstill (no active travel), the last bit pattern to be output is linked to the holding current mask as a logical AND operation. This holding current mask is intended to reduce the motor current and therefore the motor temperature of the axis at standstill.

Example of velocity and position profile

The graph shows a typical velocity and position profile for a 'low cost' stepper motor axis vs. time [s]. The blue curve shows the velocity [mm/s], and the green curve the position [mm]. The set values and actual values are identical, because a simulation encoder was used, not a real encoder. The discrete velocity levels are noteworthy. Their value, starting from the start velocity, rises to the required set velocity, and falls again to the stop level towards the end of the target position.



5 TwinCAT NC Axis components

Each axis consists of various elements, depending on the axis type. These elements include the encoder, the drive, the controller and the set value generator, along with the PLC interface. There are also more complex axes consisting of several encoders, switchable controllers or changing set-value generators.

Encoders

Depending on their operating modes, encoders [▸ 31] determine the actual position, actual velocity or the actual acceleration. Actual values often fluctuate heavily, so a parameterizable filter is available for each mode to ensure a reasonable resolution. A wide variety of encoders versions is supported. Both absolute and incremental encoders are available. On top of this, there are simulation encoders and special encoders for the determination of force. The encoder parameters include the scaling, zero offset shift and the modulo factor. In addition, the encoder data includes the parameters for the software end locations and for the reference travel.

Drive

The drive [▸ 34] transfers the output voltage to the motor's power section. A wide variety of drive versions is supported. Servo drives, a high/low speed drives, and stepper motor drives are available. The drive parameters include the motor polarity and the reference speed.

Controllers

The purpose of the controller [▸ 36] is to operate based on the set velocity differences (following error) or other set magnitudes (acceleration) in such a way that the following error is kept as small as possible and that the axis does not undergo any overshoots in position or velocity. A wide variety of controller versions is supported. Servo position controllers and special controllers for particular types of axis are available.

Set value generators

Every axis is assigned to a set value generator [▸ 40], and this in turn consists of three components:

- Block preparation generator (in the block preparation task): checking the start parameters and, in the case of master axes, calculation of the dynamic profile.
- Block execution generator (in the block execution task): calculation of the local set values.
- Asynchronous generator for reaction to asynchronous requirements (override, new end position, position compensation etc.).

Input/Output

There are input/output [▸ 41] connections for channels, axes, encoders and drives.

5.1 Encoder

Depending on their operating modes, encoders determine the actual position, actual velocity or the actual acceleration. Actual values often fluctuate heavily, so a parameterisable filter is available for each mode in order to ensure a reasonable resolution. A wide variety of encoders versions is supported. Both absolute and incremental encoders are available. On top of this, there are simulation encoders and special encoders for the determination of force. The encoder parameters include the scaling, zero offset shift and the modulo factor. In addition, the encoder data includes the parameters for the software end locations and for the reference travel.

Interface

Every encoder that is not of the "simulation" type must be associated with an actual value acquisition module [▸ 41].

Encoder types

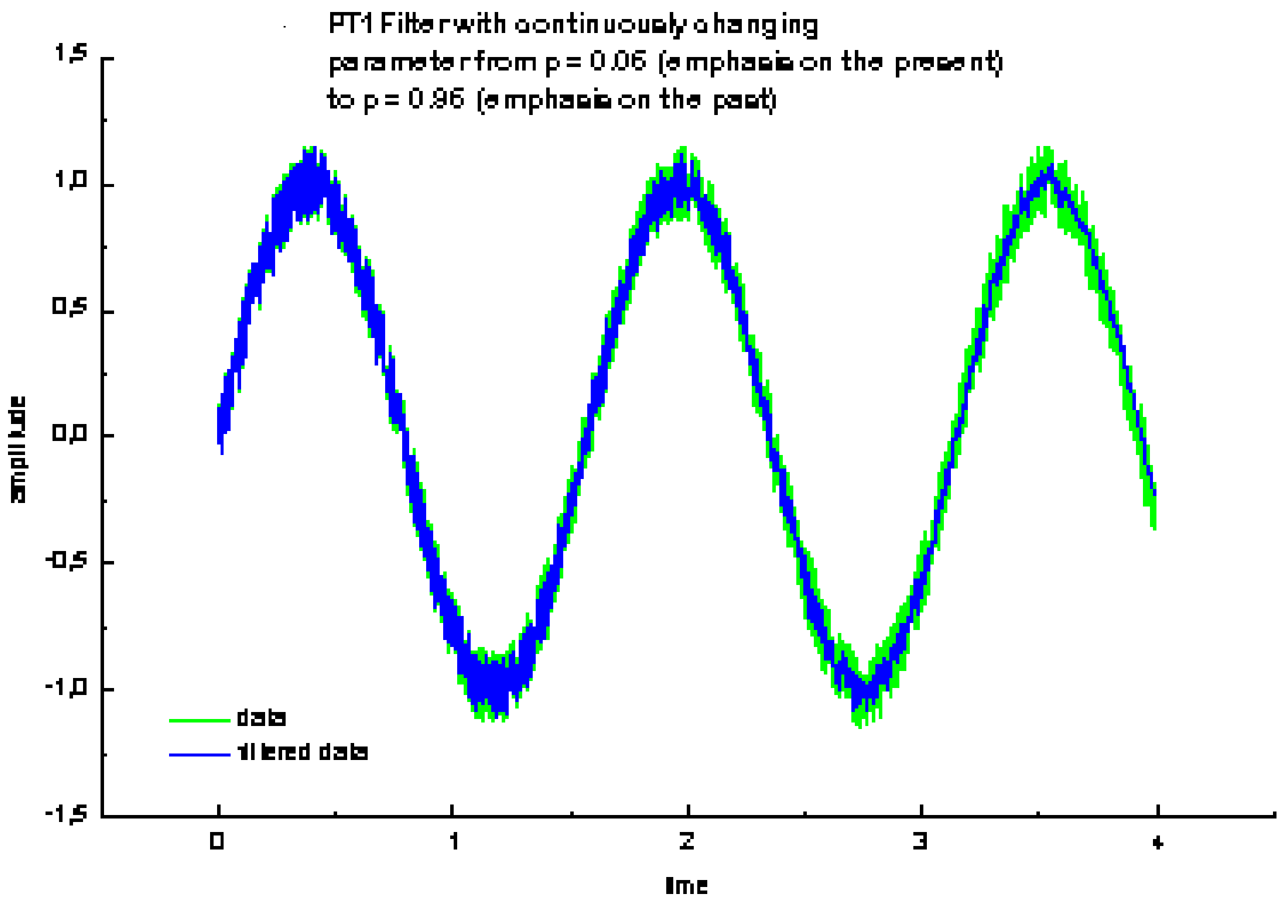
1. Simulation encoders
2. Absolute, with 24 or 25 bits, and 12 and 13 bit single turn encoders (M3000)
3. Incremental, with 24 bits (M31x0, M3100, M2000)
4. Incremental, with 16 bits (KL5101)
5. Absolute SSI with 24 bits (KL5001)
6. Absolute/incremental BISSI with 16 bits (KL5051 and PWM Terminal KL2502_30K (frq. cnt. pulse mode))
7. Absolute analog input with 16 bits (KL30xx)
8. SERCOS "Encoder" POS
9. SERCOS "Encoder" POS and VELO
10. Binary incremental encoders (0/1)
11. Absolute analog input with 12 bits (KL2510)
12. T&R Fox 50 module (24 bits absolute (SSI))
13. Force acquisition from Pa, Pb, Aa, Ab
14. Incremental with 16/20 bits (AX2000)
15. Incremental with 32 Bit
16. Incremental with variable bit mask (max. 32 bits)
17. Incremental NC backplane

Encoder parameters

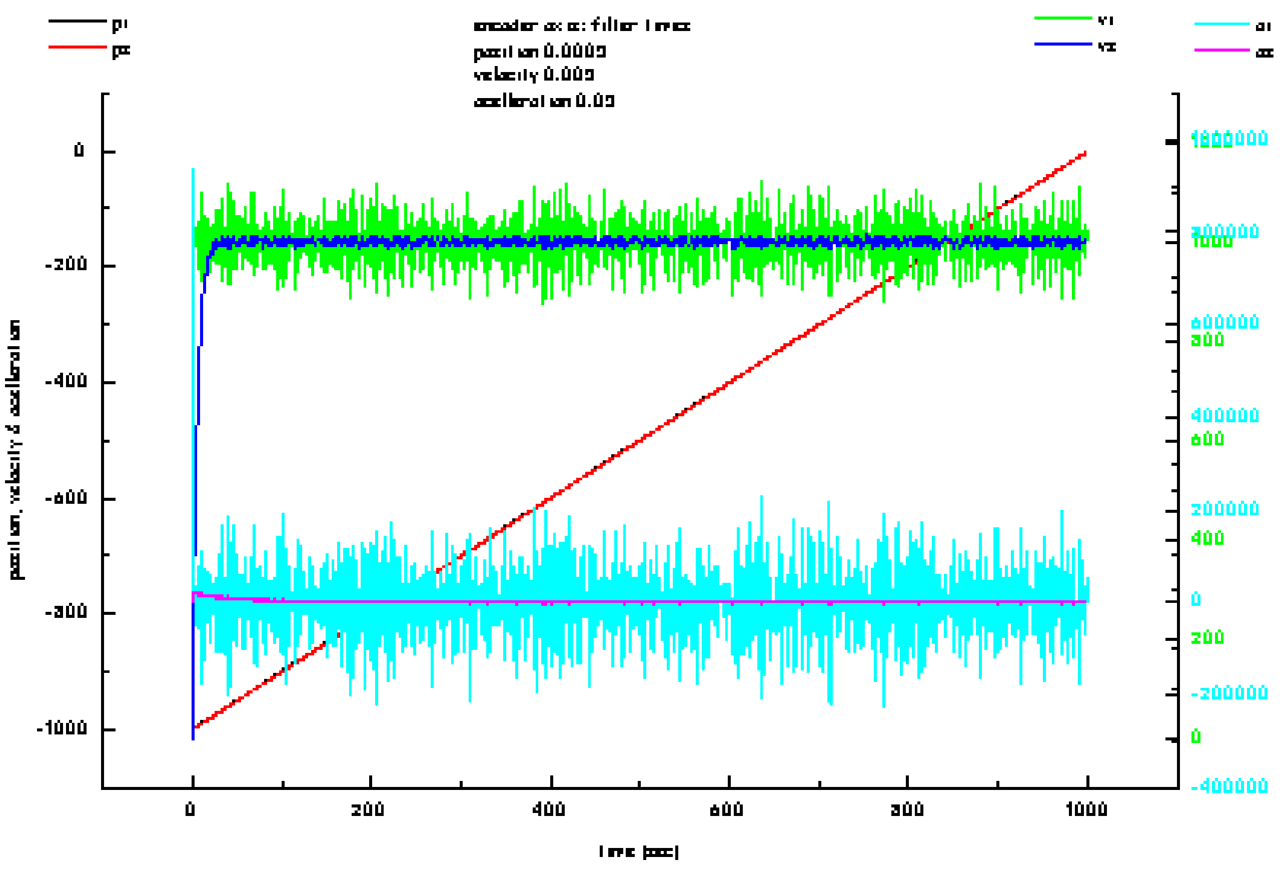
Parameter	Data type	Unit	Description
ENCODER mode	ENUM		Operation mode of the encoder: POS = the actual position is determined. POSVELO = the actual position and the actual velocity are determined. POSVELOACC = the actual position, actual velocity and the actual acceleration are determined.
Encoder counting direction inverted	BOOL		Counting direction inverted: FALSE = the polarity of the axis movement agrees with the counting direction of the acquisition hardware. TRUE = the polarity of the axis movement is in the opposite sense to that of the counting direction of the acquisition hardware.
Scaling factor	float	mm/INC	Incremental evaluation: this factor is used to convert the displacement increments into axis positions.
Zero offset shift	float	mm	With absolute encoders: this value is added to the encoder position in order to determine the axis position. It is used to specify the machine-dependent zero point.
Modulo factor	float	mm	With rotary axes: this is the "distance" represented by one rotation. If the actual value is acquired, for instance, in degrees, 360.0 should be entered here.
OPERATING MODE: min. end position monitoring	BOOL		Activation and deactivation of the monitoring of the minimum end position.
Software end position, min.	float	mm	Location of the minimum end position
OPERATING MODE: max. end position monitoring	BOOL		Activation and deactivation of the monitoring of the maximum end position.
Software end position, max.	float	mm	Location of the maximum end position
Actual position filter time	float>0	s	Filter time for the PT1 filtering of the actual position.
Actual velocity filter time	float>0	s	Filter time for the PT1 filtering of the actual velocity.

Parameter	Data type	Unit	Description
Actual acceleration filter time	float>0	s	Filter time for the PT1 filtering of the actual acceleration.

PT1 filter



A PT1 filter is a transfer function that performs convex interpolation between a new value x_n and an old value (from one cycle ago) x_a. The filter time parameter (>= 0.0) in seconds is to be entered. If l=Saf cycle time/(Saf cycle time+filter time), then x = l x_n + (1-l) x_a, l Î [0.0,1.0]. If the filter time is close to 0.0, the new value has a high weighting. If the filter time is long, the older value has relatively high weighting.



Referencing parameters

Parameter	Data type	Unit	Description
Search direction for referencing cam inverted	BOOL		Search direction inverted: FALSE = cam is looked for in the direction of positive movement. TRUE = cam is looked for in the direction of negative movement.
Search direction for sync pulse inverted	BOOL		Search direction inverted: FALSE = synchronization pulse is looked for in the direction of positive movement. TRUE = synchronization pulse is looked for in the direction of negative movement.
Reference position	float	mm	The axis position that is assigned to the synchronization pulse that is active during homing. If the actual position of the axis is acquired in any unit other than mm, then that unit must be used here too.
External sync pulse	BOOL		Reserved

SERCOS Axes

The modulo scaling (only in the weighting of the position data in the modulo display (S-0-0076 bit 7)) determines the modulo value at which the position data begins again at 0. This value must agree with the value set at the drive (S-0-0103) if the position is to be correctly indicated and controlled.

When TwinCAT is active and SERCOS is at least in Phase 2, the value can be read off via "Calculate". Download and Upload can be used to write or read the value to and from the running TwinCAT system.

5.2 Drive

The drive transfers the output voltage to the motor's power section. A wide variety of drive versions is supported. Servo drives, a high/low speed drives, and stepper motor drives are available. The drive parameters include the motor polarity and the reference speed.

Interface

Every drive that does not belong to a simulated axis must be associated with a set value output module [▶ 41].

Drive types

- M2400-DAC1,
- M2400-DAC2,
- M2400-DAC3,
- M2400-Dac4,
- KL4XXX and PWM Terminal KL2502_30K (frq. cnt. pulse mode) and KL4132 (16 bit) (see KL5051),
- KL4XXX-NONLINEAR, new analog type for non linear curves,
- TWOSPEED,
- STEPPER,
- SERCOS,
- KL5051, BISSI drive KL5051 with 32 bits (see KL4XXX),
- AX2000, incremental with 32 bits (AX2000),
- SIMO611U, incremental with 32 bits,
- UNIVERSAL, variable bit mask (max. 32 bits, signed value),
- CBACKPLANE, variable bit mask (max. 32 bit, signed value).

Drive parameters

Parameter	Data type	Description
DRIVE-Mode	ENUM	reserved.
Invert Motor Polarity	BOOL	Direction of motor rotation is inverted. FALSE = in response to positive drive the axis moves in the direction of larger positions. TRUE = in response to positive drive the axis moves in the direction of smaller positions.
Minimum Drive Output Limitation	float	Lower control limit. The output to the drive is limited to this minimum value.
Maximum Drive Output Limitation	float	Upper control limit. The output to the drive is limited to this maximum value.

Analog drive

Analog does not mean here that the speed is represented by a voltage (e.g. ±10 V) or current (e.g. ±20 mA), but rather that the axis can be adjusted over an effectively continuous range of values. This is of course also possible for drives with digital interfaces such as the BISSI Terminals of type KL5051. Here a speed, which can be adjusted as an analog value, is transported in the form of digital information.

Analog parameters

Parameter	Data type	Description
Reference velocity	float	The reference velocity for the axis. When generating the setpoints for the axis, the pre-control assumes that the axis will react by moving at this velocity when driven with the reference output value.
Reference output	float	See above.
Drift compensation	float	This value is added to the drive control level. In this way, a constant offset can be added to the output, in order to compensate, for instance, for zero errors in analog drives.

Stepper motors

See: "Low Cost" stepper motor axis with digital control (24V / 2A) [▶ 21]

High/low speed motors

See: High/low speed motors [▶ 14].

SERCOS

Output scaling (only significant in the speed control operating mode) determines the relationship between the internal representation of the speed and the weighting of the speed data that is set in the drive unit. Since the calculation depends on a number of parameters that are internal to the drive unit, there is a facility for allowing the correct value to be calculated (the "Calculate" button). To do this, the necessary parameters are read from the drive, and the correct value is calculated (it is assumed that the system is operating and that SERCOS is at least in Phase 2). Download and Upload can be used to write or read the data to and from the running TwinCAT system.

5.3 Controller

The purpose of the controller is to operate on the basis of the set velocity differences (following error) or other set magnitudes (acceleration) in such a way that the following error is kept as small as possible and that the axis does not undergo any overshoots in position or velocity. A wide variety of controller versions is supported. Servo position controllers and special controllers for particular types of axis are available.

Controller types

Two different types of controller can be chosen: position controllers (controllers whose task is to control the actual position in such a way that it follows the set position as precisely as possible) and controllers for special axes (high/low speed axes, stepper motor, SERCOS). The way in which the controller operates is described under the position control loop [▶ 13] topic. Positions controller types: position controller P; following error proportional controller. Position controller with two P-constants: following error proportional controller with different constants for the stationary state and for movement. Position controller PID: position PID-T1 controller with proportional acceleration feed forward.

Automatic DAC offset adjustment

Any controller with no I component has automatic DAC offset adjustment as an option. This adjustment is only active when the velocity feed forward of the axis falls below a certain magnitude. This prevents it from being affected by the dynamic behaviour of the axis. If the axis is subject to position control (or is moving at a suitably low-velocity) an offset velocity is generated by integrating the control velocity. This is added to the output. The negative feedback of the position control loop results in a PT1 behaviour, which means that an exponential function is created.

Offset calibration parameters

Parameter	Data type	Unit	Description
Offset filter time	double	s	Time constant for the offset calibration
Offset limit	double		Relative control, above which the offset is kept constant.

If necessary, it is possible to affect the behavior of the offset calibration at runtime. A range of "switches" is available for this: for instance, at runtime the PLC or another ADS device can modify the parameters for the time constants and the pre-control limit.

If necessary, the offset calibration can be entirely switched off. In this case it will not always be possible to avoid a jump in the output voltage. Soft deactivation can be achieved with "fade out". This will reduce the adjustment to zero over time following its own curve. "Hold" mode can be activated if the adjustment is to be kept steady for a period of time. This can, for instance, be used if the power section of the drive is to be stopped temporarily. If the adjustment were to remain active, it would be impossible for the offset not to run out of control.

Acceleration feed forward

In addition to proportional feedback of the following error, nearly all position controllers contain a proportional acceleration feed forward (the K_a factor). This should normally be used only in association with the proportional component (K_v factor) of the position controller. It is necessary that the axis is adjusted for strict symmetry:

- when stationary, the following error is symmetrical about *0* (DAC offset).
- when moving steadily the following error is symmetrical about *0* (reference velocity).
- Set K_v.
- Measure the extreme value of the acceleration (deceleration) a_{+max} (a_{-max}), and the associated following error d_{+max} (d_{-max}) in the middle of the accelerating/braking phase.
- $K_{a+} = K_v \cdot d_{+max} / a_{+max}$
  $K_{a-} = K_v \cdot d_{-max} / a_{-max}$
  $K_a = (K_{a+} + K_{a-}) / 2$

Global parameters for a controller

Controller parameters

Parameter	Data type	Unit	Description
CONTROLLER mode	ENUM		Controller mode: only STANDARD is possible.
Weighting of the pre-control	float >=0		Relative weighting of the pre-control.
OPERATION MODE: position lag monitoring	BOOL		The default value of 1.0 corresponds to 100 % pre-control weighting.
Maximum position lag	float >=0	mm	Maximum permitted position lag.
Maximum position lag error filter time	float >=0	s	If the position lag exceeds the maximum permitted position lag error for a period of time that is longer than the maximum lag error filter time, and if the monitoring is active, the axis will be stopped instantaneously through a voltage output of 0 V and placed in the logical "error" state.

Position P controller

Controller parameters

Parameter	Data type	Unit	Description
Position control: proportional factor *Kv*	float >=0	[mm/s]/mm	Proportional gain of the P component. Output velocity = pre-control velocity + *Kv*· lag error.
OPERATION MODE: automatic offset calibration	BOOL		Activation or deactivation of the automatic offset calibration. The automatic offset calibration is only active in the range of the relative pre-control velocity defined by [-Offsetlimit,+Offsetlimit]. It calculates and activates a DAC offset that minimizes the lag error in the position control.
Offset filter time	float >=0	s	Integrator time constant.
Offset limit	float >=0		The upper limit of the active range of the offset calibration as a relative proportion of the output magnitude.

Position PP controller

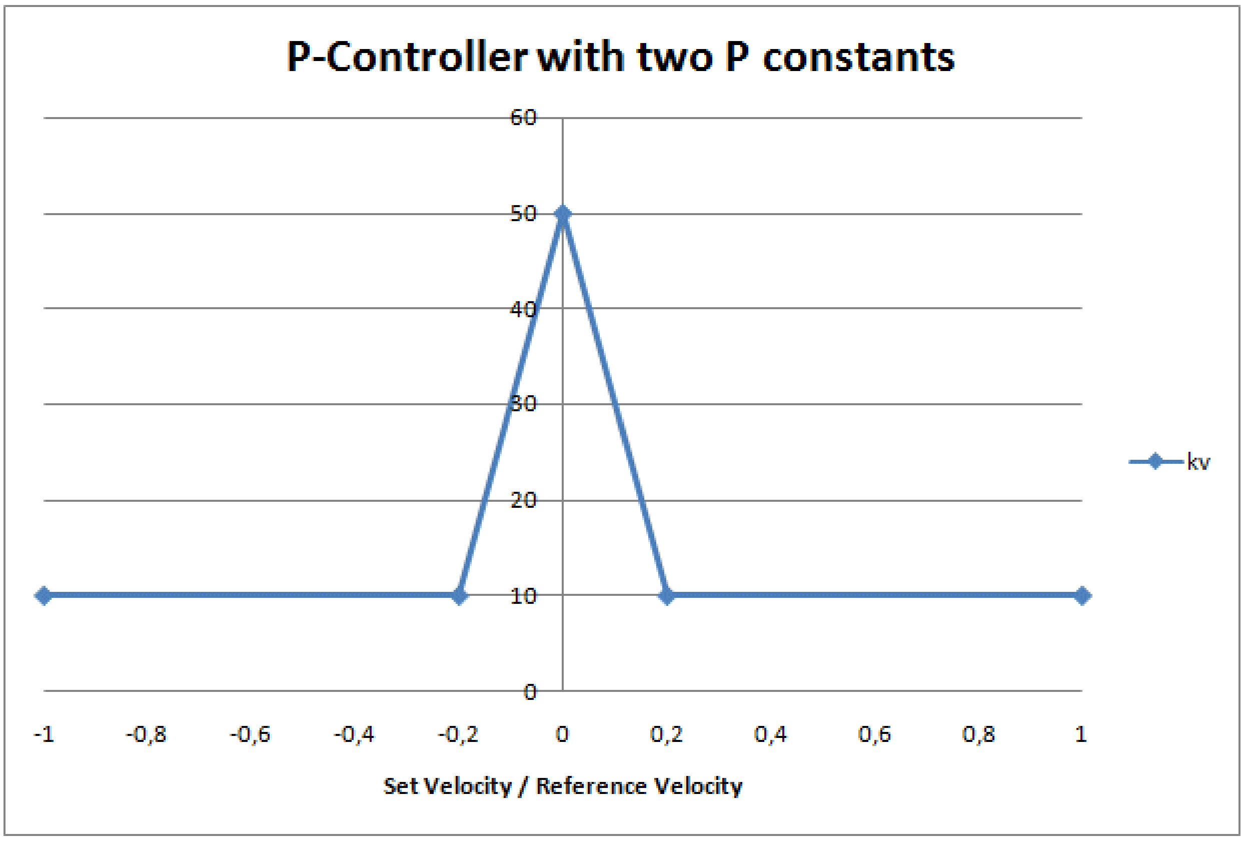
Controller parameters

Parameter	Data type	Unit	Description
Standstill: proportional factor *Kvs*	float >=0	[mm/s]/mm	Proportional gain of the P component when stationary. Output velocity when stationary = pre-control velocity + *Kvs* lag error
Position control travel: Proportional factor *Kvf*	float >=0	[mm/s]/min	Proportional gain of the P component when moving.
Position control: velocity threshold *Vdyn*	float >=0		Output velocity when moving = pre-control velocity + *Kvf* lag error.
OPERATION MODE: automatic offset calibration	BOOL		Activation or deactivation of the automatic offset calibration. The automatic offset calibration is only active in the range of the relative pre-control velocity defined by [-Offsetlimit,+Offsetlimit]. It calculates and activates a DAC offset that minimizes the lag error in the position control.
Offset filter time	float >=0	s	Integrator time constant.
Offset limit	float >=0		The upper limit of the active range of the offset calibration as a relative proportion of the output magnitude.
Acceleration pre-control proportional factor *Ka*	float >=0	s	Output velocity component = *Ka* set acceleration.

Example

A PP controller uses two P constants *Kvs* and *Kvf* and a velocity threshold *Vdyn* to define a function for a velocity dependent *kv* factor. The velocity threshold *Vdyn* is a relative value, calculated as *SetVelocity / ReferenceVelocity.*

The diagram below shows the connection



The sample is parameterized with *Kvs* = 50, *Kvf* = 10 and *Vdyn* = 0,2.

Position PID controller

Controller parameters

Parameter	Data type	Unit	Description
Position control: proportional factor *Kv*	float >=0	[mm/s]/mm	Proportional gain of the P component. Output velocity = pre-control velocity + Kv lag error.
Position control: integral action time *Tn*	float >=0	s	Integral action time of the I component (integration time)
Position control: rate time *Tv*	float >=0	s	Rate time of the real D component (D-T1 component).
Position control: damping time *Td*	float >=0	s	Damping time of the real D component (D-T1 component).
OPERATION MODE: automatic offset calibration	BOOL		Activation or deactivation of the automatic offset calibration. The automatic offset calibration is only active in the range of the relative pre-control velocity defined by [-Offsetlimit,+Offsetlimit]. It calculates and activates a DAC offset that minimizes the lag error in the position control.
Offset filter time	float >=0	s	Integrator time constant.
Offset limit	float >=0		The upper limit of the active range of the offset calibration as a relative proportion of the output magnitude.
Position control: acceleration pre-control *Ka*	float >=0	s	Output velocity component = *Ka* · set acceleration.

Controllers for special axes types (high/low speed, stepper motors, SERCOS)

Controller parameters

Parameter	**Data type**	**Unit**	**Description**
CONTROLLER mode	ENUM		Controller mode: only STANDARD is possible.
Weighting of the pre-control	float >=0		Relative weighting of the pre-control.
OPERATION MODE: position lag monitoring	BOOL		The default value of 1.0 corresponds to 100 % pre-control weighting.
Maximum position lag	float >=0	mm	Maximum permitted position lag.
Maximum position lag error filter time	float >=0	s	If the position lag exceeds the maximum permitted position lag error for a period of time that is longer than the maximum lag error filter time, and if the monitoring is active, the axis will be stopped instantaneously through a voltage output of 0 V and placed in the logical "error" state.

Delay time compensation

For every controller at there is an option of compensation for the delay between the time when a set value is output and the time when it has an effect. Here, the set position is delayed by an adjustable time in comparison with the set velocity. The delay time compensation is an element in the set value generator [▶ 40] for an axis.

5.4 Set value generators

Every axis is assigned to a set value generator, and this in turn consists of three components:

- block preparation generator (in the block preparation task): checking the start parameters and, in the case of master axes, calculation of the dynamic profile.
- block execution generator (in the block execution task): calculation of the local set values.
- asynchronous generator for reaction to asynchronous requirements (override, new end position, position compensation etc.).

Master generator: SVB

The SVB master generator checks whether a start is permissible in the current operating state of the axis, checks the start parameters in general and in relationship to the axis parameters, and calculates, from the **global set values**, a dynamic profile which is saved in compressed form (a run-time table) for the generation of the local set values. When these actions have been successfully completed, the SVB master generator **automatically** starts the SAF master generator.

Master generator: SAF

Each time the task is called, the SAF master generator calculates the **local set values** of from the runtime table for

- position (for the position controller),
- velocity with arithmetic sign (for the velocity feed forward),
- acceleration (for a controller with acceleration feed forward),
- direction (for the feed enable, which depends on the direction).

Master generator: Asynchronous

The generator for reaction to asynchronous instructions (override, new end position, position compensation etc.) checks the parameters of the instruction as well as whether the instruction is permissible in the current operating state, and initiates execution of the instruction the next time the SAF is called. This implies that final knowledge as to whether the instruction has indeed been initiated does not exist until **after** this SAF call.

Slave generator: SVB

The SVB slave generator checks whether the coupling is permitted and whether a start is permitted in the current operating state of the axis, checks the coupling parameters and sets global start parameters. When these actions have been successfully completed, it couples the slave **logically** to the master.

Slave generator: SAF

Each time the task is called, the SAF master generator calculates, from the local set values of the master axis (which can, in turn, be a slave axis) and the coupling factor, the **local set values** for

- position (for the position controller),
- velocity with arithmetic sign (for the velocity feed forward),
- acceleration (for a controller with acceleration feed forward),
- direction (for the feed enable, which depends on the direction).

Slave generator: asynchronous

The generator for reaction to asynchronous instructions (position compensation, online decoupling etc.) checks the parameters of the instruction as well as whether the instruction is permissible in the current operating state, and initiates execution of the instruction the next time the SEC task is called. This implies that final knowledge as to whether the instruction has indeed been initiated does not exist until after this SEC call.

Slave generator: flying saw

Because of the wide variety of its functions, the flying saw has its own type of set value generator, consisting of an SPP master generator working together with an SEC master generator in the waiting and synchronization phase. In the synchronous phase, the flying saw has an SEC slave generator. This can be converted into either an initialized SEC braking generator (in the stop phase) or (decoupled) online into a suitably initialized SEC master generator.

5.5 Input/Output

There are input/output connections for channels, axes, encoders and drives.

Input/Output for NC channels

Standard channels, in other words channels for PTP axes, do not require any input/output connections. These are only required for interpreter channels.

A variable of type PLCTONC_CHANNEL is to be created in the PLC for each channel. These variable(s) must be associated with the NC channel variables Channelx_FromPlc (to be found under channel inputs). A variable of type NCTOPLC_CHANNEL is to be created in the PLC for each channel. These variable(s) must be associated with the NC channel variables Channelx_ToPlc (to be found under channel outputs). These variables are required to allow signals to be exchanged between the NC and the PLC. The following data is exchanged here:

- NC block skipping
- interpreter working mode
- interpreter status and error codes
- channel overrides and program numbers
- M-functions as flying signal bits and as handshake

It is **not** normally necessary to associate the individual variables

The description of the cyclic channel interface can be found in the appendix to the NC I.

Input/output for axes

A variable of type NCAXLESTRUCT_FROMPLC is to be created in the PLC for each axis. These variable(s) must be associated with the NC axis variables Axisx_FromPlc (to be found under axis inputs). A variable of type NCAXLESTRUCT_TOPLC is to be created in the PLC for each axis. These variable(s) must be associated with the NC axis variables Axisx_ToPlc (to be found under axis outputs). These variables are required to allow signals to be exchanged between the NC and the PLC. The following data is exchanged here:

- controller and direction-dependent feed enables
- calibration cams
- calibration status
- positioning mode
- error codes and miscellaneous axis status information
- axis overrides
- coupling status

It is **not** normally necessary to associate the individual variables

A description of the interfaces is to be found under

- cyclic axis interface from PLC to NC [▶ 53]
- cyclic axis interface from NC to PLC [▶ 46]
- cyclic interface from NC to encoder and drive [▶ 43]

Input/output for encoders

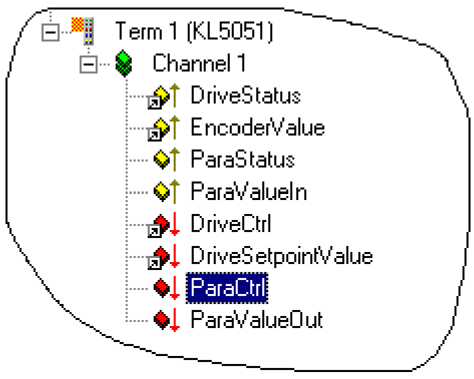
Every encoder that is not of the "simulation" type must be associated with an actual value acquisition module. It is necessary to ensure here that the types in use are compatible with one another. A suitable type of encoder must be chosen.

cyclic interface from NC to the encoder [▶ 43]

Input/output for drives

Every drive that does not belong to a simulated axis must be associated with a set value output module. It is necessary to ensure here that the types in use are compatible with one another. A suitable type of drive must be chosen.

cyclic interface from NC to the drive [▶ 43]



5.6 Process image of an NC axis

The process image of an axis is used to connect it to different drive components. In the simplest case, a link is established between the axis and the drive and the necessary links between the process images are made automatically. In individual cases, in particular when unknown hardware components are integrated in the system, these links must be established manually. To this end, the process image of an NC axis is described in the following.

NOTICE

The data structures described here form an internal interface between NC drivers and connected drive hardware. This interface is constantly under development and can change in the future.

Encoder process image of an axis

Various encoder hardware or the corresponding Bus Terminals for detecting the position of an axis are connected via the encoder process image (cyclic data exchange). Insofar as this hardware is directly supported by the system, no manual configuration of individual variables is necessary.

Input data to the encoder process image of an axis

I/O variable	Description	Remark
nInData1	Current actual position of the encoder or the drive in *increments*. The incremental value is processed by the NC and converted to the actual position in physical units, e.g. mm or degrees. Overflows of the incremental values are also counted by the NC. The system normally makes no distinction between incremental and absolute encoders (see also encoder parameter *Reference System INC/ ABS*).	Unit: increments [INC] Data type: DINT (32 bit). sometimes also INT (16 bit)
nInData2	Optional actual latch position of the encoder or the drive in *increments*. The incremental value is processed by the NC and converted to the actual position in physical units, e.g. mm or degrees. (see also function block MC_TouchProbe TcMc.lib).	Unit: increments [INC] Data type: DINT (32 bit). sometimes also INT (16 bit)
nStatus1	Optional status information e.g.: *Participant in Data Exchange*, *Encoder Error*, communication to *Position Latch* or *Register Communication*.	This information depends on the type of encoder or drive.
nStatus2	Additional optional status information e.g.: Communication to *Position Latch*, communication *Encoder Reset*, communication to *Read an Absolute Position*	This information depends on the type of encoder or drive.
nStatus3	reserved	
nStatus4	Optional field bus-dependent IO status such as *WcState* (Working Counter) in the case of EtherCAT or *CdlState* in the case of Beckhoff Lightbus.	Meaning: 0 = I/O data valid 1 = I/O data invalid

Output data from the encoder process image of an axis

I/O variable	Description	Remark
nOutData1	Current actual position of the encoder or drive in *increments*, which the NC copies directly from the input variable *nInData1* to the output variable *nOutData1*.	Unit: increments [INC] Data type: DINT
nOutData2	reserved	
nCtrl1	Optional control information e.g.: Communication to *Position Latch*, communication *Encoder Reset*, *Register Communication*	This information depends on the type of encoder or drive.
nCtrl2	Additional optional control information e.g.: Communication to *Position Latch*, communication *Encoder Reset*, communication to *Read an Absolute Position*	This information depends on the type of encoder or drive.
nCtrl3	reserved	
nCtrl4	reserved	

Drive process image of an axis

Various drive hardware or the corresponding Bus Terminals (+/- 10V, PWM etc.) are connected via the drive process image. Insofar as this hardware is directly supported by the system, no manual configuration is necessary.

Input data to the drive process image of an axis

I/O variable	Description	Remark
nInData1	reserved	
nInData2	reserved	
nStatus1	Optional status information e.g.: *Drive Error*, *Drive Enable/Disable*, *Communication Drive State Machine* (e.g. EtherCAT, SERCOS, CANopen, Profibus), *Register Communication*.	This information depends on the type of drive.
nStatus2	Additional optional status information e.g.: *Drive Error*, *Drive Enable/Disable*, *Communication Drive State Machine* (e.g. EtherCAT, SERCOS, CANopen, Profibus), *Register Communication*.	This information depends on the type of drive.
nStatus3	reserved	
nStatus4	Optional field bus-dependent IO status such as *WcState* (Working Counter) in the case of EtherCAT or *CdlState* in the case of Beckhoff Lightbus.	Meaning: 0 = I/O data valid 1 = I/O data invalid

Output data from the drive process image of an axis

I/O variable	Description	Remark
nOutData1	Current set velocity or current set position in *increments*.	This information depends on the type of drive.

I/O variable	Description	Remark
	The set velocity or set position of the NC in physical units, e.g. mm or degrees, is mathematically converted back to an incremental value by the NC and transferred to the drive. Overflows of the incremental value are hereby taken into account by the NC in the set position. Depending on the type of drive, a standardization of the set velocity or set position is carried out in *increments*. If the drive type *Universal Drive* has been selected, nOutData1 contains the overall velocity (incl. position control part) with sign.	
nOutData2	Current set velocity or current set position in *increments*. The set velocity or set position of the NC in physical units, e.g. mm or degrees, is mathematically converted back to an incremental value by the NC and transferred to the drive. Overflows of the incremental value are hereby taken into account by the NC in the set position. Depending on the type of drive, a standardization of the set velocity or set position is carried out in *increments*. If the drive type *Universal Drive* has been selected, nOutData2 contains the value of the overall velocity (incl. position control part, without sign).	This information depends on the type of drive.
nCtrl1	Optional control information: e.g.: *Drive Reset*, *Drive Enable/Disable*, *Communication Drive State Machine* (e.g. EtherCAT, SERCOS, CANopen, Profibus), *Register Communication*.	This information depends on the type of drive.
nCtrl2	Additional optional control information: Digital direction output of the setpoint generator (corresponds to the sign of the set velocity, hence without position controller) Digital Outputs Setpoint Generator: 0x41 (0100 0001) = Minus 0x42 (0100 0010) = Plus 0x80 (1000 0000) = Stop The only exception to this is the AX2xxx-B200/B900 drive. In this case, communication takes place with the Drive State Machine.	This information depends on the type of drive (except for AX2xxx B200 and B900)
nCtrl3	Additional optional control information:	

I/O variable	Description	Remark
	Digital direction output or drive stages of the overall output (sum of the setpoint generator and the position controller) Digital Outputs (Setpoint Generator + Position Controller): 0x41 (0100 0001) = Minus 0x42 (0100 0010) = Plus 0x80 (1000 0000) = Stop	
nCtrl4	reserved	

5.7 Cyclic NC/PLC interfaces

5.7.1 NC->PLC axis interface (128 bytes)

TYPE NCTOPLC_AXLESTRUCT

```
TYPE NCTOPLC_AXLESTRUCT
STRUCT
    nStateDWord          : DWORD; (* Status double word *)
    nErrorCode           : DWORD; (* Axis error code *)
    nAxisState           : DWORD; (* Axis moving status *)
    nAxisModeCon         : DWORD; (* Axis mode confirmation (feedback from NC) *)
    nCalibrationState    : DWORD; (* State of axis calibration (homing) *)
    nCoupleState         : DWORD; (* Axis coupling state *)
    nSvbEntries          : DWORD; (* SVB entries/orders (SVB = Set preparation task) *)
    nSafEntries          : DWORD; (* SAF entries/orders (SAF = Set execution task *)
    nAxisId              : DWORD; (* Axis ID *)
    nOpModeDWord         : DWORD; (* Current operation mode *)
    nReserved2_HIDDEN    : DWORD; (* reserved *)
    fPosIst              : LREAL; (* Actual position (absolut value from NC) *)
    fModuloPosIst        : LREAL; (* Actual position as modulo value (e.g. in degrees) *)
    nModuloTurns         : DINT;  (* Actual modulo turns *)
    fVeloIst             : LREAL; (* Actual velocity (optional) *)
    fPosDiff             : LREAL; (* Position difference (lag distance) *)
    fPosSoll             : LREAL; (* Setpoint position *)
    fVeloSoll            : LREAL; (* Setpoint velocity *)
    fAccSoll             : LREAL; (* Setpoint acceleration, OLD: "fReserve1_HIDDEN" *)
    fReserve2_HIDDEN     : LREAL; (* reserved *)
    fReserve3_HIDDEN     : LREAL; (* reserved *)
    fReserve4_HIDDEN     : LREAL; (* reserved *)
END_STRUCT
END_TYPE
```

TYPE NCTOPLC_AXLESTRUCT2

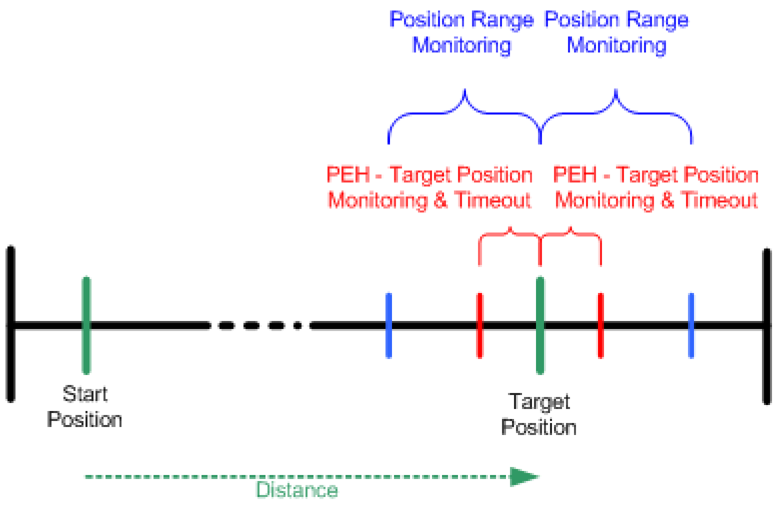
```
TYPE NCTOPLC_AXLESTRUCT2
STRUCT
    nStateDWord              : DWORD; (* Status double word *)
    nErrorCode               : DWORD; (* Axis error code *)
    nAxisState               : DWORD; (* Axis moving status *)
    nAxisModeCon             : DWORD; (* Axis mode confirmation (feedback from NC) *)
    nCalibrationState        : DWORD; (* State of axis calibration (homing) *)
    nCoupleState             : DWORD; (* Axis coupling state *)
    nSvbEntries              : DWORD; (* SVB entries/orders (SVB = Set preparation task) *)
    nSafEntries              : DWORD; (* SAF entries/orders (SAF = Set execution task *)
    nAxisId                  : DWORD; (* Axis ID *)
    nOpModeDWord             : DWORD; (* Current operation mode *)
    nActiveControlLoopIndex : WORD;  (* Active control loop index (equivalent to old variable "nCtrl
LoopIndex") *)
    nControlLoopIndex        : WORD;  (* Axis control loop index (0, 1, 2, … when multiple control lo
ops are used) *)
    fPosIst                  : LREAL; (* Actual position (absolut value from NC) *)
    fModuloPosIst            : LREAL; (* Actual position as modulo value (e.g. in degrees) *)
    nModuloTurns             : DINT;  (* Actual modulo turns *)
    fVeloIst                 : LREAL; (* Actual velocity (optional) *)
    fPosDiff                 : LREAL; (* Position difference (lag distance) *)
    fPosSoll                 : LREAL; (* Setpoint position *)
    fVeloSoll                : LREAL; (* Setpoint velocity *)
    fAccSoll                 : LREAL; (* Setpoint acceleration, OLD: "fReserve1_HIDDEN" *)
    fPosTarget                : LREAL;  (* Estimated target position *)
    fModuloPosSoll            : LREAL;  (* Setpoint modulo position (e.g. in degrees) *)
    nModuloTurnsSoll          : DINT;   (* Setpoint modulo turns *)
    nCmdNo                    : WORD;   (* Continuous actual command number *)
    nCmdState                 : WORD;   (* Command state *)
END_STRUCT
END_TYPE
```

No.	Data type	Byte	Bit	Def.-Range	Variable name	Variable name (from 2.11 or TcMc2)	Description
1	UINT32	0-3	-	-	nStateDWord	StateDWord	State double word. See also detailed description of the StateDWord [▶ 51]
			0	0/1	Operational	Operational	Axis is ready for operation
			1	0/1	Homed	Homed	Axis is referenced ("axis calibrated")
			2	0/1	NotMoving	NotMoving	Axis is logically stationary ("Axis not moving")
			3	0/1	InPositionArea	InPositionArea	Axis is in position window (physical feedback)
			4	0/1	InTargetPosition	InTargetPosition	Axis is at target position (PEH) (physical feedback)
			5	0/1	Protected	Protected	Axis is in protected operation mode (e.g. slave axis)
			6	0/1	ErrorPropagation Delayed	ErrorPropagationD elayed	Axis signals a preliminary error warning (from TC 2.11)
			7	0/1	HasBeenStopped	HasBeenStopped	Axis has been stopped or is presently executing a stop
			8	0/1	HasJob	HasJob	Axis has job, is executing job
			9	0/1	PositiveDirection	PositiveDirection	Axis moving to logically larger values
			10	0/1	NegativeDirection	NegativeDirection	Axis moving to logically smaller values
			11	0/1	HomingBusy	HomingBusy	Axis is referencing ("axis is being calibrated")
			12	0/1	ConstantVelocity	ConstantVelocity	Axis has reached its constant velocity or rotary speed
			13	0/1	Compensating	Compensating	Section compensation passive[0]/active[1] (see MC_MoveSuperImposed)
			14	0/1	ExtSetPointGenE nabled	ExtSetPointGenEn abled	Enable external setpoint generation
			15	0/1			Operation mode not yet executed (Busy). Not yet released!
			16	0/1	ExternalLatchVali d	ExternalLatchValid	External latch value or measuring probe has become valid
			17	0/1	NewTargetPos	NewTargetPos	Axis has received a new end position or a new velocity
			18	0/1			Axis not in target position or cannot/will not reach it (e.g. axis stop). Not yet released!
			19	0/1	ContinuousMotio n	ContinuousMotion	Axis executing endless positioning task

No.	Data type	Byte	Bit	Def.-Range	Variable name	Variable name (from 2.11 or TcMc2)	Description
			20	0/1	ControlLoopClosed	ControlLoopClosed	Axis ready to operate and axis control loop closed (e.g. position control)
			21	0/1	CamTableQueued	CamTableQueued	New table ready for "Online Change" and waiting for activation
			22	0/1	CamDataQueued	CamDataQueued	Table data (MF) ready for "Online Change" and waiting for activation
			23	0/1	CamScalingPending	CamScalingPending	Table scalings ready for "Online Change" and waiting for activation
			24	0/1	CmdBuffered	CmdBuffered	Follow-up command is available in the command buffer (see BufferMode) (from TwinCAT V2.10 Build 1311)
			25	0/1	PTPmode	PTPmode	Axis in PTP operating mode (no slave, no NCI axis, no FIFO axis) (from TC 2.10 Build 1326)
			26	0/1	SoftLimitMinExceeded	SoftLimitMinExceeded	Software minimum end position is active/occupied (from TC 2.10 Build 1327)
			27	0/1	SoftLimitMaxExceeded	SoftLimitMaxExceeded	Software maximum end position is active/occupied (from TC 2.10 Build 1327)
			28	0/1	DriveDeviceError	DriveDeviceError	Drive hardware has an error (no warning); interpretation possible only if drive is in I/O data exchange. e.g. EtherCAT "OP" state (from TC 2.10 Build 1326)
			29	0/1	MotionCommandsLocked	MotionCommandsLocked	Axis is blocked for motion commands (TcMc2)
			30	0/1	IoDataInvalid	IoDataInvalid	I/O data invalid (e.g. "WcState" or "CdlState" of the fieldbus)
			31	0/1	Error	Error	Axis is in an error state
2	UINT32	4-7	-	≥0	nErrorCode	ErrorCode	Axis error code
3	UINT32	8-11	-	ENUM	nAxisState	AxisState	Present state of the axis movement
4	UINT32	12-15	-	ENUM	nAxisModeCon	AxisModeConfirmation	Axis operating mode (feedback from the NC)
5	UINT32	16-19	-	ENUM	nCalibrationState	HomingState	Reference status of the axis ("calibration status")
6	UINT32	20-23	-	ENUM	nCoupleState	CoupleState	Axis coupling state
7	UINT32	24-27	-	≥0	nSvbEntries	SvbEntries	SPP entries/tasks
8	UINT32	28-31	-	≥0	nSafEntries	SafEntries	SEC entries/tasks (NC interpreter, FIFO group)
9	UINT32	32-35	-	>0	nAxisId	AxisId	Axis ID
10	UINT32	36-39	-	-	nOpModeDWord, bOpMode	OpModeDWord, OpMode...	Axis operation mode double word
			0	0/1	PosAreaMonitoring	...PosAreaMonitoring	Position range monitoring

No.	Data type	Byte	Bit	Def.-Range	Variable name	Variable name (from 2.11 or TcMc2)	Description
			1	0/1	TargetPosMonitoring	...TargetPosMonitoring	Target position window monitoring
			2	0/1	Loop	...Loop	Looping distance
			3	0/1	MotionMonitoring	...MotionMonitoring	Physical motion monitoring
			4	0/1	PEHTimeMonitoring	...PEHTimeMonitoring	PEH time monitoring
			5	0/1	BacklashComp	...BacklashComp	Backlash compensation
			6	0/1	DelayedErrorReaction	...DelayedErrorReaction	Delayed error reaction of the NC
			7	0/1	Modulo	...Modulo	Modulo axis (modulo display)
			8-15	0/1			RESERVE
			16	0/1	PosLagMonitoring	...PosLagMonitoring	Lag monitoring - position
			17	0/1	VeloLagMonitoring	...VeloLagMonitoring	Lag monitoring - velocity
			18	0/1	SoftLimitMinMonitoring	...SoftLimitMinMonitoring	End position monitoring min.
			19	0/1	SoftLimitMaxMonitoring	...SoftLimitMaxMonitoring	End position monitoring max.
			20	0/1	PosCorrection	...PosCorrection	Position correction ("Measuring system error compensation")
			21	0/1	AllowSlaveCommands	...AllowSlaveCommands	Allow motion commands to slave axes
			22	0/1			RESERVE
			23	0/1	ApplicationRequest	ApplicationRequest	Request bit for the application software (PLC code), e.g. for an "ApplicationHomingRequest" from TwinCAT V2.11 Build 1546
			24-31	0/1			RESERVE
11	UINT16	40-41	-	≥0	nActiveControlLoopIndex	ActiveControlLoopIndex	Active axis control loop index (identical with old variable "nCtrlLoopIndex") from TwinCAT V2.10 Build 1311
12	UINT16	42-43	-	≥0	nControlLoopIndex	ControlLoopIndex	Axis control loop index (0, 1, 2, if more than one axis control loop is used) from TwinCAT V2.10 Build 1311
13	REAL64	44-51	-	±∞	fPosIst	ActPos	Actual position (calculated absolute value)
14	REAL64	52-59	-	>∞	fModuloPosIst	ModuloActPos	Modulo actual position (calculate value in, for example, degrees)
15	INT32	60-63	-	±∞	nModuloTurns	ModuloActTurns	Modulo actual rotations
16	REAL64	64-71	-	±∞	fVeloIst	ActVelo	Actual velocity (optional)
17	REAL64	72-79	-	±∞	fPosDiff	PosDiff	Lag error (position)

No.	Data type	Byte	Bit	Def.-Range	Variable name	Variable name (from 2.11 or TcMc2)	Description
18	REAL64	80-87	-	±∞	fPosSoll	SetPos	Set position (calculated absolute value)
19	REAL64	88-95	-	±∞	fVeloSoll	SetVelo	Set velocity
20	REAL64	96-103	-	±∞	fAccSoll	SetAcc	Set acceleration
21	REAL64	104-111	-	±∞	fPosTarget	TargetPos	Estimated target position of the axis from TwinCAT V2.10 Build 1311
22	REAL64	112-119	-	>∞	fModuloPosSoll	ModuloSetPos	Modulo set position (calculate value in, for example, degrees) from TwinCAT V2.10 Build 1311
23	INT32	120-123	-	±∞	nModuloTurnsSoll	ModuloSetTurns	Modulo set rotations from TwinCAT V2.10 Build 1311
24	UINT16	124-125	-	≥0	nCmdNo	CmdNo	Command number of the active axis job (see BufferMode) from TwinCAT V2.10 Build 1311
25	UINT16	126-127	-	≥0	nCmdState	CmdState	Command status information (see Buffer Mode) from TwinCAT V2.10 Build 1311


Position Range Monitoring
Position Range Monitoring
PEH - Target Position Monitoring & Timeout
PEH - Target Position Monitoring & Timeout
Start Position
Target Position
Distance

Description of the contents of the individual fields:

Define	Referencing state of the axis (nCalibrationState or HomingState)
0	Referencing process completed (READY)
1	Continuous start in the direction of the referencing cam. **Info:** If the cam is occupied at the beginning, then the program starts directly with referencing status 3.
2	Wait for positive edge of the referencing cam and initiate axis stop
3	Wait until the axis is stopped (check whether cam is still occupied) and then endless start of the referencing cam in the direction of the sync pulse
4	Wait for falling edge of the referencing cam
5	Activate latch, wait until latch has become valid and then initiate axis stop

Define	Referencing state of the axis (nCalibrationState or HomingState)
6	If axis has stopped, then set actual position (actual position = reference position + braking distance)

See also: Notes on MC_Home

Define	Coupling state of the axis (nCoupleState or CoupleState)
0	Single axis that is neither a master nor a slave (SINGLE)
1	Master axis with any number of slaves (MASTER)
2	Slave axis that is the master of another slave (MASTERSLAVE)
3	Slave axis only (SLAVE)
Define	Master: motion state / travel phase of the continuous master axis (servo) (nAxisState or AxisState)
0	Setpoint generator not active (INACTIVE)
1	Setpoint generator active (RUNNING)
2	Velocity override is zero (OVERRIDE_ZERO)
3	Constant velocity (PHASE_VELOCONST)
4	Acceleration phase (PHASE_ACCPOS)
5	Deceleration phase (PHASE_ACCNEG)
Define	Master: motion state / travel phase of the discrete master axis (rapid/creep) (nAxisState or AxisState)
0	Setpoint generator not active
1	Travel phase (rapid or slow traverse)
2	Switchover delay from rapid to slow traverse
3	Creep motion (within the creep distance)
4	Deceleration time (starting from the braking distance in front of the target)
Define	Slave: motion state / travel phase of the continuous slave axis (servo) (nAxisState or AxisState) Note: for the time being only for slaves of the synchronizing generator type!
0	Slave generator not active (INACTIVE)
11	Slave is in a movement pre-phase (PRE-PHASE)
12	Slave is synchronizing (SYNCHRONIZING)
13	Slave is synchronized and moves synchronously (SYNCHRON)

Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT v2.7.0	PC (i386)	PlcNc.Lib
TwinCAT v2.8.0	PC (i386)	TcNC.Lib

5.7.2 Description of StateDWord

The *StateDWord* is a 32 bit data word in the axis interface NC->SPS [▸ 46]. The detailed function of each status bit is described in the table below.

Requirements

Bit	Variable-Name	Description
0	Operational	Axis is ready for operation
1	Homed	Axis has been referenced/ homed ("Axis calibrated")
2	NotMoving	Axis is logically stationary ("Axis not moving")
3	InPositionArea	Axis is in position window (physical feedback)

Bit	Variable-Name	Description
4	InTargetPosition	Axis is at target position (PEH) (physical feedback)
5	Protected	Axis is in a protected operating mode (e.g. as a slave axis)
6	ErrorPropagationDelayed	Axis signals an error pre warning (from TC 2.11)
7	HasBeenStopped	Axis has been stopped or is presently executing a stop
8	HasJob	Axis has instructions, is carrying instructions out
9	PositiveDirection	Axis moving to logically larger values
10	NegativeDirection	Axis moving to logically smaller values
11	HomingBusy	Axis referenced ("Axis being calibrated")
12	ConstantVelocity	Axis has reached its constant velocity or rotary speed
13	Compensating	Section compensation passive[0]/active[1] (s. "MC_MoveSuperImposed")
14	ExtSetPointGenEnabled	External setpoint generator enabled
15		Operating mode not yet executed (Busy). Not implemented yet!
16	ExternalLatchValid	External latch value or sensing switch has become valid
17	NewTargetPos	Axis has a new target position or a new velocity
18		Axis is not at target position or cannot reach the target position (e.g. stop).Not implemented yet!
19	ContinuousMotion	Axis has target position (±) endless
20	ControlLoopClosed	Axis is ready for operation and axis control loop is closed (e.g. position control)
21	CamTableQueued	CAM table is queued for "Online Change" and waiting for activation
22	CamDataQueued	CAM data (only MF) are queued for "Online Change" and waiting for activation
23	CamScalingPending	CAM scaling are queued for "Online Change" and waiting for activation
24	CmdBuffered	Following command is queued in then command buffer (s. Buffer Mode) (from TwinCAT V2.10 Build 1311)
25	PTPmode	Axis in PTP mode (no slave, no NCI axis, no FIFO axis) (from TC 2.10 Build 1326)
26	SoftLimitMinExceeded	Position software limit switch minimum is exceeded (from TC 2.10 Build 1327)
27	SoftLimitMaxExceeded	Position software limit switch maximum is exceeded (from TC 2.10 Build 1327)
28	DriveDeviceError	Hardware drive device error (no warning), interpretation only possible when drive is data exchanging, e.g. EtherCAT "OP"-state (from TC 2.10 Build 1326)
29	MotionCommandsLocked	Axis is locked for motion commands (TcMc2)
30	IoDataInvalid	IO data invalid (e.g. 'WcState' or 'CdlState')
31	Error	Axis is in a fault state

5.7.3 TYPE PLCTONC_AXLESTRUCT

TYPE PLCTONC_AXLESTRUCT

```
TYPE PLCTONC_AXLESTRUCT
STRUCT
    nDeCtrlDWord       : DWORD;  (* control double word *)
    nOverride          : DWORD;  (* velocity override *)
    nAxisModeReq       : DWORD;  (* axis operating mode (PLC request) *)
    nAxisModeDWord     : DWORD;  (* *)
    fAxisModeLReal     : LREAL;  (* *)
    fActPosCorrection  : LREAL;  (* correction value for current position *)
    fExtSetPos         : LREAL;  (* external position setpoint *)
    fExtSetVelo        : LREAL;  (* external velocity setpoint *)
    fExtSetAcc         : LREAL;  (* external acceleration setpoint *)
    nExtSetDirection   : DINT;      (* external direction setpoint *)
    nReserved1         : DWORD;  (* reserved *)
    fExtCtrlOutput     : LREAL;  (* external controller output *)
    nReserved_HIDDEN   : ARRAY [72..127] OF BYTE;
END_STRUCT
END_TYPE
```

For each NC axis a data block of 128 bytes is available for data transport from the NC to the PLC, and another data block, also 128 bytes, is available for data transport from the PLC to the NC. The PLC programmer must create a variable for each direction and each axis, and must fix them in the I/O area with the AT instruction to the input and output area. The assignment of NC variables to PLC variables is carried out by the TwinCAT System Manager.

No.	Data type	Byte	Bit	Def.-Range	Variable name	Variable name (from 2.11 or TcMc2)	Description
1	UINT32	0-3	-	0/1	nDeCtrlDWord	ControlDWord	Control double word
			0	0/1	Enable	Enable	Controller enable
			1	0/1	FeedEnablePlus	FeedEnablePlus	Feed enable plus
			2	0/1	FeedEnableMinus	FeedEnableMinus	Feed enable minus
			3	0/1	-	-	RESERVE
			4	0/1	-	-	RESERVE
			5	0/1	HomingSensor	HomingSensor	Referencing cam or referencing sensor
			6	0/1	-	-	RESERVE
			7	0/1	-	-	RESERVE
			8	0/1	AcceptBlockedDrive	AcceptBlockedDrive	Accept blocking of the drive setpoint adoption (e.g. hardware end positions) from TwinCAT V2.10 Build 1311
			9	0/1	BlockedDriveDetected	BlockedDriveDetected	User signal *Axis is blocked* (e.g. mechanical fixed stop). Not yet released!
			10-29	0/1	-	-	RESERVE
			30	0/1	PlcDebugFlag	PlcDebugFlag	PLC debug function. For internal use only!
			31	0/1	NcDebugFlag	NcDebugFlag	NC debug function. For internal use only!
2	UINT32	4-7	-	0...1000000	nOverride	Override	Velocity override (0% to 100%)
3	UINT32	8-11	-		nAxisModeReq	AxisModeRequest	Axis operating mode. Only provided for internal use!

No.	Data type	Byte	Bit	Def.-Range	Variable name	Variable name (from 2.11 or TcMc2)	Description
4	UINT32	12-15	-		nAxisModeDWord	AxisModeDWord	Only provided for internal use!
5	REAL64	16-23	-		fAxisModeLReal	AxisModeLReal	Only provided for internal use!
6	REAL64	24-31	-		fActPosCorrection	PositionCorrection	Actual position correction value
7	REAL64	32-39	-		fExtSetPos	ExtSetPos	External set position
8	REAL64	40-47	-		fExtSetVelo	ExtSetVelo	External set velocity
9	REAL64	48-55	-		fExtSetAcc	ExtSetAcc	External set acceleration
10	INT32	56-59	-		nExtSetDirection	ExtSetDirection	External set travel direction [-1,0,1]
11	UINT32	60-63	-		nReserved1	-	RESERVE
12	REAL64	64-71	-		fExtCtrlOutput	ExtControllerOutput	External controller output. Not yet released!
13	REAL64	72-79	-	±∞	-	GearRatio1	Gear ratio (coupling factor) 1
14	REAL64	80-87	-	±∞	-	GearRatio2	Gear ratio (coupling factor) 2
15	REAL64	88-95	-	±∞	-	GearRatio3	Gear ratio (coupling factor) 3
16	REAL64	96-103	-	±∞	-	GearRatio4	Gear ratio (coupling factor) 4
17	-	104-127	-	-	nReserved	-	RESERVE

Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT v2.7.0	PC (i386)	PlcNc.Lib
TwinCAT v2.8.0	PC (i386)	TcNC.Lib

5.7.4 Discrete high/low speed axis (two speed)

Starting velocity value range V	Interpretation of the starting velocity with 100% override (required travel velocity / travel stage)
V > 50	Rapid traverse
0 < V £ 50	Slow traverse
V ≤ 0	ERROR
Value range override X	Interpretation of the override value (100% = 1,000,000)
X > 50% (500000)	Rapid traverse
0% < X ≤ 50 % (500000)	Slow traverse
X = 0%	Stationary (tolerance window: <100 = <0.01%)

An override modification (also override = 0) only takes effect within the main travel phase [▶ 14].
If the override is set to 0 within one of the braking phases [▶ 14], the initiated braking phase is ended without being influenced.

5.7.5 Drive interface for high/low speed axes NC->IO (12 bytes)

No.	Data type	Byte	Bit	Def.-Range	Variable name	Description
1	UINT32	0-3	-	-	nOutData1	Drive output data 1 (NC->I/O)
2	UINT32	4-7	-	-	nOutData2	Drive output data 2 (NC->I/O)
3	UINT8	8	-	-	nControlByte	Control byte
			0	0/1	bMinusHigh	Direction: negative velocity: fast
			1	0/1	bMinusLow	Direction: negative velocity: slow
			2	0/1	bPlusLow	Direction: positive velocity: slow
			3	0/1	bPlusHigh	Direction: positive velocity: fast
			4	0/1	-	RESERVE
			5	0/1	-	RESERVE
			6	0/1	bBreakInv	Inverse braking bit (0 = ACTIVE, 1 = PASSIVE)
			7	0/1	bBreak	Braking bit (0 = PASSIVE, 1 = ACTIVE)
4	UINT8	9	-	-	nExtControlByte	Extended control byte
			0	0/1	bDirectionMinus	Direction: negative
			1	0/1	bDirectionPlus	Direction: positive
			2	0/1	bVeloLow	Velocity: slow
			3	0/1	bVeloHigh	Velocity: high
			4	0/1	-	RESERVE
			5	0/1	-	RESERVE
			6	0/1	bBreakInv	Inverse braking bit (0 = ACTIVE, 1 = PASSIVE)
			7	0/1	bBreak	Braking bit (0 = PASSIVE, 1 = ACTIVE)
5	UINT16	10-11	-	-	nReserved	Reserved bytes

An axis start will only be initiated if the distance from the target point is in fact larger than the parameterized braking distance.

5.7.6 "Low Cost" stepper motor axis with digital control (stepper)

Drive interface for "low cost" stepper motor axis NC → I/O (12 bytes)

No.	Data type	Byte	Bit	Def. Range	Variable Name	Description
1	INT32	0-3	-	-	nOutData1	Drive output data 1 (NC->I/O)
2	INT32	4-7	-	-	nOutData2	Drive output data 2 (NC->I/O)
3	UINT8	8	-	-	nControlByte	Control byte
3.0	...	8	0	0/1	bPhaseA	Phase A
3.1	...	8	1	0/1	bPhaseAInv	Phase A inverse
3.2	...	8	2	0/1	bPhaseB	Phase B
3.3	...	8	3	0/1	bPhaseBInv	Phase B inverse

No.	Data type	Byte	Bit	Def. Range	Variable Name	Description
3.4	...	8	4	0/1	-	RESERVE
3.5	...	8	5	0/1	-	RESERVE
3.6	...	8	6	0/1	bBreakInv	Inverse braking bit (0 = ACTIVE, 1 = PASSIVE)
3.7	...	8	7		bBreak	Braking bit (1 = ACTIVE, 0 = PASSIVE)
4	UINT8	9	-	-	nExtControlByte	Extended control byte
4.0	...	9	0	0/1	bFrequency	Frequency (square wave signal)
4.1	...	9	1	0/1	bDirectionPlus	Direction: positive
4.2	...	9	2	0/1	-	RESERVE
4.3	...	9	3	0/1	-	RESERVE
4.4	...	9	4	0/1	-	RESERVE
4.5	...	9	5	0/1	-	RESERVE
4.6	...	9	6	0/1	-	RESERVE
4.7	...	9	7	0/1	-	RESERVE
5	UINT16	10-11	-	-	nReserved	Reserved bytes

6 Axis commissioning

General

The requirements for commissioning axes [▶ 57] include, in particular, knowledge and use of the NC's safety functions along with a grasp of the necessary precautions along with the observation of a specific sequence of commissioning steps for the axis elements (encoder, drive and controller), as well as setting the axis parameters.

Encoders and actual values

Determination and setting the most important encoder parameters [▶ 59]: encoder polarity, scaling (increments per mm), position offset (zero offset shift).

Drive and voltage output

Determination [▶ 61] and setting the most important drive parameters: drive polarity, reference output, reference speed, drift compensation.

Position control and reduction of the following error

Determination and setting the most important position control parameters [▶ 62]: Kv factor and parameters for reduction of the following error.

Axis parameters, axis monitoring and axis calibration

Determination and setting the most important axis parameters [▶ 64]: maximum velocity, following error monitoring, end position monitoring and the parameters for dynamic behaviour and calibration.

6.1 General

Axis commissioning particularly requires knowledge and use of the safety functions of the NC, application of the necessary safety precautions, and the observance of a particular sequence of commissioning steps.

⚠ DANGER

Risk of injury due to movement of axes!

The commissioning results in a movement of axes.

- Make sure that neither you nor others are harmed by the movement, e.g. by maintaining a suitable safety distance.
- Do not perform any action whose consequences you cannot estimate

⚠ WARNING

Incorrect axis position during initial commissioning

Without referencing / calibrating the axis position, the displayed axis position may deviate from the actual axis position.

- Perform a homing to determine the correct actual position using a reference signal.

General safety precautions

Make sure that you know about the NC safety functions [▶ 10], and make use of them.

Handling the emergency stop case is one of the most important and safety-relevant features of a machine. For this reason it is usual for an additional hardware safety circuit to be built on top of all the logical (logical axis enables) and software enables (PLC: controller and feed enables). Extensive statutory regulations exist governing the reaction and the implementation in hardware of the emergency stop system. Here we shall only make reference to these.

Within the limits prescribed by the statutory regulations, there remains often a degree of free choice as to how, for instance, NC axes are stopped the in the event of an emergency stop. In some cases the mechanical properties of the machine mean that it is not always possible to bring axes that are moving to an abrupt halt (e.g. to bring an analog interface abruptly to 0.0 Volts). It is therefore quite common for NC axes to be slowed following special braking ramps in an emergency stop situation, after which they are then electrically and mechanically locked.

One of the first procedures for commissioning a machine is to test the emergency stop circuit. We of course begin with safe, standard situations (the axes are stationary under emergency stop conditions, although special attention must be paid here to axes that move vertically). The testing then moves on to more complicated cases (axes moving slowly, then axes moving rapidly).

It is possible, in order to test and commission rapidly, to set the logical axis enables (controller and feed enables) and the velocity override of the axes from the TwinCAT System Manager "Axis online" for the particular axis. This is, however, only possible if the axis interface is not involved in cyclical exchange with the PLC (mapping between the NC and PLC tasks). In that case, the single write of information from the System Manager would immediately be overwritten again by the cyclic data exchange. It must, however, be pointed out that this procedure can be very dangerous, because the usual safety monitoring and manipulation facilities of the PLC are disabled. This property can be used without risk on an axis test stand with endless axes. It is extremely inadvisable to use this facility on a real machine.
System Manager: Axis: Online: Set

Essential preconditions

The description below covers all the steps of commissioning an axis. Several types of axis and a range of different situations are covered. In each step, all the settings which must be made are mentioned, even when they are identical with the settings of the previous step. In practice, you will only actually carry out of some of the partial steps described for an axis.
If you already have experience of axis commissioning under TwinCAT, it will not always be necessary to carry out every step. In that case, you should use this description as a memory aid and for assistance in problem cases.
Essential preconditions:
Before you can begin the commissioning itself, a number of preparations must be made.

- Check their completeness and correctness of all the electrical connections.
- All parts of the axis (encoder, drive, controller, PLC interface) must be of appropriate types, and must be provided with the correct resources (PLC variables, I/O hardware in the fieldbus etc.).
- The NC architecture must not only be created, but must also be written into the registry, and TwinCAT must be started with it.

Recommended safety precautions

The following safety precautions should be taken if at all possible.

- Ensure that no one can enter or reach into the machine: the machine can behave unpredictably during commissioning.
- Isolate those parts of the equipment on which you are currently working. Anything that is not required at any particular time should be securely brought to a halt - it can only be a source of interference or distraction.
- Inform all personnel in the area that there is an increased danger of accident.
- Make sure that you will not be disturbed: nobody should be nearby without good reason.

Sequence

It is necessary to observe a particular sequence, because the individual steps of axis commissioning are logically built one on another. The sequence, however, does depend on the particular axis composition, i.e. on the combination of the encoder, controller and drive types.

Stepper motor drives

This section only concerns stepper motors.
Masks are used to send drive control signals to a stepper motor. These masks are packages of eight bits that correspond to the motor's control signals. The signals are sent out in one phase, and switch the

associated motor coils on. A particular mask is output when the motor is stationary, and has the purpose of switching on the holding current. The masks depend on the motor drive unit in use and on the wiring of the control cabinet, which means that a universally valid version cannot be given.
Find the bit combinations from the documentation provided by the drive manufacturer or by the electrical constructors, and enter them on the drive's stepper motor tab. If, during later commissioning steps, the axis behaves incorrectly, these masks a should be checked first.
The mask is entered in the form of a decimal number. In order to convert a bit pattern into the value that must be entered, the weighting must be determined for each bit that is set.

Table 3: Bit patterns

Bit	0	1	2	3	4	5	6	7
Weight	1	2	4	8	16	32	64	128

Example: In the bit pattern 00101011, bits 0, 1, 3 and 5 are set to "1". The associated weights are 1, 2, 4 and 32, so the total to be entered is 39.

6.2 Encoders and actual values

Determination and setting the most important encoder parameters: encoder polarity, scaling (increments per mm), position offset (zero offset shift).

Interface

The encoder parameters are accessed via the System Manager.

Determining actual values: determining the direction (encoder polarity)

Without correctly determining the actual value, an axis is "blind", and cannot behave in a coordinated manner. Commissioning, therefore, must begin here. This requires two items of information to be found or checked: the direction of the counting, and the value of the increment.

1. Determine or specify which direction of the axis movement is to be considered "positive", "rising", or "upwards". When moving in this direction, the actual value acquisition system must report rising values.
2. If you can disconnect the drive electrically and to move the axis manually, turn the drive in such a way that the machine, for instance, moves in the "positive" direction, and observe the display of the actual value in the online image of the axis.
3. If this is not possible, it will be necessary for you to turn the axis using the drive. This means that you will be starting operation of an axis that has not been parameterized. Special care is required because this procedure will disable all the controls and monitors. The axis will be operated with nothing but the feed forward and fictional data. Proceed as follows:
    - If it is possible to separate the load from the drive (e.g. by removing belts or other transmission media) without disconnecting the encoder at the same time, this is undoubtedly what you should do!
    - On the controller's global tab, set a value of 1.0 for the feed forward weighting.
    - If the controller has a PID tab, set the position control factor Kp (and all other control factors) to zero. Deactivate the automatic offset adjustment.
    - On the drive's global tab, adjust the output limit to +0.1 or -0.1.
    - If the drive has an analog tab, set the **reference speed** to a fictional value (not too small) and set the **reference output** to 1.0. Set the **drift compensation** to zero.
    - On the global tab for the axis, set the same value for the **maximum permitted speed** as for the drive's reference speed. Set one tenth of the same value under **manual speed max** and **manual speed min**. Deactivate all the end position monitors, following error monitors and other checks.
    - Make sure that there is nobody in the hazardous area.

- ◦ Using extreme care (with your hand by the emergency off switch) switch the axis on electrically. There should not be any drift movement.
- ◦ Now use the <F2> and <F3> keys in the online image of the axis to move in the "smaller" and "larger" directions, and observe the machine. If, when the <F2> key is pressed the machine moves in the "larger" direction, and when the <F3> key is pressed it moves in the "smaller" direction, alter the motor polarity on the drive's global tab, and repeat this step.
- ◦ Now compare the direction of the machine's movement with the counting direction of the actual position display in the online image.

4. If the counting direction is supposed to the specified movement direction, change the **inverse transducer counting direction (polarity)** parameter on the encoder's global tab. Then you should remove all current from the drive and reset the axis in the online image <F8>.

Determining actual values: increment evaluation (scaling)

After the drive's counting direction has been determined, it is necessary to set the value of the increment. This value is to be set in the **scaling factor** on the encoder's global tab.

1. The value can be calculated if the feed rate per rotation and the number of increments (not the number of impulses!) are known. Nevertheless, you should check the value as follows.
2. If you still do not know any value for the increment, at least set an estimated value. If you have no information at all about the value to be set, select a fictional value such as 0.01 mm.
3. If you are performing a check later, then of course you start with values that have been determined beforehand.
4. It is now necessary to travel over a known distance as a test movement (as in the determination of direction). The correct value is now calculated from the currently set scaling factor multiplied by the actual travel divided by the indicated travel.

Determining actual values: zero offset shift

This section is only to be carried out for **absolute encoders**.
In incremental systems at this point you should set the actual value to a reasonable figure (with the drive switched off and followed by a reset through <F8> in the online image for the axis). You will find the input boxes and function keys for this purpose on the function tab for the axis. Enter the value 0 for the zero correction. To adjust the axis measurement system to the machine, the calibration function of these encoders is used, the parameters for which will be determined at a later stage.

1. Determine the position of the axis within the machine. The rules for this cannot be generally stated and can only be specified by the machine constructor. In many cases, the distance of a reference point on the moving part of the machine to a reference point on the fixed part of the machine body is chosen.
2. If it is not possible to determine the position in this way with the necessary precision, make a preliminary adjustment, and repeat this step again later.
3. Find the difference between the determined position and the displayed position and enter this with the opposite arithmetic sign in the **zero-offset shift** of the encoder's global tab. If a value has already been entered there, modify the existing value by adding the new value to it. If (when repeating this step later) the position control is already active, you should electrically halt the drive and reset the axis before switching on again <F8>.
4. Example: The position is determined to be 752.0 mm, while the displayed position is 1983.52 mm. The difference, in other words, is 1231.52 mm. If no value has yet been entered in the zero-offset shift, you can simply enter -1231.52. If a zero offset has already been entered, add the new value to it. In this example, an existing value of +150.0 would yield the new value -1081.52 mm.

Further encoder parameters

The encoders offer a range of working modes for optional setting. These include, for instance, determination of the actual speed or of the actual acceleration. If necessary, these measurements can be activated on the encoder's global tab. If these values are not interesting, they should be specifically deselected, since determination of the values increases the real-time loading.
It is also possible to set a modulo factor. This is of particular interest for rotating axes, since it allows the relative position within one rotation and the number of rotations to be found automatically.

Finally, the encoder values (position, speed, acceleration) can be filtered, which is of particular significance for encoder axes.

⚠ WARNING

Risk of injury and damage due to incorrect behavior of the controller

Filtering the position, even if small values are used, can result in disastrously incorrect controller behavior. Always proceed carefully with the settings.

6.3 Drive and voltage output

⚠ WARNING

Unpredictable behavior of the axis under special circumstances

Inform yourself about the behavior of the drive in case of overload. There are drives that automatically shut down when overloaded, so that the axle could then be force-free, full of energy and no longer braking.

Determination and setting the most important drive parameters: drive polarity, reference output, reference velocity, drift compensation.

Interface

The drive parameters are accessed via the System Manager.

The drive parameter: determination of the direction

1. Use the encoder! Restrict the distance of travel using the software limit position min / max on the global tab for the axis.
2. On the drive's global tab, adjust the output limit to *+1.0* or *-1.0*.
3. On the controller's global tab, set a value of *1.0* for the feed forward weighting.
4. If the controller has a PID tab, set the position control factor Kv (and all other control factors) to zero. Deactivate the automatic offset adjustment.
5. If the drive has an analog tab, set the *reference speed* in accordance with the information provided by the machine manufacturer to something in the region of the correct value, and set the *reference output* to *1.0*. Set the *drift compensation* to zero.
6. On the global tab for the axis, set the same value for the *maximum permitted speed* as for the drive's *reference speed*. Set one tenth of this value under *manual speed max* and *manual speed min*.
7. If the controller has a PID tab, set the position control factor Kv (and all other control factors) to zero. Deactivate the automatic offset adjustment.
8. Now use the <F2> and <F3> keys in the online image of the axis to move in the "smaller" and "larger" directions, and observe the machine. If, when the <F2> key is pressed the machine moves in the "larger" direction, and when the <F3> key is pressed it moves in the "smaller" direction, alter the motor polarity on the drive's global tab, and repeat this step.

The drive parameter: feed forward

This section is only necessary if the drive has an analog tab.

1. On the controller's global tab, set a value of *1.0* for the feed forward weighting.
2. On the drive's global tab, adjust the output limit to *+1.0* or *-1.0*.
3. On the controller's PID tab, set the position control factor Kv to a low value. When correctly set, it will be easy to detect the following error during the test transits that are to be made out, but it will be constant. Example: 1 (mm/sec)/mm or 60 (mm/min)/mm.
4. On the drive's analog tab, set the *reference speed* to an approximately correct value in accordance with the information provided by the machine manufacturer. Set the *drift compensation* to zero. In accordance with the information provided by the machine manufacturer, set the *reference output* to 1.0 if this refers to 100% (or 10 Volts). If the figure provided is, for example, for 90% (or 9 Volts), 0.9 should correspondingly be set.
5. On the global tab for the axis, set the *maximum permitted speed* to about 90% of the drive's *reference speed*. Set one tenth of this value under *manual speed max* and *manual speed min*.

6. On the dynamic tab of the axis set a provisional value of about 200 mm/s² for the **acceleration** and **deceleration** and a **jerk** of 500 mm/s³. If these values do not allow the constant speed to be reached within the space available, increase the values while retaining the relationship.
7. Now use the <F2> and <F3> keys in the online image of the axis to start in the "smaller" and "larger" directions, and observe the arithmetic sign of the total output and the controller output.
8. If the two values have the same sign, the axis is reacting with a lower speed than that expected by the feed forward. In this case, the *reference speed* should be reduced. If the arithmetic signs are opposed, the axis is moving faster than expected by the feed forward. In that case, the *reference speed* should be increased.
9. If the controller output is significantly less than 10% of the total output, the *maximum manual speed* can be increased in steps up to about 50% of the maximum speed. The test movements should be continued using the <F1> and <F4> keys.
10. If the feed forward is well adjusted, the controller output should represent less than 1% of the total output. Poor mechanical conditions, however, often do not permit such accurate adjustment.

Drive parameter: Drift compensation

This adjustment is only possible if the drive has an analog tab.

If the set values are provided in analog form, there can often be a voltage difference between the controller and the power section. This can be the result of a real voltage drop. However the cause is nearly always that the zero adjustment of the two sides is not the same. In this case, when the position control is switched off, the axis will slowly leave its position: it drifts. Active position control will stop it not far from the set position. Nevertheless there is then a certain amount of following error.

It is possible in this case to compensate for the offset by means of a constant speed offset. However the DAC resolution of a few mV has a negative effect here.

As an alternative, some controllers offer a facility for automatic offset correction.

The drive parameter: output limiting

It is necessary in some cases to limit the control level output given with a drive to the power section. You will find appropriate input facilities on the drive's global tab. You should, however, only make these adjustments at the end of the axis commissioning. Until that time, the values of –1.0 and +1.0 should be entered for the minimum and maximum limits. Otherwise, the measurements described here will be affected.

6.4 Position control and reduction of the following error

Determination and setting the most important position control parameters: Kv factor and parameters for reduction of the following error.

Interface

The position controller parameters are accessed via the System Manager.

Weighting of the feed forward

Axes are usually operated under TwinCAT with a **feed forward weighting** of *1.0*. In unusual cases it may be necessary to vary this setting. This can be the case if the axis is subject to a highly non-linear relationship between control and velocity. A hydraulic valve whose characteristic curve has a sharp knee would be an example. In such a case the weighting should be reduced until the axis no longer demonstrates excessive advance. In these cases it is generally necessary to use more complex controller types including an I-component in order to achieve acceptable positioning behaviour.

Position control

The purpose of the position control is to compensate for small residual errors in the positioning. The possible causes of these residual errors can consist of load changes, non-linearities originating with friction or other causes, small inaccuracies in a range of axis parameters, offsets, temperature dependencies and so forth.

Position P-controller

In this controller, a compromise is to be found between the desire for the highest possible level of control (i. e. a large Kv) and a low tendency for the system to oscillate (i.e. a small Kv). In practice it has been found that this compromise can be found quickly by increasing the **Kv value** step-by-step.

The procedure is extremely simple: test transits are made in both directions. During the constant velocity part of the movement, the axis should move steadily without detectable oscillation being created by the position control. A small proportion of noise in the controller output is not significant here. As it approaches the target, the axis must under no circumstances show any signs of oscillation (springing back). The Kv factor can be increased in small steps (with a factor of between *1.2* and *1.5*) until the axis shows the first signs of oscillation. The Kv factor should then be reduced by about *20*%, so that there is some safety margin to the stability limit.

As has already been explained, some controllers offer the facility for **automatic offset adjustment**. This function can be activated on the controller's PID tab. The **filter time** and the **feed forward limit** are to be provided as parameters. A value in the range of seconds is to be recommended as a rule for the filter time. It must always be borne in mind that the filter time should be very large in comparison to the response time of the axis. If it is too short, a phase-shift oscillator is created: the axis then oscillates at a very low frequency. The feed forward limit prevents the automatic adjustment during active positioning from being affected by the dynamic behaviour of the axis. A set velocity above this value has the effect of "freezing" the offset.

Position 2P-controller

The only difference between this controller and the one mentioned above is that it is possible for a different Kv value to be used during active operation from the value used for position control when stationary.
A **second Kv value** and a **velocity threshold** are to be set here as additional parameters. The effect is a smooth transition between the Kv values over the range of velocities between a *0.0* (stationary) and the threshold value.

Reducing the lag error

It is essential to use the TwinCAT Scope to minimize the lag error.

1. Minimize the lag error for position control. Determine the mean lag error d [mm] when stationary under position control (with sign) (online menu). Set the DAC offset to $Offset = d \cdot Kv$ [mm/s] (If the lag error is positive, the actual value lags behind the setpoint and needs somewhat more voltage to catch-up). This formula only applies if the motor polarity and the encoder polarity are in agreement, otherwise it is necessary to take the value with the opposite arithmetic sign. This should allow the lag error to be set to an increment precisely symmetrical about 0.0.
2. Minimize the lag error at constant set velocity. The set velocity V_S, reference velocity V_{ref}, proportionality constant Kv and the lag error that has been found d [mm] are given. The output is then to be the output velocity proportional to the voltage $V = (V_S + Kv \times d) / V\_ref$. To avoid the lag error, the same velocity should now be output with the new reference velocity V_r , so that $V = V\_S / V\_r$ applies. We therefore get V_r=*Vref/(1.0+1.0/(Kv×d))* for the new reference velocity.

Position PP-controller

If a position P-controller is correctly adjusted, the lag error is, in theory, proportional to the acceleration. In order to minimize the lag error in the acceleration and braking phases, it is possible to use a second proportionality factor *Ka* in association with the set acceleration and, with the aid of which the lag error under ideal conditions comes only to consist of noise. The factor *Ka* is approximately equal to the system transfer time, and so is in a range that is less than *10 × T_saf* s (*T_saf* set execution task cycle time).

6.5 Axis parameters, axis monitoring and axis calibration

Determination and setting the most important axis parameters: maximum velocity, following error monitoring, end position monitoring and those for dynamic behaviour and calibration.

Interface

The axis parameters are accessed via the System Manager.

General axis parameters

A variety of further parameters can be adjusted on the global tab for the axis. These determine the behaviour of the axis. A range of monitoring functions and status signals are also controlled here.

You can specify the level to which all commands sent to the axis can be limited with the **maximum velocity**. You should set a value here of about *90* to *95*% of the maximum velocity of the drive. The remaining control capacity is required for the controller in order to ensure correct positioning.

When an NC interpreter is used, certain commands will cause the axis to be positioned using an adjustable control for the **high-velocity travel**. Not more than about 90% of the maximum velocity of the axis should be chosen here.

The **min and max manual operation velocities** can be freely chosen, and largely depend on the ”confidence” of the user. In practice it has been found appropriate to set these velocities with a ratio of about *1* to *5*. The values must, of course, be smaller than the maximum velocity.

The **reference travel velocity in the positive direction** is only effective on axes with incremental encoders. This is the velocity with which the axis moves when looking for the calibration cam. The direction is set on the encoder's incremental tab.

The **reference travel velocity in the negative direction** is only effective on axes with incremental encoders. This is the velocity with which the axis moves when leaving the calibration cam and when looking for the synchronisation pulse. The direction is set on the encoder's incremental tab.

The **positive / negative direction pulse widths** specify the distances to be travelled by the PLC when the axis starts in the corresponding movement modes. Fuller information is found in the PLC's function block descriptions.

You can activate and set the position of a movement limit (**software limit switch**) both below and above.

The **following error** (the difference between the set and actual positions) is continuously determined for axes with an analog tab (i. e. not for high/low-speed or for stepper motor axes). An adjustable threshold for monitoring these values can be activated. It is possible here to specify a **filter time** which may prevent an alarm being signaled if the threshold is briefly exceeded, for instance, during acceleration or braking.

Two independent windows can be specified around the target position of a movement (**target position windows**). Any size can be chosen for the windows. This makes it possible to create a signal for the PLC at any desired distance from the target. One of these windows also offers a filter time. The signal is only generated if the axis remains continuously within the window for whatever period is set there.

It is possible for high/low speed axes to approach each destination always from the same direction, and having covered a minimum travel distance. This allows a greater precision to be achieved if the axis does not behave symmetrically, or if there is mechanical play. This behavior can be activated, and the travel distance can be specified. The arithmetic sign of the specified distance determines the direction.

In servo axes "**reverse backlash compensation**" can be used to compensate for mechanical play in the axes. The value entered here as the "reverse backlash" specifies the distance by which a target is exceeded. The arithmetic sign of the value specifies the direction of the overshoot. If the reverse backlash is positive, the target is only overshot when moving in a positive direction. Movement in the direction of smaller positions is not affected. Negative reverse backlash results in the opposite behavior.

Axis dynamics

On the dynamic tab for an axis, values for the **acceleration**, **deceleration** and **jerk** are to be set. Two methods are available: direct entry as a numerical value, or indirect specification via the run-up time and the characteristic acceleration curve.

The values to be set here are the parameters to be observed by the set value generator. Drive manufacturers often mention values that would result in very hard axis positioning behavior if they were to be set here. This is because the values mentioned represent the limiting value of the motor's or of the power section. Usually, however, the machine's behavior would then be very dependent on the load. The result of this is considerable variation in the following error, which causes the position controller to become active. However, the axis is hardly able to react to the controller output, because it is already operating at the limit of its current.

The "correct" values for an axis depend heavily on the desired behavior and on the properties of the drive technology and of the machine. They can only be determined through test runs. Values should be tested as they are increased step by step, observing the following error when starting and stopping. It has been found in practice that the numerical value for the jerk should be somewhere between twice (for processing axes) to ten times (for transfer axes) greater. For indirect adjustment, these correspond respectively to soft and hard adjustments.

Use the values for the acceleration, deceleration and jerk from TwinCAT Scope for the adjustment. Whether the values for the acceleration *A+* or deceleration *A-* are really achieved depends on the **critical jerk***J±*, and on the ratio $(A+)^2/V$ or $(A-)^2/V$, where *V* represents the set velocity. Check, in particular, whether the actual acceleration value can follow the set acceleration value (and the same for the deceleration).

The effects of incorrect cycle time settings

Two errors result from incorrect cycle time settings, i.e. a difference between the set cycle time (NC task; SEC task; online; cycle ticks in ms) and the real cycle time taken by the SEC task:

- The set value generator information (position, velocity, acceleration) does not match the real cycle time, i.e. the velocity pre-control is incorrect.
- The position setpoints of the position controller and the set velocity values of the velocity pre-control are inconsistent.

Velocity pre-control: if the real cycle time *TC_eff* is greater than the set cycle time *TC_saf* the time axis becomes stretched out (scaling *TC_eff/TC_saf*), and this implies a non-linear transformation of the setpoints. Every calculated set velocity is maintained (*TC_eff/TC_saf*) times longer than initially calculated. The integrated effective target velocity reaches a target position which is (*TC_eff/TC_saf*) times larger than the calculated one.
Position control: since the target velocity and the set position do not match ($v(t) \neq dp(t)/dt$), the position controller must work against the pre-control. The resulting behavior depends on the proportionality factor, the size of the cycle time error setting, and the controller resources (reference velocity). In any event there will be a lag error in those phases where a force is present, and large overshoots.

Delay between pre-control and position control

The purpose of the delay generator is to compensate for the effects of a system-related delay time (hardware/communication) in the P part of an position controller between the output of the set velocity value and its effect on the positioning system. Instead of the output velocity *v_o(t)* as a function of the velocity pre-control *v_g(t)* and of the P part *K_v (p_g(t) - p_i(t))*, where *p_g(t)* is the position setpoint and *p_i(t)* is the actual position value,
v_o(t) = v_g(t) + K_v (p_g(t) - p_i(t)) + ...
the position setpoint is output subject to a delay period of delta, so that
v_o(t) = v_g(t) + K_v (p_g(t-delta) - p_i(t))+ ...

Parameter: delta delay time in s >= 0.0

Parameters for axis calibration

These settings are only necessary for axes with **incremental encoders**. So that the actual value system in such axes can be adjusted to the machine's reference system, a calibration travel is started. A number of parameters and signals are required for this.

When the **calibration travel** of an axis is started, it begins with the **velocity reference travel in a positive direction** (the global tab for the axis) mentioned above. The actual direction is specified by the encoder's incremental tab. The axis stops when the signal bit for the calibration cam is set in its NC-PLC interface. This cam allows a specific pulse to be selected on rotating encoder systems with one synchronizing pulse per rotation.

The axis now starts again to leave the cam. It now executes the **velocity reference travel in the negative direction** (global tab for the axis) until the signal bit is cleared. Without changing its velocity, the axis now continues until the encoder's synchronous latch has asserted. Only now does the axis stop.

The distance covered between the assertion of the latch and the location at which the axis of comes to a halt is determined. The total formed by adding this distance to the **reference position** (incremental tab of the in encoder) is set as the new actual axis position. This procedure allows the synchronous pulse from a selected rotation of the axis to be assigned to an absolute position in the machine.

The calibration is started by means of <F9> under Axis Online in the System Manager .

7 Samples

The following list presents a survey of existing TwinCAT NC samples:

• **PTP**: Positioning of an independent point to point axis • **Axis Coupling**: Ordinary Master/Slave coupling (electronic gearbox) • **Superposition**: Overlayed motion (Superposition) • **Camshaft**: Coupling of a slave to a master axis with a position table (mono-table coupling).	TcMC2
• **Flying Saw:** Positionprecise synchronisation of a slave axis to a moving master axis.	TS5055 | TwinCAT NC Flying Saw
• **FIFO-Axes**: Moving of axes with position data, which are constantly topped up by the PLC.	TS5060 | TwinCAT NC FIFO Axes
• **External Set Point Generator**: Moving of an axis using a PLC set point generator. • **Drive Controlled Homing**: Drive controlled homing with a SERCOS drive	TX1250 | TwinCAT NC PTP
• **Interpolation**: Interpolated path movements with an NC program.	TX1260 | TwinCAT NC I

All the samples are available in english.

More Information:
www.beckhoff.com/automation

Beckhoff Automation GmbH & Co. KG
Hülshorstweg 20
33415 Verl
Germany
Phone: +49 5246 9630
info@beckhoff.com
www.beckhoff.com