

BECKHOFF New Automation Technology

Manual | EN

TX1250

TwinCAT | NC PTP



TwinCAT 2 | NC PTP

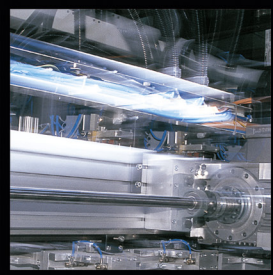


Table of contents

1 Foreword	5
1.1 Notes on the documentation	5
1.2 Safety instructions	6
1.3 Notes on information security.....	7
2 TwinCAT NC Master Axes	8
2.1 Axis start	9
2.2 Override and Stop	16
2.3 New end position.....	18
2.4 New velocity and new end position	19
2.5 Position compensation	20
2.6 External set value generation.....	23
3 TwinCAT Slave Axes	26
3.1 NC Error Handling	26
3.2 Linear Slave Axes	28
3.2.1 Interfaces	29
3.2.2 Coupling and coupling factor.....	29
3.2.3 Uncoupling and new end position	32
3.2.4 Position compensation	35
3.3 Cyclic Slave Axes.....	38
4 Examples	40
4.1 External Set value generator.....	40
4.2 Drive Controlled Homing (SERCOS)	40
4.3 Move Axis.....	41

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

The documentation has been prepared with care. The products described are, however, constantly under development.

We reserve the right to revise and change the documentation at any time and without prior announcement. No claims for the modification of products that have already been supplied may be made on the basis of the data, diagrams and descriptions in this documentation.

Trademarks

Beckhoff®, TwinCAT®, TwinCAT/BSD®, TC/BSD®, EtherCAT®, EtherCAT G®, EtherCAT G10®, EtherCAT P®, Safety over EtherCAT®, TwinSAFE®, XFC®, XTS® and XPlanar® are registered trademarks of and licensed by Beckhoff Automation GmbH.

Other designations used in this publication may be trademarks whose use by third parties for their own purposes could violate the rights of the owners.

Patent Pending

The EtherCAT Technology is covered, including but not limited to the following patent applications and patents:

EP1590927, EP1789857, EP1456722, EP2137893, DE102015105702
with corresponding applications or registrations in various other countries.



EtherCAT® is a registered trademark and patented technology, licensed by Beckhoff Automation GmbH, Germany

Copyright

© Beckhoff Automation GmbH & Co. KG, Germany.

The reproduction, distribution and utilization of this document as well as the communication of its contents to others without express authorization are prohibited.

Offenders will be held liable for the payment of damages. All rights reserved in the event of the grant of a patent, utility model or design.

1.2 Safety instructions

Safety regulations

Please note the following safety instructions and explanations!
Product-specific safety instructions can be found on following pages or in the areas mounting, wiring, commissioning etc.

Exclusion of liability

All the components are supplied in particular hardware and software configurations 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 engineering who are familiar with the applicable national standards.

Description of symbols

In this documentation the following symbols are used with an accompanying safety instruction or note. The safety instructions must be read carefully and followed without fail!

DANGER

Serious risk of injury!

Failure to follow the safety instructions associated with this symbol directly endangers the life and health of persons.

WARNING

Risk of injury!

Failure to follow the safety instructions associated with this symbol endangers the life and health of persons.

CAUTION

Personal injuries!

Failure to follow the safety instructions associated with this symbol can lead to injuries to persons.

NOTE

Damage to the environment or devices

Failure to follow the instructions associated with this symbol can lead to damage to the environment or equipment.



Tip or pointer

This symbol indicates information that contributes to better understanding.

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 TwinCAT NC Master Axes

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 co-ordinate 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 advanced TwinCAT NC functionalities are achieved on the basis of the PTP modes by reconfiguration (FIFO, NCI) or by coupling (all slave types).

Axis start

At the [axis start \[► 9\]](#), a jerk-limited dynamic profile is generated via the global **dynamic parameters** (acceleration, deceleration, jerk) and the start parameters (set velocity and target position), and stored in form of a table (run time table), which is then used for the generation of the set values within the block execution task.

Override and Stop

The [override \[► 16\]](#) is used to **change the target velocity** of the axis online in the quickest possible way. The request 'Override on x%' causes a new runtime table to be calculated and activated, if the axis was started and no stop is active and the brake phase has not yet been reached. In case of failure, the function is not called again automatically.

Stop is realized via override *0.0* and has - provided the brake phase has not yet been reached - priority over all other functions.

New end position

The function '[New target position \[► 18\]](#)' is used to **change the target position** of the axis online in the quickest possible way. If the axis was started and no stop is active, the request 'new target position' causes a query to ascertain whether

- the axis has exceeded the new target or whether
- the axis cannot be stopped before the new target.

In these cases, the axis is stopped internally via a stop and must automatically be moved back to the new target via a new start. Otherwise a new runtime table is calculated and activated.

New velocity and new end position

The function '[New velocity and new end position \[► 19\]](#)' is used for **instantly changing the target position** and - after an activation position *APos* has been crossed - **for (later) changing the target velocity** of the axis. The specified velocity *NewV* in *mm/s* must be genuinely larger than *0.0*, and the new end position *EPos* must be within the software limit switches. The function 'New velocity and new end position' cannot be activated during position compensation.

Position compensation

Sometimes it is desirable for an axis to carry out **additional positioning** by a preset distance (compensation difference *D* - may also be negative) within a certain distance (compensation length *L*) compared with the positioning resulting from the runtime table. This procedure is usually used for spatial synchronization with other axes and can be activated on any axis (master, slave).

If position compensation cannot be achieved with the specified compensation parameters, the maximum possible compensation is automatically carried out.

[Position compensation \[► 20\]](#) may be aborted via a compensation stop (continuous).

2.1 Axis start

At the axis start, a jerk-limited dynamic profile is generated via the standard **dynamic parameters** (acceleration, deceleration, jerk) and the start parameters (set velocity and target position), and stored in form of a table (run time table), which is then used for the generation of the set values within the block execution task.

Units and reference systems

Basic and pre-set units are millimeter [mm] and second [s], although the position can also be displayed in modulo values. Here, the units used are **mm and s**.

The axis can be started from its current position absolute, relative or modulo (in relation to a given modulo factor) to a target position.

The direction of travel of the axis is determined on the basis of the sign of the **modulo position**. If a modulo position smaller than the modulo factor is specified, (default value 360.0) the shortest path to that modulo position will be taken, with the specified direction being maintained. This end position may still be in the same or in the next modulo period (depending on the start position). In case of modulo position 0.0, the sign cannot be used to determine the direction of travel. By definition, this case is interpreted as positive direction of travel. To approach the position 0.0 in negative direction, a negative value close to 0, e.g. -1e-6 can be specified.

If a modulo position is specified as being greater than or equal to the modulo factor, the target position is no longer reached by the shortest route, but only after the modulo position has been passed over once or several times. As always, the sign defines the direction of travel and the position value divided by the modulo factor defines the number of additional revolutions until subsequently stopping at the corresponding orientation. An example: modulo start at position +540.0 degrees with modulo factor 360.0 means that the axis is moved at least one period ($360.0 + 180.0 = 540.0$) in positive direction and then is stopped on the shortest way oriented at the modulo position 180.0 degrees.

Here, only **absolute positioning** is considered.

In the following $a(t)$ is used for acceleration, $v(t)$ for velocity and $p(t)$ for position at time t after start of axis.

In this context accelerations/decelerations and velocities almost refer to **path accelerations/decelerations** (path acceleration/deceleration = $\text{signum}(\text{axis target velocity}) \times \text{axis acceleration/deceleration}$) and **path velocities** (path velocity = magnitude of axis velocity).

Global dynamic parameters

- **Acceleration** $A+ > 0$: the parameter $A+$ limits the acceleration upwards, i.e. for the entire positioning applies: $a(t) \leq A+$.
- **Deceleration** $A- > 0$: the parameter $A-$ limits the acceleration downwards, i.e. for the entire positioning applies: $-A- \leq a(t)$.
- **Jerk unit (change in acceleration)** J : the jerk unit controls the increase and reduction in acceleration/deceleration. If $a(t)$ is not constant, $|da(t)/dt| = J$, i.e. the change in acceleration will always occur with the maximum permissible value.
- **Acceleration limits** $A+$, $A-$. Whether the acceleration/deceleration limit(s) ($A+$, $A-$) are reached depends on the relationship between acceleration limit/deceleration limit and target velocity V . The value of the jerk J at which the acceleration/deceleration limit is just reached (but not maintained) is the **critical jerk** J_{\pm} , which is defined by the equation $J+ = (A+)^2/V$, or $J- = (A-)^2/V$ in units mm and s.

Start parameters

- Set velocity $V > 0$: the parameter V defines the upper velocity limit, i.e. the following applies for the complete positioning: $|v(t)| \leq V$.
- Target position.
- The length the positioning path indirectly results from the start position (actual position) and the target position.

Velocity profile and acceleration profile

The standard dynamics for servo axes is a jerk-limited 7-phase dynamics. The 7 phases are (optional phases are shown in brackets [])

1. Acceleration phases $a(t) > 0$:
 - Acceleration increase: $da(t)/dt = +J$,
 - [Constant acceleration: $a(t) = A+$],
 - Acceleration decrease: $da(t)/dt = -J$,
2. [Constant velocity phase $|v(t)| = V$],
3. Deceleration phases $a(t) < 0$:
 - Deceleration increase: $da(t)/dt = -J$,
 - [Constant deceleration: $a(t) = -A-$],
 - Deceleration decrease: $da(t)/dt = +J$.

General notes and special cases

- The start occurs from the current actual position.
- A start does not occur, if the length of the positioning path is less than an encoder increment (Warning).
- If the set velocity V cannot be reached (the distance is too short or V is too large), the largest achievable velocity (under the symmetric acceleration/deceleration $A_{\pm} = \min(A+, A-)$) is determined and activated instead of V without notification.
- If no dynamic profile can be generated with the specified start data, because the jerk is too large, the jerk is automatically rescaled.
- An override is available via the PLC that changes the set velocity to a percentage between 0.0 and 100.0 . One can, for example, start the axis with override 0.0 , and then increase the override as required.

Procedure

1. Prior to a start request: the Set PreParation task is awaiting new tasks, and in the Set ExeCution task only position control is operating.
2. **Set PreParation task:** once the start parameters have been checked, a dynamic profile is generated and stored in a runtime table. The setpoint generation of the Set ExeCution task is than **automatically** enabled.

Set ExeCution task: the setpoint generator calculates the current setpoints based on the runtime table and the expired time since the start and transfers them to the position controller. Setpoints for the setpoint generation are the local setpoints, i.e. the following parameters calculated during every cycle:

 - the position setpoint (for position control),
 - the velocity command value (for velocity pre-control),
 - the acceleration setpoint (for control with special controllers),
 - the set direction (for direction-independent feed monitoring).
3. After the end of path: in the Set ExeCution task, once again only the position controller is executed.

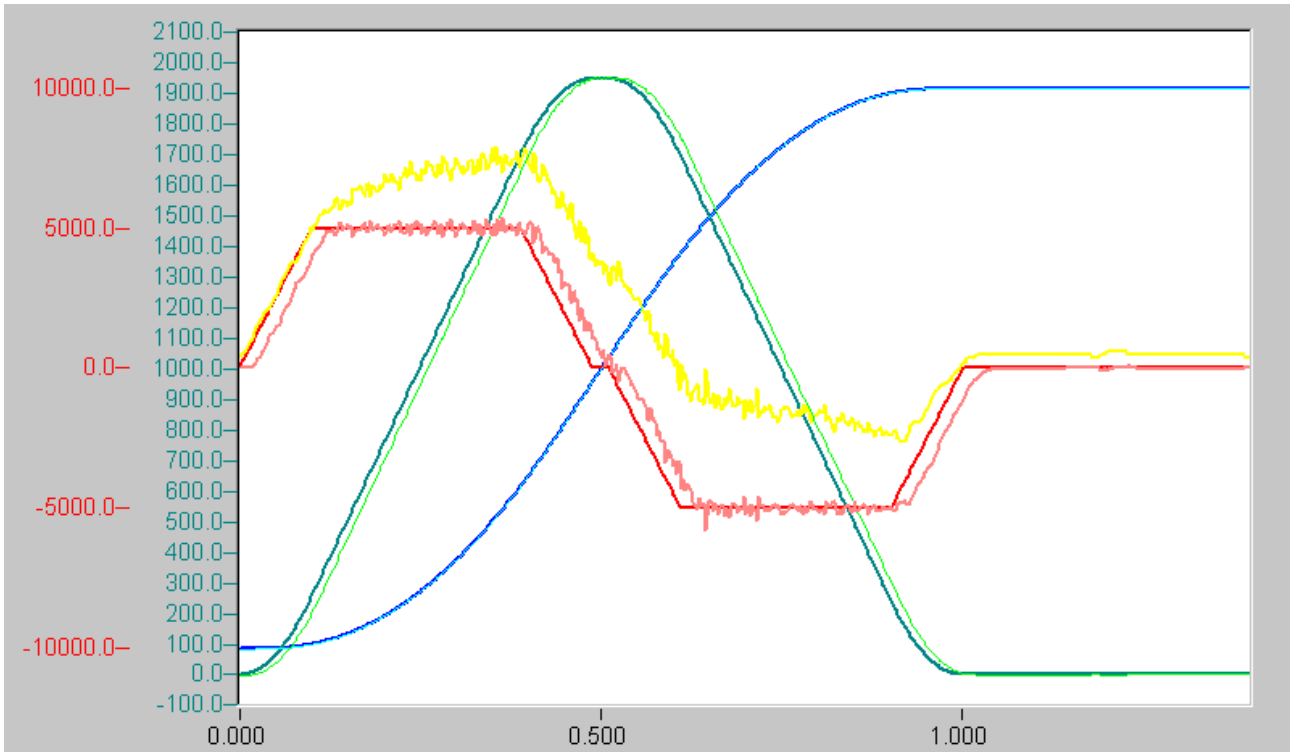
Notes on the figures:

Scales: velocity [mm/s] / time [s]

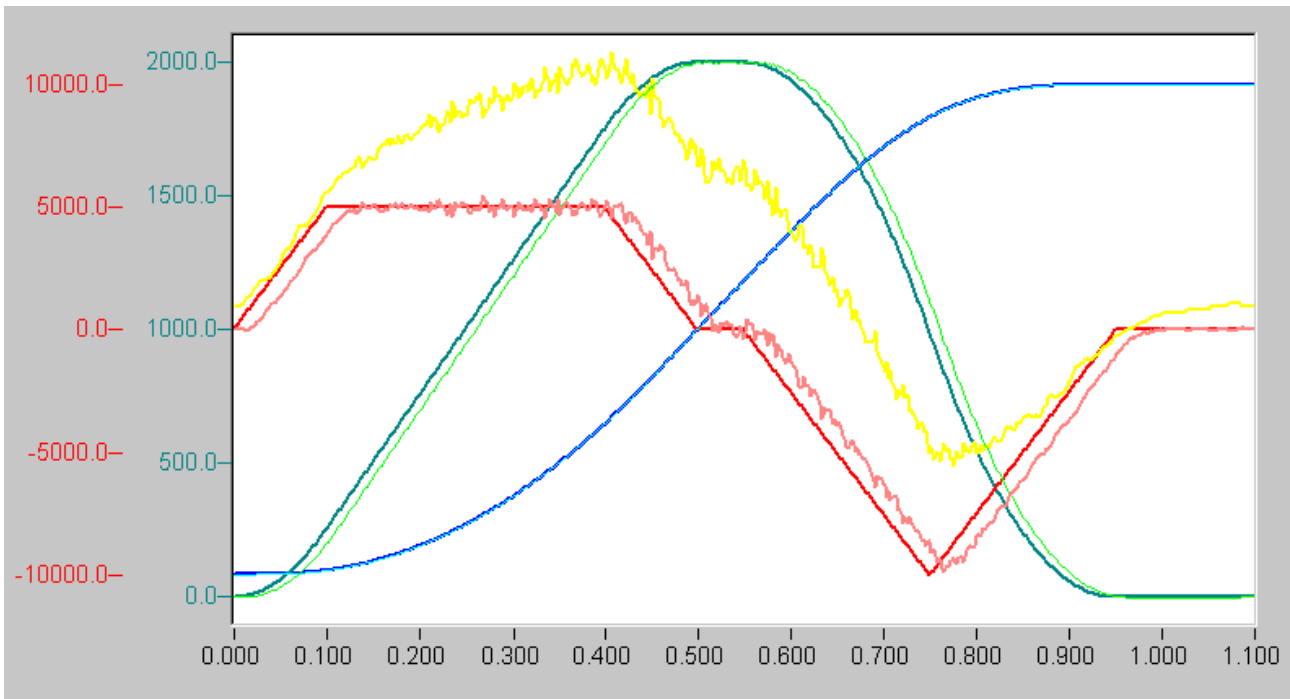
Colors:

Set position: dark blue	Actual position: light blue
Target velocity: dark green	Actual velocity: light green
Set acceleration: dark red	Actual acceleration: light red
Following error: yellow	

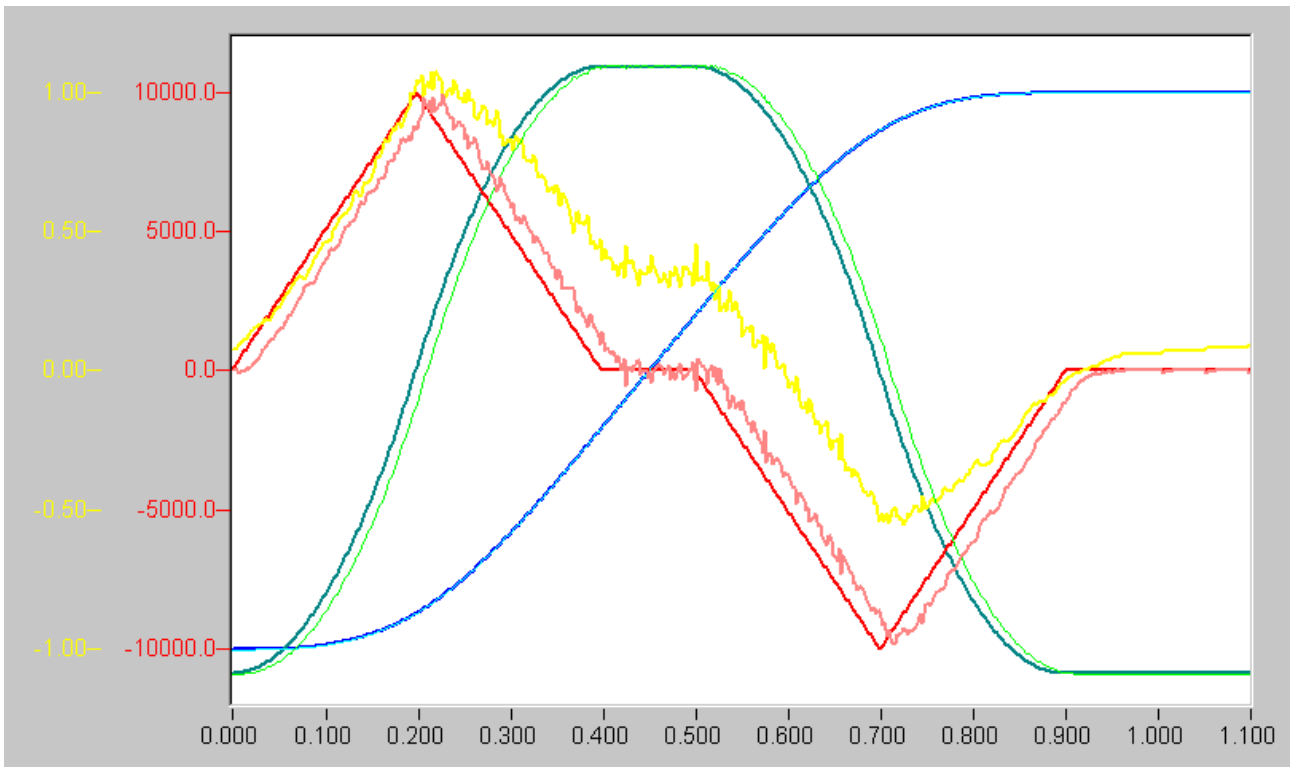
Dynamics I: acceleration/deceleration



Dynamics 1.1 Dynamic profile with $A+ = A- = 5000.0$ mm/s/s, $R = 50000.0$ mm/s/s, $V = 2000.0$ mm/s from 0.0 mm to 1000.0 mm

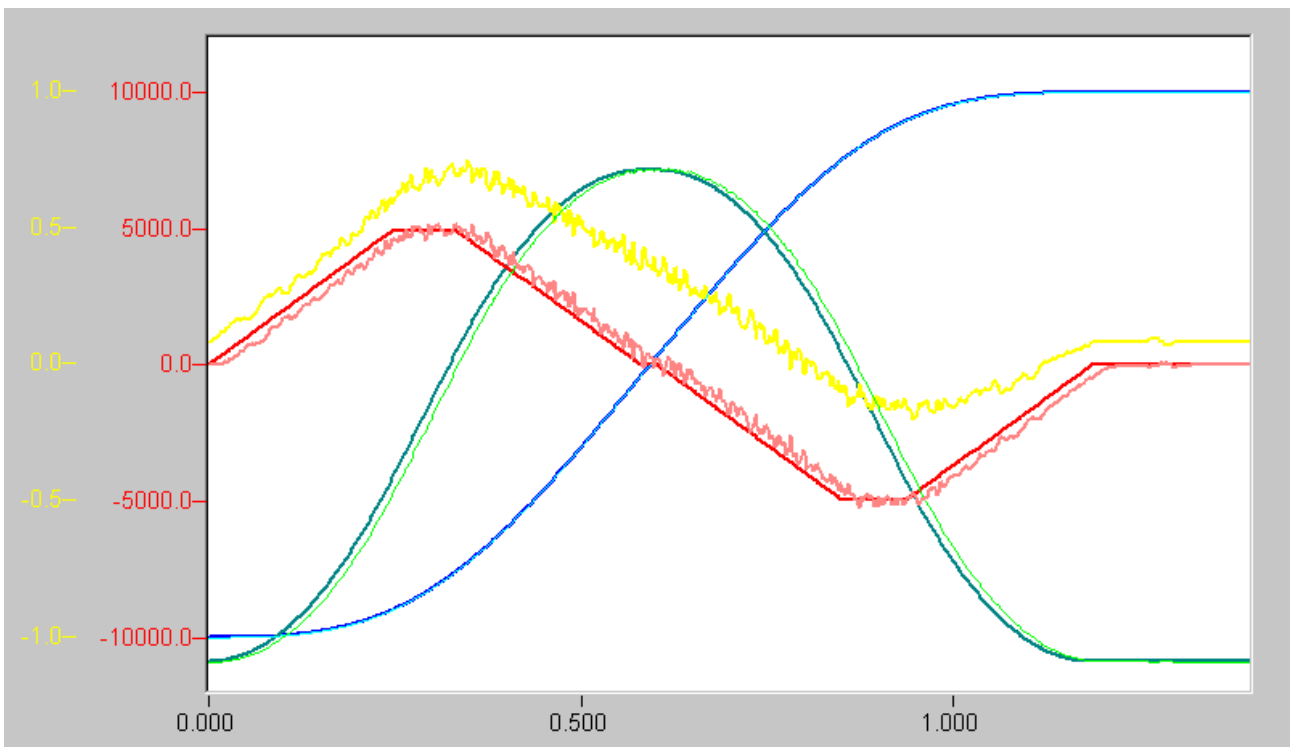


Dynamics 1.2 Dynamic profile with $A+ = -5000.0$, $A- = 10000.0$ mm/s/s, $R = 50000.0$ mm/s/s, $V = 2000.0$ mm/s from 0.0 mm to 1000.0 mm

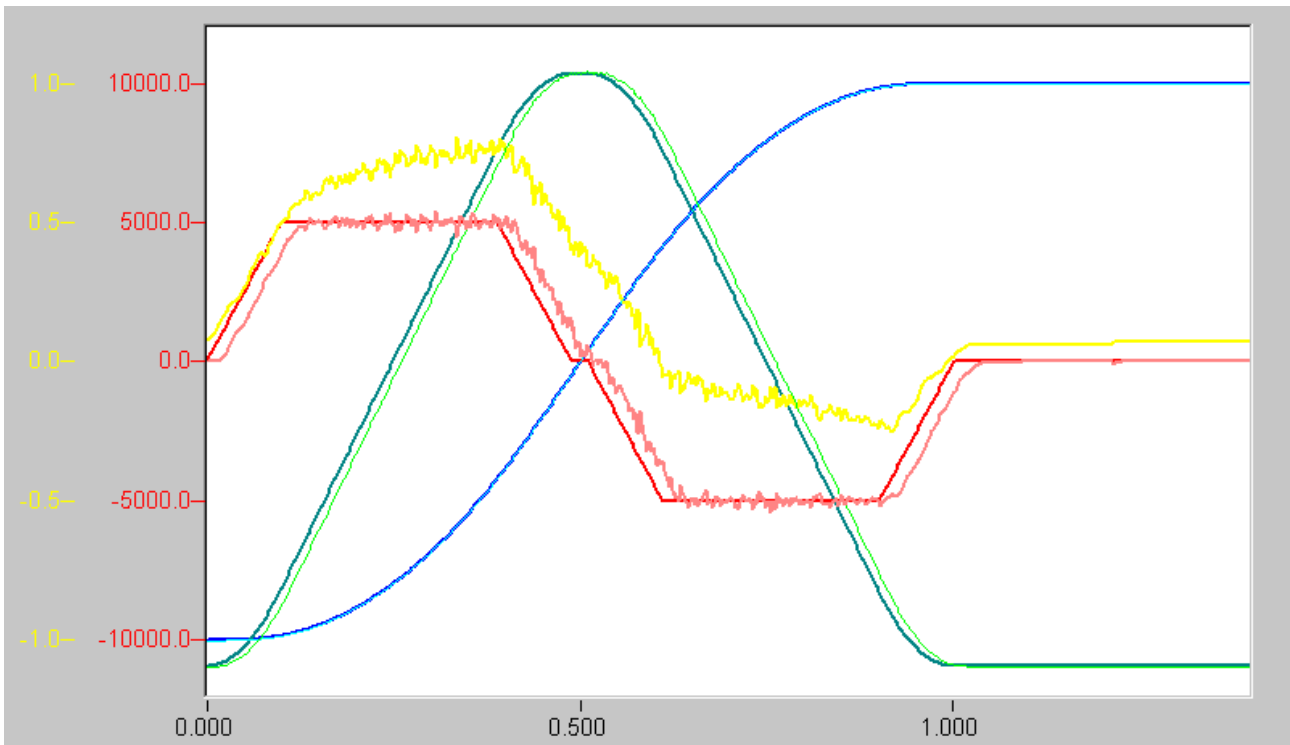


Dynamics 1.3 Dynamic profile with $A+ = A- = 15000.0$ mm/s/s, $R = 50000.0$ mm/s/s, $V = 2000.0$ mm/s from 0.0 mm to 1000.0 mm

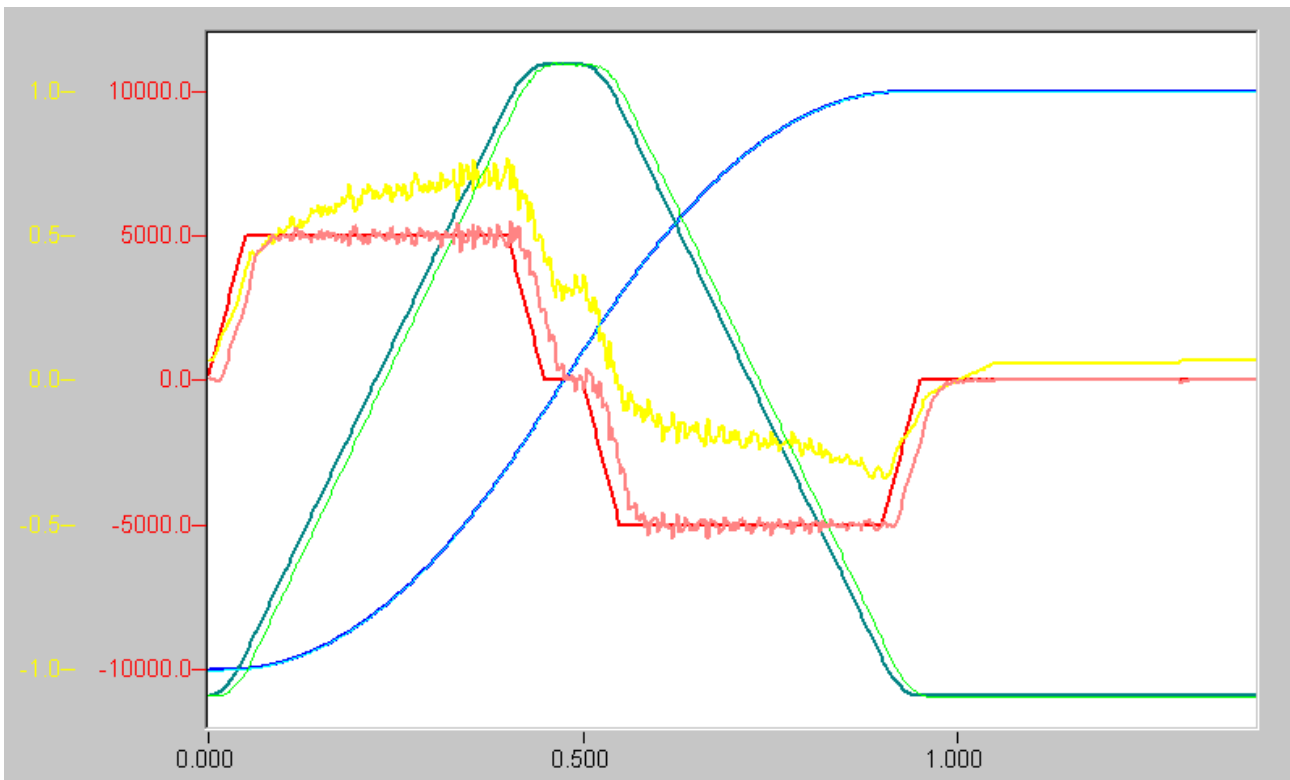
Dynamics II: jerk (change in acceleration)



Dynamics 2.1 Dynamic profile with $A+ = A- = 5000.0$ mm/s/s, $R = 20000.0$ mm/s/s, $V = 2000.0$ mm/s from 0.0 mm to 1000.0 mm

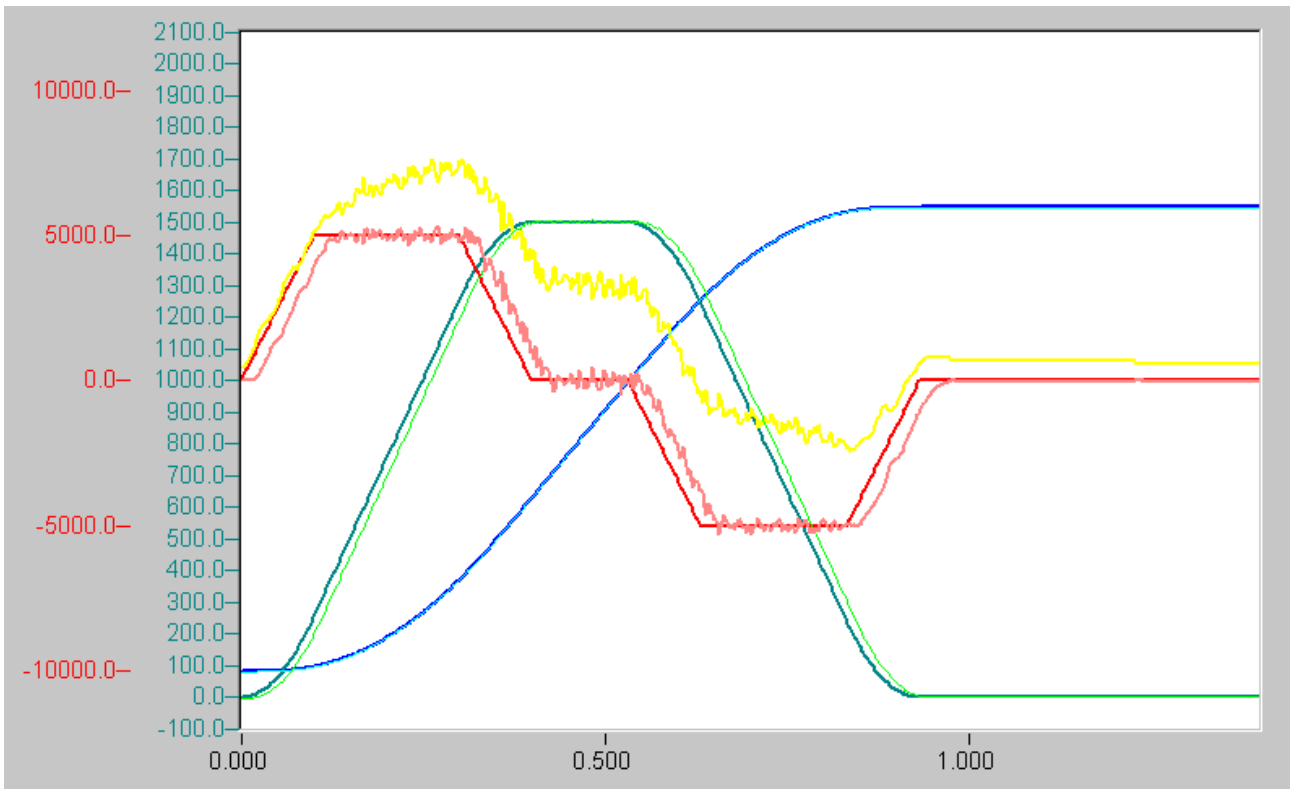


Dynamics 2.2 Dynamic profile with $A+ = A- 5000.0$ mm/s/s, $R = 50000.0$ mm/s/s, $V = 2000.0$ mm/s from 0.0 mm to 1000.0 mm

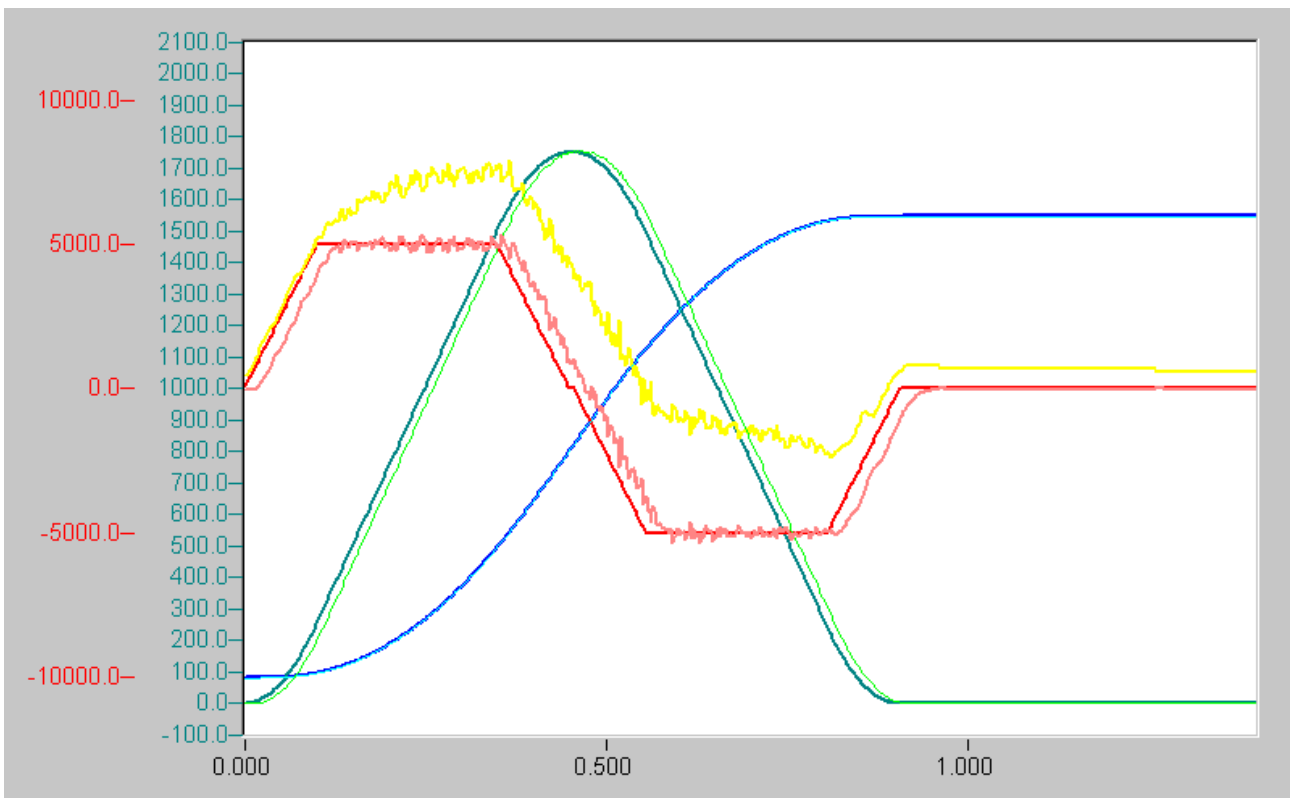


Dynamics 2.3 Dynamic profile with $A+ = A- 5000.0$ mm/s/s, $R = 100000.0$ mm/s/s, $V = 2000.0$ mm/s from 0.0 mm to 1000.0 mm

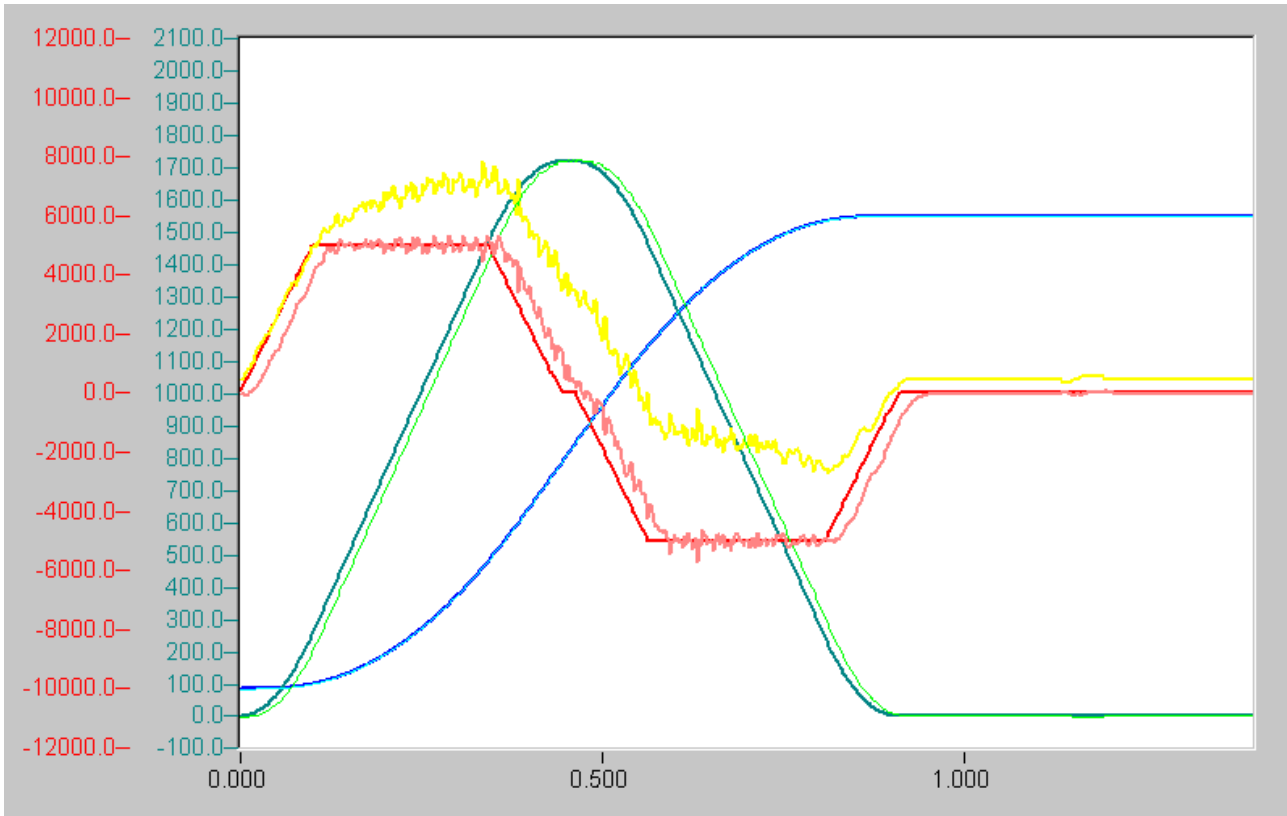
Dynamics III: velocity reduction



Dynamics 3.1 Dynamic profile with $A+ = A- 5000.0$ mm/s/s, $R = 50000.0$ mm/s/s, $V = 1500.0$ mm/s from 0.0 mm to 800.0 mm



Dynamics 3.2 Dynamic profile with $A+ = A- = 5000.0$ mm/s/s, $R = 50000.0$ mm/s/s, $V = 1750.0$ mm/s from 0.0 mm to 800.0 mm



Dynamics 3.3 Dynamic profile with $A+ = A- = 5000.0$ mm/s/s, $R = 50000.0$ mm/s/s, $V = 2000.0$ mm/s from 0.0 mm to 800.0 mm

Table 1: Global dynamic parameters

Dynamic parameters	Meaning and boundary conditions
Acceleration $A+$	Acceleration limit ($A+ > 0.0$)
Deceleration $A-$	Deceleration limit ($A- > 0.0$)
Jerk J	Amount of the acceleration change constant ($J > 0.0$)

Notes on start parameters:

1. For orders of magnitude from J to $A+$ or $A-$ in units of mm and Saf cycle time T should apply with sufficiently large distances: $J \ll \min(A+, A-)$. (An example: with Saf- cycle time $T = 1/1000$ s, acceleration as well as deceleration $A+ = A- = 10000$ mm/s/s = $1/100$ mm/T/T, jerk $J = 100000$ mm/s/s = $1/10000$ mm/T/T results in $J = 0.0001$ " $A+ = A- = 0.01$.)
2. To ensure that the set acceleration $A+$ or deceleration $A-$ is actually reached, the jerk J must at least correspond to the **critical jerk**: $J \geq \max((A+)^2, (A-)^2)/V$.

Table 2: Start parameters

Start parameters	Meaning and boundary conditions
Target velocity V	requested path set velocity (amount of the axis target velocity) ($V > 0.0, V \leq V_{max}$)
Target position P	Position to be approached P ($P_{min} \leq P \leq P_{max}$)

Notes on start parameters:

1. If the target velocity is greater than the maximum **permitted velocity** V_{max} for the axis, the start is refused and an error is generated.
2. If the target velocity is greater than the **reference velocity** V_{ref} specified for the axis, then the start is refused and an error is generated.
If the target velocity is equal or nearly equal to the reference velocity, then the axis has no or only little reserves for control.
3. For orders of magnitude from $A+$ and $A-$ to V in units mm and Saf- cycle time T should be valid with sufficiently large distances: $max(A+, A-) \gg V$. (An example: with Saf- cycle time $T = 1/1000$ s, target velocity $V = 1000$ mm/s = 1 mm/T, acceleration as well as deceleration $A+ = A- = 10000$ mm/s/s = $1/100$ mm/T/T results in $A+ = A- = 0.01 \cdot 1 = V$.)
4. If the target position P is outside of the **software limit switches** (P_{min} , P_{Max}), the start is refused and an error is generated.
5. A start from outside the software limit switches (P_{min} , P_{max}) into the range within the software limit switches is permitted.

2.2 Override and Stop

The override is used to change the target velocity of the axis online in the quickest possible way. Stop is realized via override 0.0 and has priority over all other functions.

Override

The override is an **instantaneous (not time-scaled) override**, i.e. an attempt is made to reach the new velocity level as quickly as possible with the maximum permissible changes in acceleration. The override can be changed from 0.0% to 100.0% in steps of $1/1,000,000$.

There are four override types:

1. The override refers to the target velocity level calculated and possibly reduced at axis start. If the new velocity level is not achievable, the old runtime table is maintained without notification.
2. The override refers to the target velocity specified by the user. If the new velocity level is not achievable, the old runtime table is maintained without notification.
3. The override refers to the target velocity level calculated and possibly reduced at axis start. If the new velocity level is not achievable then an attempt is made to achieve a smaller optimal velocity level.
4. The override refers to the target velocity specified by the user. If the new velocity level is not achievable then an attempt is made to achieve a smaller optimal velocity level.

The override is only activated if there is no stop and if the brake phase has not yet been reached. There is a possibility to start the axis with override 0% .

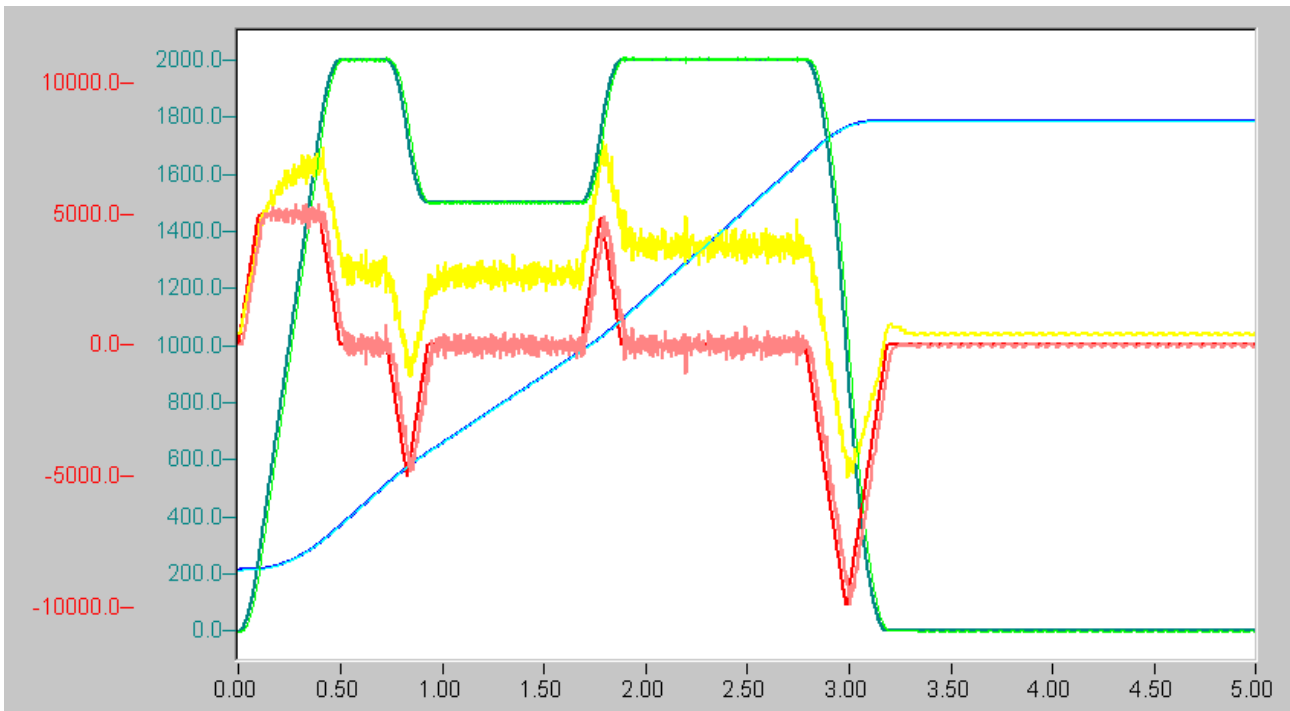
In case of excessively quick (especially periodical) changes of the overrides between two fixed velocity levels, the axis, whilst calculated deterministically, will behave indeterministically for the user.

Notes on the figures:

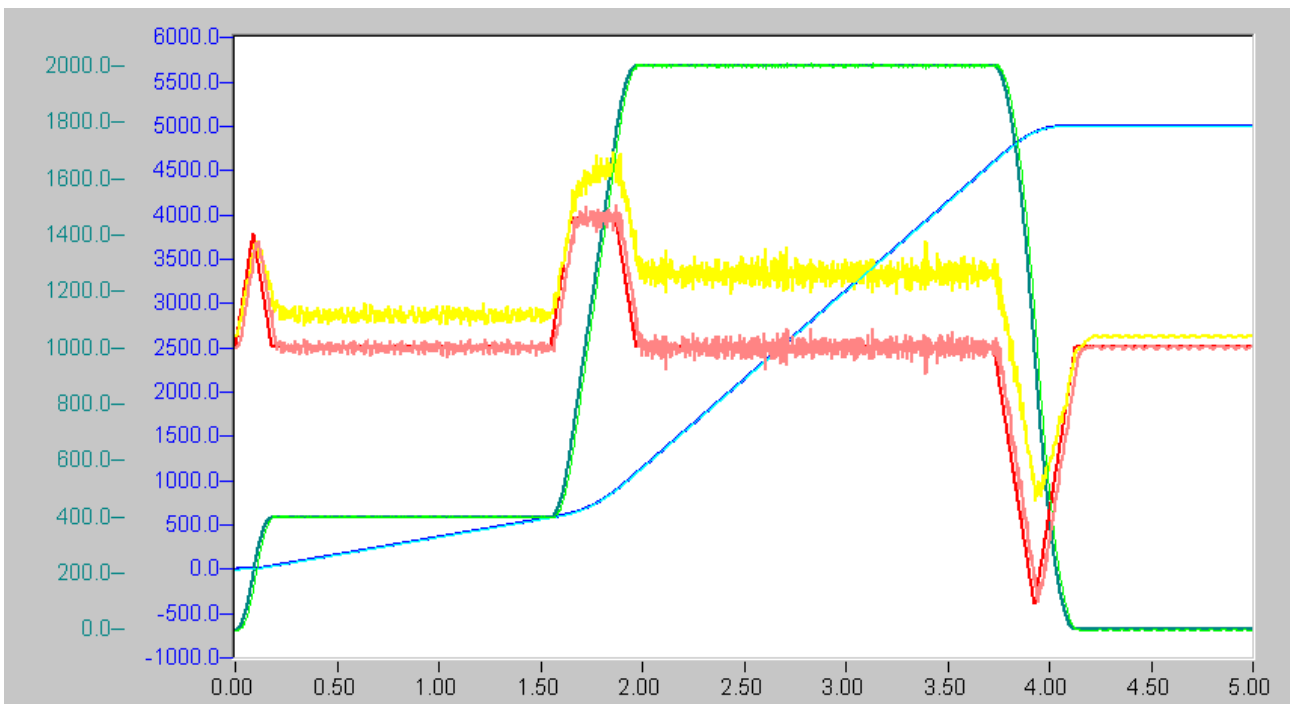
Scales: velocity [mm/s] / time [s]

Colors:

Set position: dark blue	Actual position: light blue
Target velocity: dark green	Actual velocity: light green
Set acceleration: dark red	Actual acceleration: light red
Following error: yellow	



Override changes: start override 100% -> 75% -> 100% (V = 2000.0 mm/s, from 0.0 mm to 5000.0 mm)



Override changes: start override 20% -> 100% (V = 2000.0 mm/s, from 0.0 mm to 5000.0 mm)

Stop

Stop is realized via override 0.0 and has - provided the brake phase has not yet been reached - **priority** over all other functions. If stop is requested and position compensation is active, compensation stop is activated simultaneously with the stop of the basic dynamics, in order to stop the axis as quickly as possible on the maximum permissible breaking ramp.

Emergency stop

In case of an emergency stop with parameters jerk R [mm/s/s] and deceleration $A-$ [mm/s/s], the parameters jerk and deceleration are reset, if both parameter are genuinely larger than the parameters provided at the start. The stop is then initiated with the new parameters.

N.B.: in the brake phase (natural or induced by the user) **no** emergency stop is possible.

2.3 New end position

The function 'New target position' is used to change the target position of the axis online in the quickest possible way.

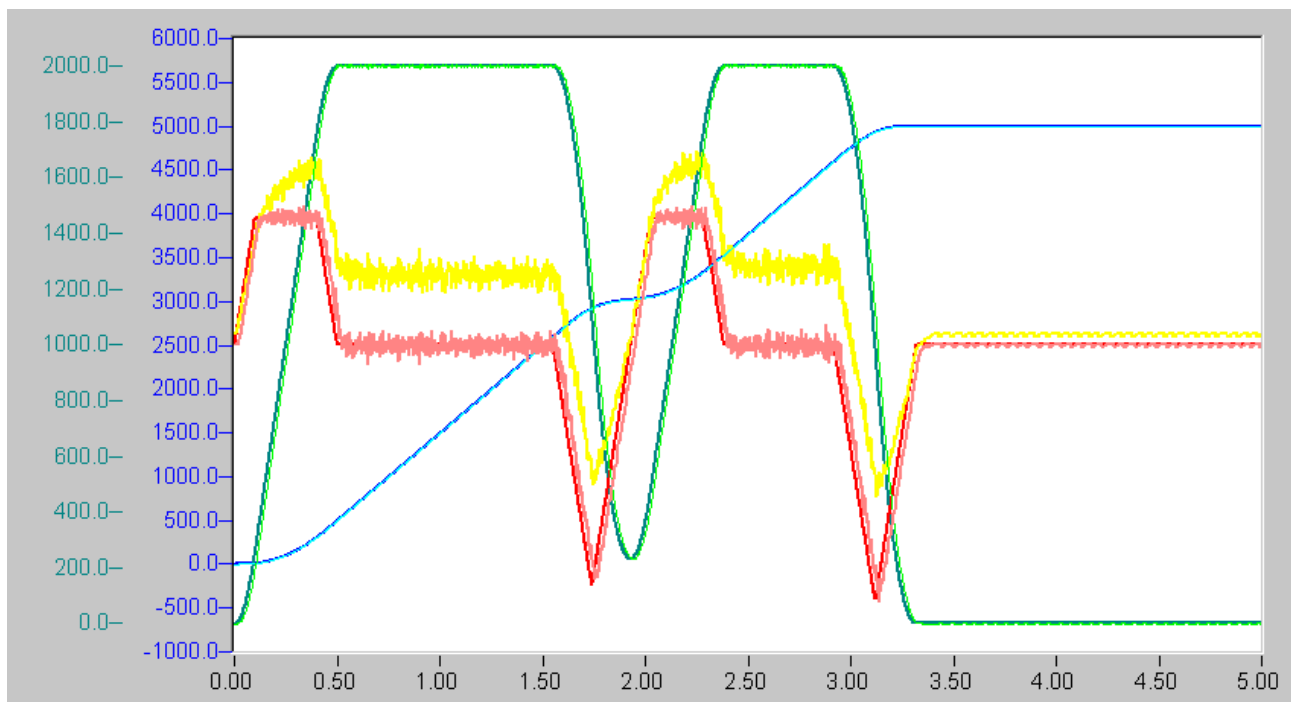
Internal sequence

If the axis was started and no stop is active, the request 'new target position' causes a query to ascertain whether

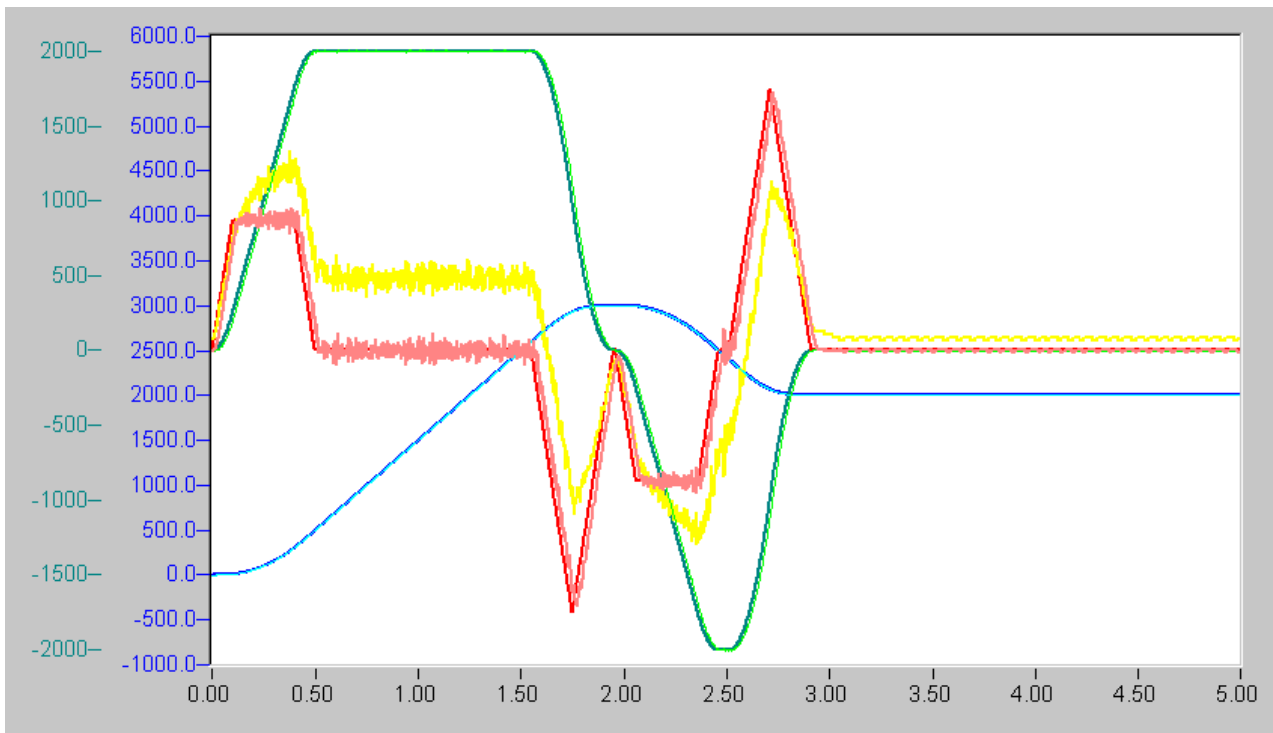
- the axis has exceeded the new target or whether
- the axis cannot be stopped before the new target.

In these cases, the axis is stopped internally via a stop and then **automatically moved back** to the new target position. Otherwise, a new runtime table is calculated and activated.

The function 'New target position' **must not be activated** as long as a position compensation is carried out. If the desired target velocity was not attainable when the axis was started, and if the old desired target velocity is attainable after a new target position is requested, then this is also attained, i.e. an axis that was previously moving slowly can suddenly become faster after a new target position is requested.



New end position during the stop phase ($V = 2000.0$ mm/s, start from 0.0 mm to 3000.0 mm, restart to 5000.0 mm)



New end position with reversal ($V = 2000.0$ mm/s, start from 0.0 mm to 3000.0 mm, restart to 2000.0 mm)

2.4 New velocity and new end position

The function 'New velocity and new end position' is used for **instantly changing the target position** and, if previously the target velocity $VOld$ required by the user could not be reached, also - provided that this is possible - for **instantly changing the current internal target velocity** towards the target velocity $VOld$ required by the user, and - after an activation position $APos$ has been crossed - **for (later) changing the target velocity** of the axis. The specified velocity $VNew$ in mm/s must be genuinely larger than 0.0 and must not exceed the maximum axis velocity. The new end position $EPos$ must be within the software limit switches. The function 'New velocity and new end position' cannot be activated during position compensation.

Internal sequence: activation position

When requesting: New velocity $VNew$ mm/s , new end position $EPos$ mm , Activation position $APos$ mm

the relationship of the positions $EPos$, $APos$ and the current position Pos to each other is first determined in relation to the logical direction of the positioning. Let the logical direction be positive:

1. if: $Pos \leq APos \leq EPos$, the new target velocity $NewV$ is activated once position $APos$ has been crossed,
2. if: $APos \leq Pos \leq EPos$, the activation position has already been crossed, and the target velocity $NewV$ is activated instantaneously,
3. if: $APos \leq EPos \leq Pos$, a stop is activated, and the system is returned automatically with the old target velocity $VOld$,
4. if: $Pos \leq EPos \leq APos$, only the new end position and the old target velocity $VOld$ is activated, but not the new target velocity,
5. if: $EPos \leq APos \leq Pos$, a stop is activated, and the system is returned automatically with the old target velocity $VOld$ to the new end position (not implemented: and the new target velocity $VNew$ is activated when the activation position is crossed).
6. if: $EPos \leq Pos \leq APos$, a stop is activated, and the system is returned automatically with the old target velocity (not implemented: new) to the new end position.

The same conditions apply analogically for negative logical direction.

If 'new velocity and new target position' is requested, a query occurs once the axis has been started, whether

- the axis has exceeded the new target or whether
- the axis cannot be stopped before the new target.

In these cases, the axis is stopped **internally** via a stop and then **automatically moved back** to the desired new target position with the appropriate target velocity V_{Old} (if this is achievable) or V_{New} (if this is achievable).

Internal sequence: New velocity

If no return is required and either the new set velocity is requested instantaneously or the activation position is crossed, a new run time table is calculated with the new set velocity. If, due to the ratio of remaining distance and new set velocity, the new set velocity is not achievable, the set velocity is reduced **iteratively** 7-8 times. If no solution can be found (adverse scenarios are: small remaining distance and/or large new set velocity - relative to the old set velocity), the old run time table is maintained.

Special features

In contrast to the function 'New end position', a stop does not have priority with the function 'New velocity and new target position'. This means that incoherent short successive instructions do not have to lead to the axis being stopped. The function 'New velocity and new target position' is compatible with the override function. If the request occurs during the brake phase, the instantaneous instruction, provided it cannot be carried out instantaneously, is suspended for later action, the instantaneous action is carried out with optimum delay, and the suspension is subsequently lifted. If the function with instantaneous (Event 1) and later activation (Event 2) is interrupted between Event 1 and Event 2 by a function that can only act instantaneously (Event 3), the activation of Event 3 is deleted. If the function is interrupted between Event 1 and Event 2, by a function acting again instantaneously (Event 3) and later (Event 4), Event 3 is replaced with Event 4.

2.5 Position compensation

Sometimes it is desirable for an axis to carry out **additional positioning** by a preset distance (compensation difference D - may also be negative) within a certain distance (compensation length L) compared with the positioning resulting from the runtime table. This process is mostly used for spatial synchronization with other axes and can be switched to each axis (master, slave).

If the position compensation cannot be carried out with the specified compensation parameters, then the maximum possible compensation is automatically carried out.

The position compensation can be aborted by compensation stop (continuous).

⚠ WARNING

If the compensation difference D is negative (deceleration), (double) motion reversal is possible.

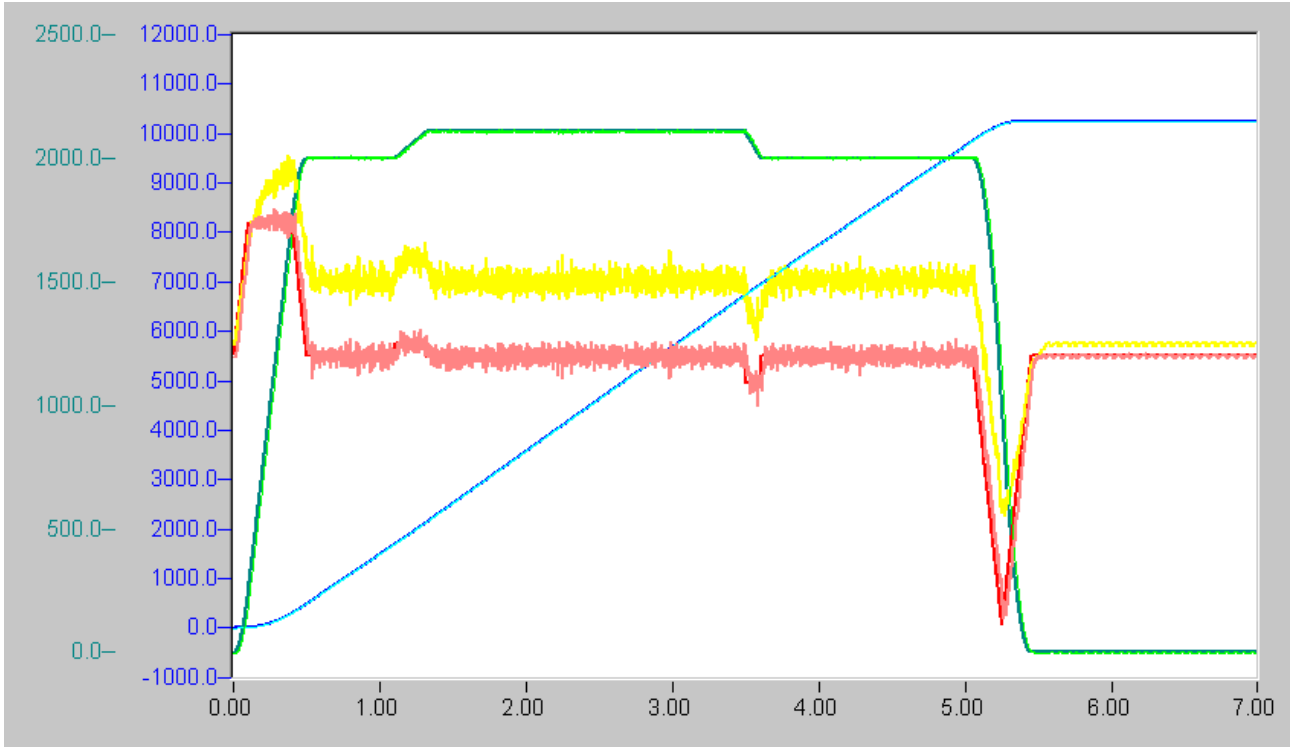
General notes:

In the following sections, the dynamics and positioning specified by the start parameters will be referred to as basic dynamics and **basic positioning**.

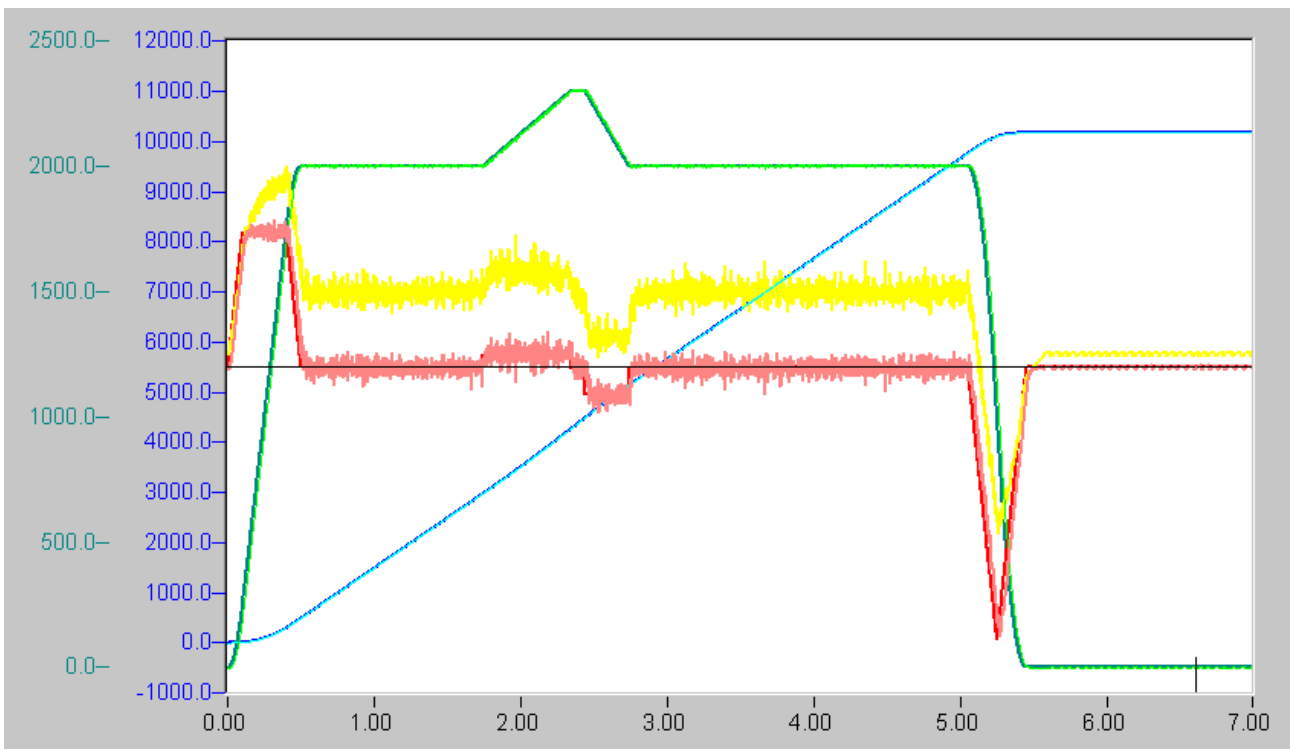
1. D may be negative. In this case, the overall positioning lags behind the basic positioning.
2. Compensation is complete once the distance $L \pm D$ has been travelled.
3. Compensation is a **linear path control**, i.e. during compensation the axis will follow all the changes in basic dynamics (e.g. override 0.0).
4. During compensation, axis compensation start is locked, i.e. there is no compensation iteration.
5. During the stop phase, no compensation will be started.
6. End of path during axis compensation stops the compensation.
7. Stop during axis compensation stops the compensation at the end of path caused by the stop.
8. A virtual axis (encoder axis) cannot carry out a compensation.
9. The function 'New target position' **cannot be activated** during position compensation.

Algorithm

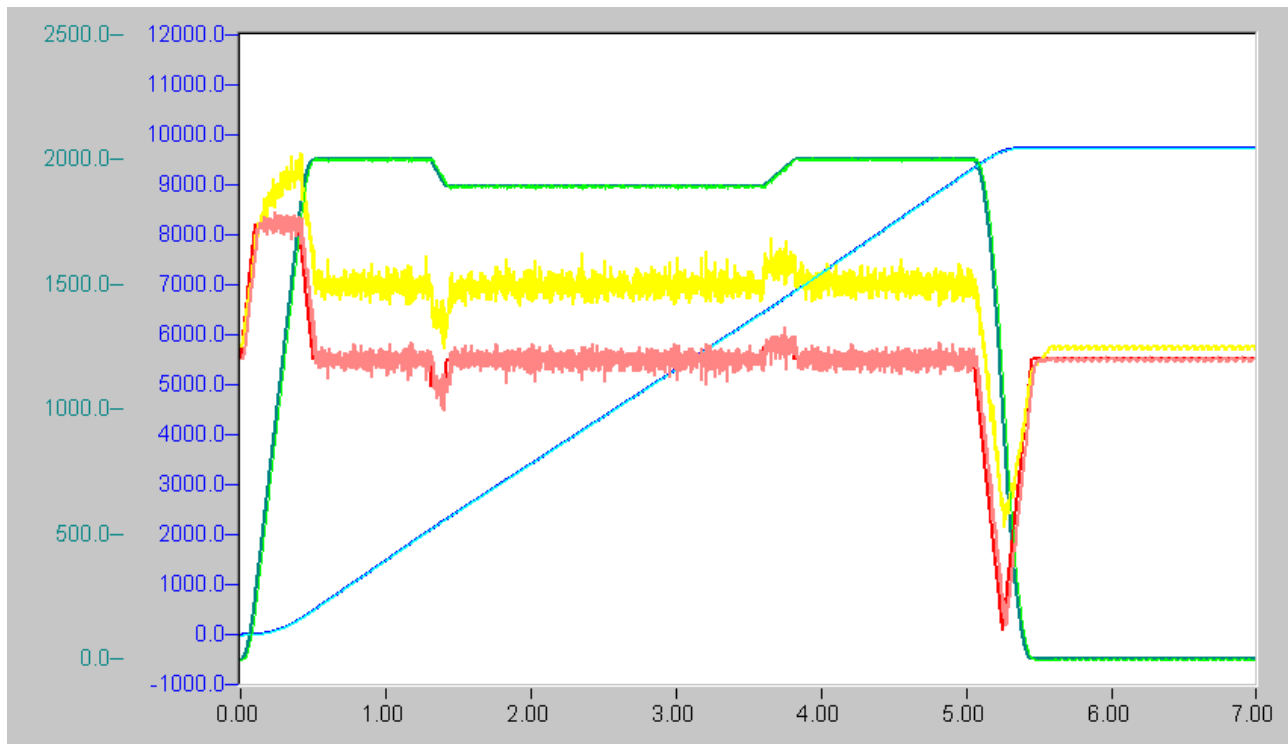
After checking the compensation start parameters, a calculation will be carried out to check whether compensation of the difference D over distance L is feasible with the compensation start parameters. If this is not the case, the maximum possible compensation difference is calculated and activated. In any case, the parameters are chosen in such a way that compensation occurs **over the whole compensation length**. The setpoint generation of the compensation is a non-jerk-limited 3-phase-algorithm (**trapezoidal velocity profile**), whereby different ramp slopes are possible.



Full compensation: start from 0.0 mm to 10,000.0 mm with $V = 2000.0$ mm/s
 Compensation $A+ = 500.0$ mm/s/s $A- = 1000.0$ mm/s/s, $V = 300.0$ mm/s, $VP = 2000.0$ mm/s,
 $L = 5000.0$ mm, $D = 250.0$ mm
 End position 10250.0 mm



Partial compensation: start from 0.0 mm to 10,000.0 mm with $V = 2000.0$ mm/s
 Compensation $A+ = 500.0$ mm/s/s $A- = 1000.0$ mm/s/s, $V = 500.0$ mm/s, $VP = 2000.0$ mm/s,
 $L = 2000.0$ mm, $D = 500.0$ mm
 End position 10165.0 mm



Full negative compensation: start from 0.0 mm to 10,000.0 mm with $V = 2000.0$ mm/s
 Compensation $A+ = 500.0$ mm/s/s $A- = 1000.0$ mm/s/s, $V = 300.0$ mm/s, $VP = 2000.0$ mm/s,
 $L = 5000.0$ mm, $D = -250.0$ mm
 End position 9750.0 mm

Start parameters

Position compensation: start parameters

Start parameters	Meaning and boundary conditions
Compensation length L	Distance, over which additional positioning occurs ($L > 0.0$)
Compensation difference D	Additional travel distance ($D \neq 0.0$)
Process velocity VP	Velocity, to which the velocity increase refers (generally axis target velocity) ($VP > 0.0$)
Velocity increase VK	Permissible velocity increase/reduction relative to the process velocity ($VK > 0.0$)
Acceleration $AK+$	Required acceleration increase relative to the global path acceleration ($AK+ > 0.0$)
Deceleration $AK-$	Required deceleration increase relative to the global path deceleration ($AK- > 0.0$)

Notes on start parameters:

1. The compensation difference D may be positive (making up) or negative (deceleration). If $D < 0.0$, (double) motion reversal is possible.
2. **Process velocity**. If compensation is activated while the axis is traveling at its (global) target velocity, then VP should be set to equal this velocity. If the axis is not yet (no longer, not - Override -) at its global target velocity, VP should be set to the amount of the current local target velocity of the axis.

3. If the preset process velocity VP does not coincide with the amount of the real current local axis velocity v , the following applies. If $VP > v$, the real compensated distance (absolute) will be less than the required one. If $VP < v$, the real compensated distance will generally be equal the required one. This means that for stop or override 0.0 , the actual compensated distance (absolute) will be less than the required one.
4. The values $AK+$ and $AK-$ are increases in path acceleration/path deceleration values, i.e. if $D > 0.0$, compensation starts with the acceleration increase $AK+$, whilst, if $D < 0.0$, compensation starts with the deceleration increase $AK-$.
5. If the compensation length L is greater than the remaining axis distance to be traveled, the remaining distance will be used instead of L .

Compensation stop

The function 'compensation stop' terminates axis compensation, if an axis is in compensation mode but not in the braking phase of compensation, by replacing the distance to be compensated D with the actually compensated distance at the end the compensation.

If the axis is stopped by the user and compensation is active, compensation stop is automatically requested together with the stop of the basic dynamics.

2.6 External set value generation

External setpoint generation (external setpoint specification) enables the implementation of individual setpoint generators within the PLC, or appropriate connection of existing internal and external generators, and their simple and transparent integration into the TwinCAT system. Axes can be positioned either with an external setpoint generator alone, or superposition (mixing) of existing internal generators and external setpoint sources is possible.

These new characteristics and the fact that an axis can change its operating modes at runtime (e.g. interpolating path axis (NCI), master PTP axis, slave axis, external setpoint generation) further enhance the flexibility of a TwinCAT axis.

In general, the superimposition of internal and external setpoint sources enables a multitude of new applications and solutions.

Examples of use:

- External setpoint specification can be used for implementing an *individual new setpoint profile*, which is not available in TwinCAT NC. This profile can be mathematically very simple or as complex as necessary. Since the externally generated setpoint profile can also be superimposed on the standard profile, *corrective functions* can thus also be supplemented. In this way, a position-based setpoint profile can be implemented for systems that are difficult to control or non-linear, for example a position-based deceleration ramp for hydraulic axes.
- One application for the *superposition (mixing) of two internal setpoint sources* is the combination of interpolating path processing (NCI) and the 'flying saw' function (PTP slave). Multi-dimensional *machining*, e.g. a circle, *on a moving workpiece* is thus possible, which previously was only possible with stationary workpieces. This measure increases the efficiency of a machine.
- External setpoint generation can also be used in order to enable a certain *lead time* (in the multi-digit second range) between the unit generating the setpoints (e.g. the execution of a DIN-NC program feeding virtual axes) and the real axes. *Collision monitoring and strategies for collision avoidance* can then be realized.
- Another important use case is *coordinate transformations* (restricted to regular and unique transformations without singularities). E.g. geometries like a 'two-bar linkage' or conversion between rotatory and translatory coordinates.

This new function is available with the PLC function blocks *MC_ExtSetPointGenEnable* and *MC_ExtSetPointGenDisable* in the *TcMc.lib* library. Furthermore, there is an example for an external setpoint generator (3-phase profile with acceleration rectangle), see section [TwinCAT NC-PTP Examples \[► 40\]](#).

Activation of the external setpoint generation:

The adoption of the external setpoint specifications is initiated by an ADS axis function from the PLC with the function block *MC_ExtSetPointGenEnable*. The axis start types *absolute* (1) or *relative* (2) and, if sensible and required, the target position of the axis, are specified as parameters. This target position has either an absolute or relative effect, depending on the starting mode, and enables position and target position window monitoring (PEH). The activation of the external setpoint specification is thus similar to a conventional axis start.

A status bit from the status double word of the cyclic axis interface from the NC to the PLC provides information about the state, i.e. whether the external setpoint generation is active or inactive (see function *AxisHasExtSetPointGen* in *TcNc.lib*).

In order for the activation to be accepted without error, no axis error must be present, the controller and feed releases must be granted, the drive hardware must be ready to run and the axis must not be coupled (no slave).



To be observed:

- In case of an axis error (runtime error of the axis) or an axis reset (including edge change of the software controller enable) the external setpoint specification is automatically disabled (similar to an abortion of PTP axis positioning in case of a runtime error or axis reset).
 - If only external setpoint generation is carried out, both *absolute* and *relative* start types are permitted.
If external setpoint generation is started in parallel or in addition to internal setpoint generation (PTP), only the *relative* starting mode is possible, since the internal positioning is ultimately always converted to an absolute start, and it is not possible to have two absolute positions simultaneously.
 - Whilst external positioning can be activated during an active internal PTP positioning via the *relative* start type, the opposite case is not permitted.
 - If the external setpoint generation is already active and this is activated for the repeated time, then the start types must match, otherwise this is answered with an error.
The repeated activation of the external setpoint generation can be useful insofar as here the target position is taken over each time and thus the target position window monitoring (PEH) is updated.
-

Disabling the external setpoint generation:

External setpoint specification is switched off from the PLC via the function block *MC_ExtSetPointGenDisable*. The above-mentioned status bit from the status double word of the cyclic axis interface from the NC to the PLC provides feedback (see function *AxisHasExtSetPointGen* in *TcNc.lib*).



After the deactivation of the external setpoint generation, the external set position is taken over for exactly one more cycle.

Specification of setpoints via the cyclic axis interface:

After faultless activation of the external setpoint generation, in each PLC cycle a setpoint packet can now be created at the axis interface. A setpoint package consists of the external set position, target velocity, set acceleration and set direction (-1, 0, +1).

The set direction is of particular importance here, because setpoints are only accepted and applied internally if this is not equal to zero (0: no movement). For the complete range of external setpoint generation, the direction flag therefore has to be set quasi-overlapping at the start and at the end.



To be observed:

- After resetting of the set direction to zero, the external set position is taken over for exactly one further SAF cycle.
 - The set direction should already be set to a value not equal zero (i.e. -1 or +1) one cycle before and for one more cycle after the motion. This provides a safe time frame that is guaranteed to prevent a value being ignored, either at the start or at the end.
-

Settings of the PLC task:

Ideally, the PLC task that generates and specifies the setpoints should run synchronously, i.e. with the same cycle time as the NC-SAF task (port 501). This PLC task also has to have the same high priority, perhaps an even higher one, than the NC-SVB task (port 511).

Optimization options:

The dead time generated due to the transport via the cyclic axis interface (also, the input information leading to the calculation of the external setpoints may be one cycle old) can be compensated for an axis that is operated in the velocity interface. This dead time compensation can be set in the TwinCAT System Manager via the global axis parameter *delay time between velocity and position* (default value: 0.0 s).

3 TwinCAT Slave Axes

Definition

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 set position values - 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 synchronisation 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.

Slave topologies

The permitted slave topologies (link options) are star and cascade.

Star, i.e. all slaves are linked directly to a master axis (virtual master axis).

Cascade, i.e. all slaves are linked consecutively through a chain, and the first slave axis is linked to a master axis (virtual master axis).

The cascade may be set up in any order.

Uncoupling within a cascade may occur in any order.

Slave types

Depending on the type of **set value generation**, a distinction is made between the following types:

- linear slave axis,
- flying saw,
- camshaft axis,
- multi-table axis,

These will now be explained briefly.

Linear slave axis

A slave axis whose position setpoints are determined from the position setpoints of another axis by an affine transformation is called Linear slave axis [► 28].

The following applies: $\text{slave set position} = \text{coupling factor} \times \text{master set position} + \text{offset}$, whereby the offset results from the coupling position and the coupling factor is determined during coupling. The coupling factor can be changed online.

While the coupling of the slave to the master is done offline (master and slave are stationary), the decoupling can be done online: the slave is thereby converted into a master that continues to operate as if it were still a slave, but that can also be stopped independently of the former master.

Cyclic slave axis

A slave axis, whose position set values are determined from

1. the position set value of another axis (master axis) through an affine transformation and
2. a coupling factor set through the cyclic PLC-NC interface within the PLC cycle,

is called cyclic slave axis [► 38].

3.1 NC Error Handling

1. Advanced NC error handling for coupled axes

Sample of the NC axis parameters from the TwinCAT System Manager:

Error Propagation Mode	'INSTANTANEOUS'	E	
Error Propagation Delay	0.0	F	s

ErrorPropagationMode

The advanced NC error handling for coupled axes can be configured by means of the parameter *Error Propagation Mode*.

INSTANTANEOUS setting:

Selecting *Instantaneous* here corresponds to the reactions to errors of TwinCAT up to 2.10 (default). In the event of a error in one axis, the NC immediately initiates an error reaction, and halts all the axes involved in the master/slave group.

DELAYED setting:

The basic idea of this setting is to not initiate an automatic, immediate NC error reaction whenever an error occurs. The user has the option of initiating an error reaction that has been individually adapted to the particular application.

If a runtime error now occurs in a coupled axis, then only this particular axis is switched without delay into an error state by means of an error reaction, whereas the other axes in the master/slave group are not affected immediately (*local error reaction*). Propagation of the error through to the other axes involved is suppressed for the delay time parameterized in *Fehlerreaktionsverzögerung* (Error reaction delay). During this period, the user is responsible for initiating the error reaction that he requires (e.g. stopping the master axis, or decoupling from parts of the master/slave structure).

Once the parameterized delay time has elapsed, the familiar NC error reaction comes into play (*global error reaction*). This error reaction, however, acts only on the group of axes present at the time to which the axis with the error belongs. It is possible, for instance, for the original master/slave group to first be decoupled to create one group with no error and another group with an error.

Axis interface

If a runtime error occurs in a master/slave axis configured as DELAYED, it is indicated at the same time by the two NC status bits Error (bit 31) and ErrorPropagationDelayed (bit 6). Once the error reaction delay time has elapsed, the ErrorPropagationDelayed bit is reset to FALSE.

Requirements:

The extended NC error handling (*error reaction mode*) and the changeover to actual value coupling (*use actual value in deactivated state*) only applies to master/slave axes whose axis types are *Continuous Axis* (PTP axis) and *Encoder Axis*.

2. Changeover to actual value coupling

Sample of the NC axis parameters from the TwinCAT System Manager:

Couple slave to actual values if not enabled	FALSE	B	
Velocity Window	1.0	F	mm/s
Filter Time for Velocity Window	0.01	F	s

CoupleSlaveToActValues

Basically the setpoint generation between coupled axes is always based on setpoints instead of actual values.

The property *Changeover to actual value coupling* is parameterized by the parameter *Couple slave to actual values if not enabled* and is always to be understood in connection with a master/slave coupling. There are two different cases to distinguish here.

1st case: Switch to actual values when an axis has an error

If an error occurs in a coupled axis when error reaction is delayed (*error reaction mode: DELAYED*), then this axis can be switched over by the NC from set values to actual values. Although this changeover does not have any effect on the faulty axis itself (which has already been halted in an error state), it remains possible to drive the subsidiary slave axes by means of the actual values for this axis. This property can result, in the case of an error, in better synchronicity between the axis with the fault and the subsidiary slave axes if the axis that has the error moves further due to its inertia.

2nd case: Change to actual values when the main master axis has been deactivated

The changeover to actual values even when no error is present is only available to the main master, i.e., the head of the coupling in a master/slave structure. A master/slave group without errors, in which the NC controller enable has set all the axes ready for operation (position control) is a precondition for this changeover.

If the NC controller enable is now removed from the main master, the entire group is controlled according to the actual values of the main master. In this way, for instance, the main master can be pushed manually, and all the other axes that are coupled to this master follow in accordance with their mathematical coupling specification (e.g. cam plates).

Requirements:

The extended NC error handling (*error reaction mode*) and the changeover to actual value coupling (*use actual value in deactivated state*) only applies to master/slave axes whose axis types are *Continuous Axis* (PTP axis) and *Encoder Axis*.

3.2 Linear Slave Axes

A slave axis whose position setpoints are determined from the position setpoints of another axis by an affine transformation is called **Linear slave axis**.

The following applies: $\text{slave set position} = \text{coupling factor} * \text{master set position} + \text{offset}$, whereby the offset results from the coupling position and the coupling factor is determined during coupling. The coupling factor can be changed online.

While the coupling of the slave to the master takes place offline (master and slave are stationary), the decoupling can take place online: the slave is thereby converted into a master, which continues to run as if it were still a slave, but which can also be stopped independently of the former master.

Glossary

Decoupling: offline: logical decoupling of the slave axis from the master axis while both axes are stationary.

Decoupling: online: conversion of a slave axis in motion into a pseudo master axis.

Control axis: axis, from which a slave axis receives its control values (it can be a master axis, a slave axis, a pseudo master axis or a virtual axis).

Basic dynamics: slave-dynamics without consideration of compensations.

Compensation: additional positioning of an axis which is given additively (may be negative) to the positioning preset by the start.

Master axis: axis with full functionality (start/ stop/ override/ new target position/ compensation).

Masterslave: slave axis whose control axis is a master axis, virtual axis or pseudo master axis.

Pseudo master axis: axis with limited functionality (stop, [override],[compensation]), generated through the conversion of a slave axis into a master axis.

Interfaces

The [interfaces](#) [► 29] consist of the System Manager, PLC blocks and the cyclic NC-PLC axis interface.

Coupling and coupling factor

During [couplin \[▶ 29\]](#)g, a master axis (which may itself be a slave axis) will be assigned to the slave axis, and the coupling factor is set.

Decoupling and new end position

Decoupling is possible in every dynamic state of the master axis - the slave axis is converted into a master axis and continues to travel as if it was still a slave axis, although it can be stopped individually. The coupling flag is reset.

During offline decoupling (decoupling at standstill), the slave axis is transformed into a PTP master axis.

During online decoupling (decoupling in motion), the slave axis is converted back into a PTP master axis. A new end position in the direction of movement may be aimed for. See: [Decoupling and new end position \[▶ 32\]](#)

Position compensation

Online position correction of the slave axis, during which the slave axis is moved along a specified path by a certain additional or reduced distance compared with the coupling to the master axis. The compensation dynamics must be parameterized. If the desired compensation cannot be carried out with the specified compensation start data, the maximum possible compensation will automatically be carried out.

Compensation may be terminated prematurely via a compensation stop. See: [Position compensation \[▶ 35\]](#)

3.2.1 Interfaces

The user can operate the linear slave axis via the System Manager or PLC blocks. Furthermore, a cyclic NC-PLC interface is available (in the System Manager or via the PLC) that contains the **axis process image** and enables access to various variables.

System Manager

The System Manager offers access to the complete slave axis functionality

- Coupling with declaration of the master axis and setting of the coupling factor
- Changing the coupling factor
- Uncoupling

PLC Blocks

The SPS Library offers access to the complete slave axis functionality

- Coupling with declaration of the master axis and setting of the coupling factor
- Changing the coupling factor
- Uncoupling

NC-PLC-Interface

The cyclic NC-PLC axis interface contains the axis process image and particularly access to the coupling information (coupling flag).

3.2.2 Coupling and coupling factor

During coupling, a master axis (which may itself be a slave axis) will be assigned to the slave axis, and the coupling factor is set.

Coupling

Coupling can only occur when both axes are stationary. Coupling is therefore only a logical allocation and has (as long as the master is not moved) no dynamic consequences, but locks certain functionalities for the slave axis (e.g. start and stop). The coupling flag is set.

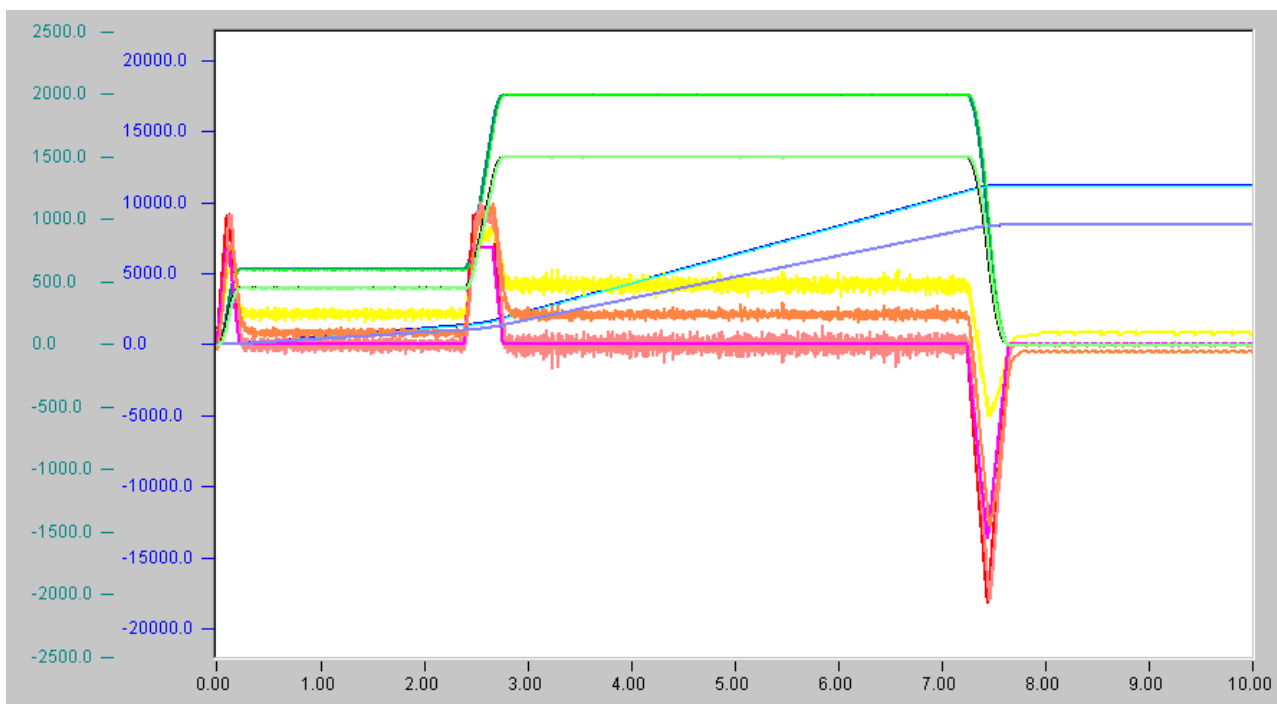
Coupling interfaces

System Manager: Coupling

PLC Blocks: Coupling

Linear path control

The slave follows the movement of the master according to the master positions, the coupling offset and the coupling factor.

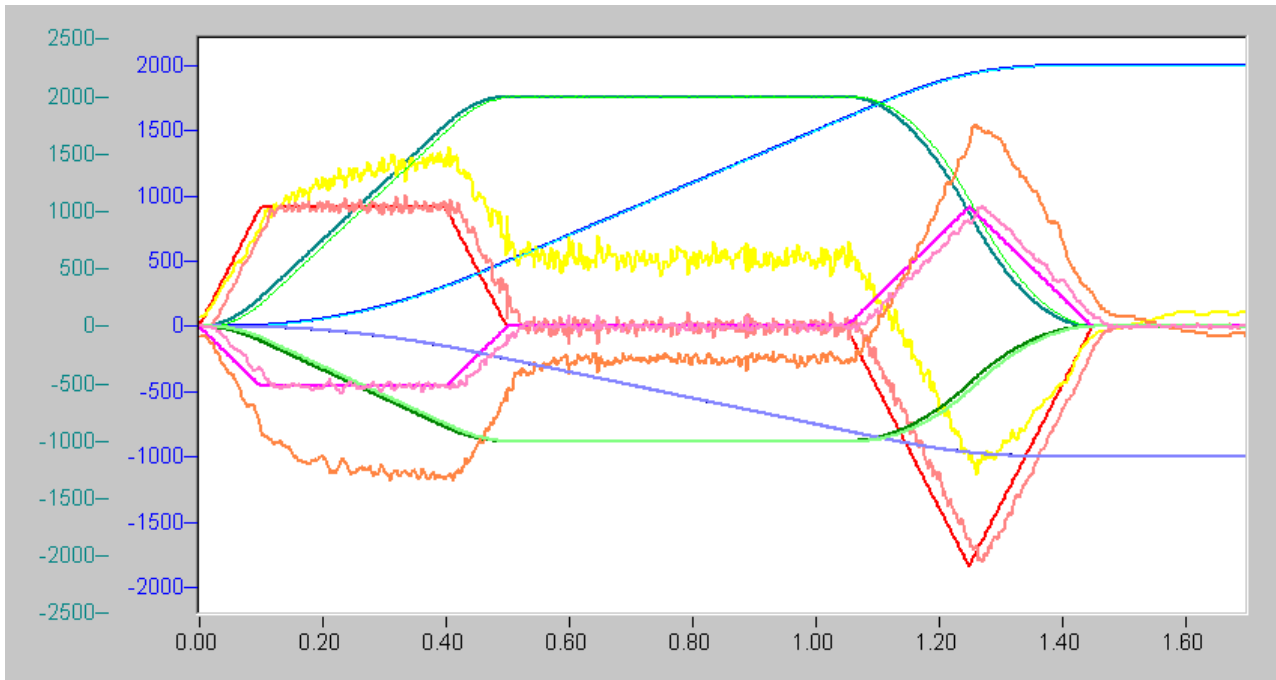


Master start from 0.0 mm to 20,000.0 mm with $V = 2000.0$ mm/s at start override 30% and coupling factor 0.75. Then (master) override to 100%, then (master) stop.

Coupling factor

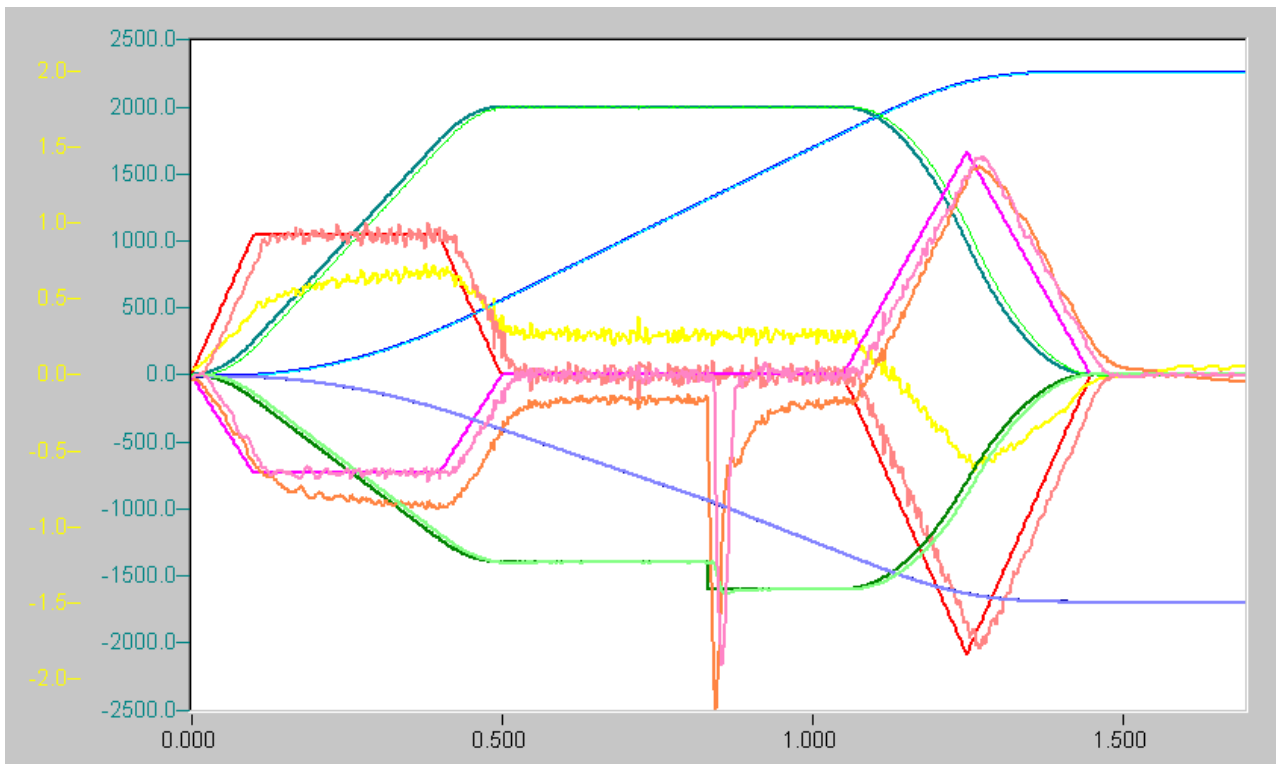
The **coupling factor** (gearing factor) must not equal zero, but may be negative. The following applies:

- Slave set position = coupling factor * master set position + offset, whereby the offset is determined from the coupling positions,
- Slave target velocity = coupling factor * master target velocity.



Coupling with fixed negative coupling factor:

Master start from 0.0 mm to 2000.0 mm with $V = 2000.0$ mm/s, coupling factor -0.5.



Coupling with variable coupling factor:

Master start from 0.0 mm to 2000.0 mm with $V = 2000.0$ mm/s, coupling factor -0.7, new coupling factor -0.8.

Changing the coupling factor

The coupling factor (not equal zero) may be changed online, but the effects (following error) are in the hands of the user. During travel, the coupling factor should only be changed moderately, since a following error is generated. Changing the coupling factor from plus to minus (of vice versa) causes a motion reversal of the slave axis.

Coupling factor interfaces

System Manager: coupling factor

PLC function blocks: coupling factor

3.2.3 Uncoupling and new end position

Decoupling is possible in every dynamic state of the master axis - the slave axis is converted into a master axis and continues to travel as if it was still a slave axis, although it can be stopped individually. The coupling flag is reset.

During offline decoupling (decoupling at standstill), the slave axis is converted into a PTP master axis.

When decoupling online (decoupling in motion), the slave axis is converted back into a PTP master axis. It is possible to move to a new end position or a new velocity in the direction of movement.

Offline decoupling

Linear slave axes may be decoupled off line (while stationary). They are thus converted back into a master axis.

Online decoupling

During online decoupling (decoupling in motion), the slave axis is converted into a master axis (**Online conversion from slave to master**).

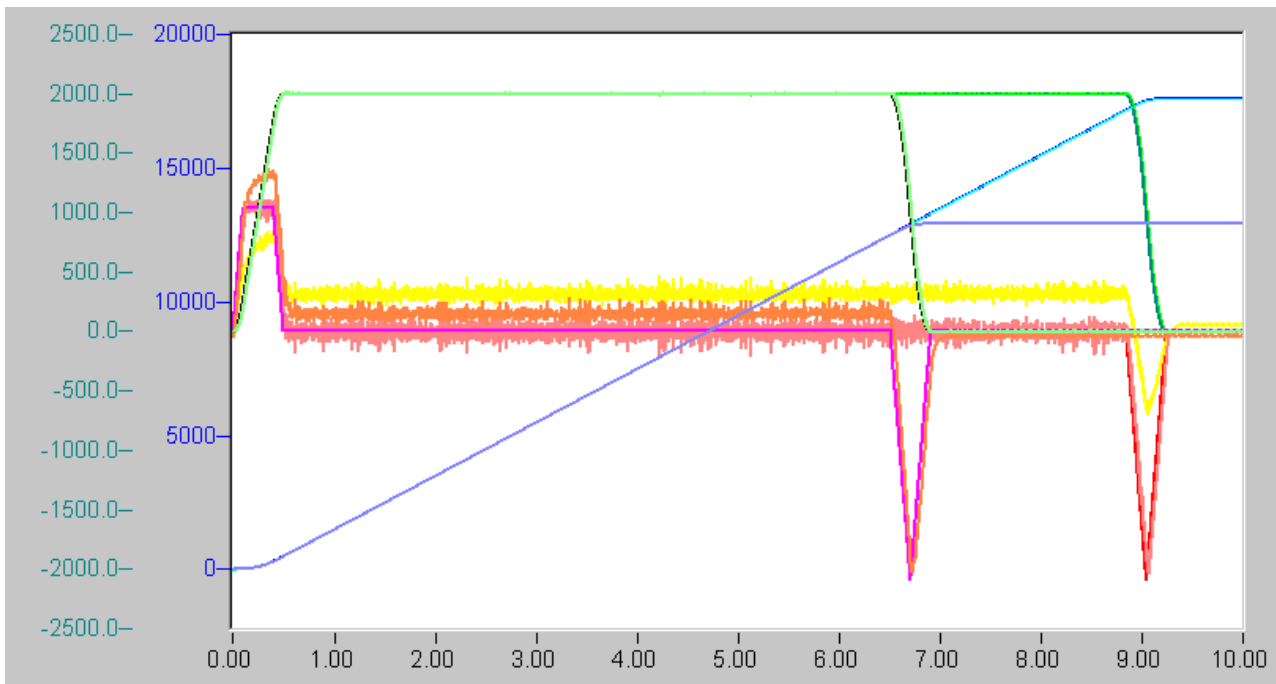
After decoupling, the converted master axis continues to move as if it were still a slave.

Override: if an override is requested after decoupling, it will relate to the absolute value of the last set coupling factor multiplied by the global internal target velocity of the former master.

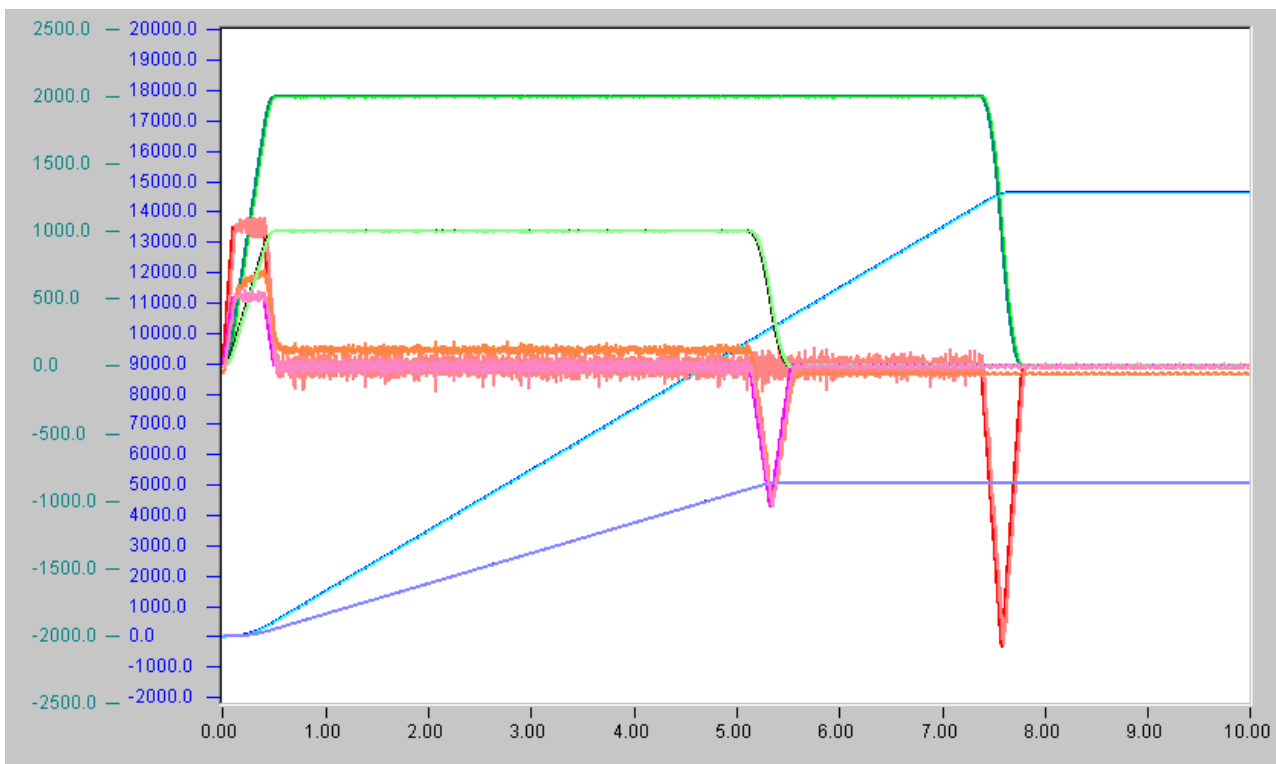
The online conversion of a slave axis into a master axis is only possible, if the velocity of the master axis (and therefore the velocity of the slave axis) is not equal zero (override > 0.0), otherwise the conversion occurs offline. The main purpose of online conversion is the separation of master and slave axis (or splitting of a cascade into partial cascades), in order to be able to stop both components independent from each other.

General notes for **decoupling during compensation**:

- Decoupling when slave to be decoupled or a slave in the chain between master (exclusively) and slave to be decoupled (inclusively) undertakes a compensation run: the current slave basic dynamics is maintained and the current compensation velocity is maintained; compensation is never terminated again.
- Decoupling during compensation travel of the master: the current basic slave dynamics and the current compensation velocity are maintained; the compensation will never be completed.

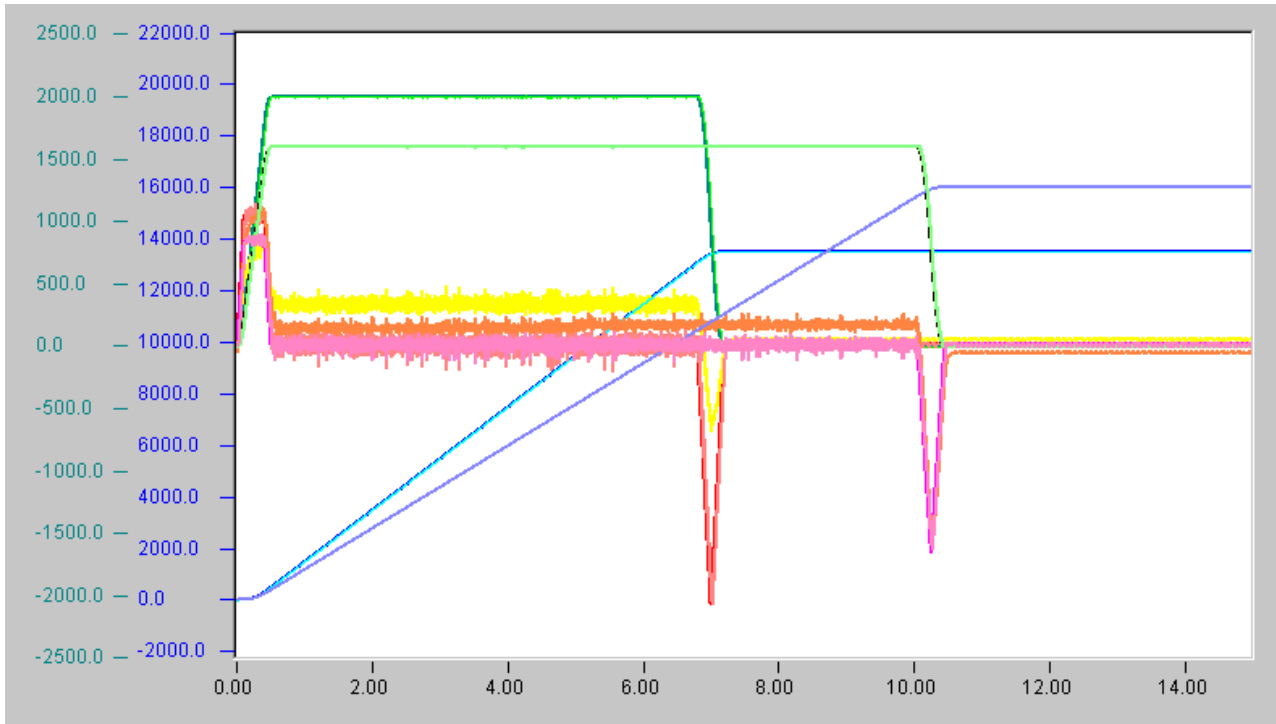


Master axis: start from 0.0 mm to 20,000.0 mm with $V = 2000.0$ mm/s.
 Slave axis: coupling and start position 0.0 mm, coupling factor 1.0.
 Online decoupling, slave stop, then master stop.

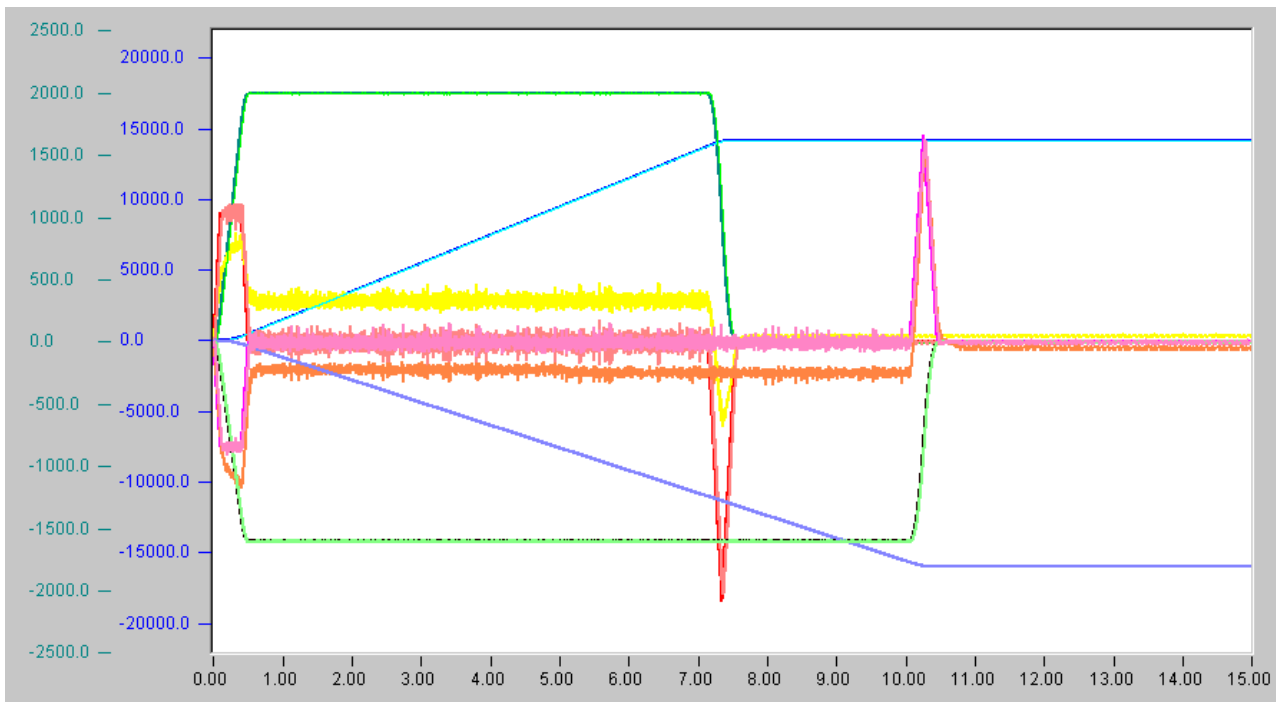


Master axis: start from 0.0 mm to 20,000.0 mm with $V = 2000.0$ mm/s.
 Slave axis: coupling and start position 0.0 mm, coupling factor 0.5.
 Online decoupling, slave stop, then master stop.

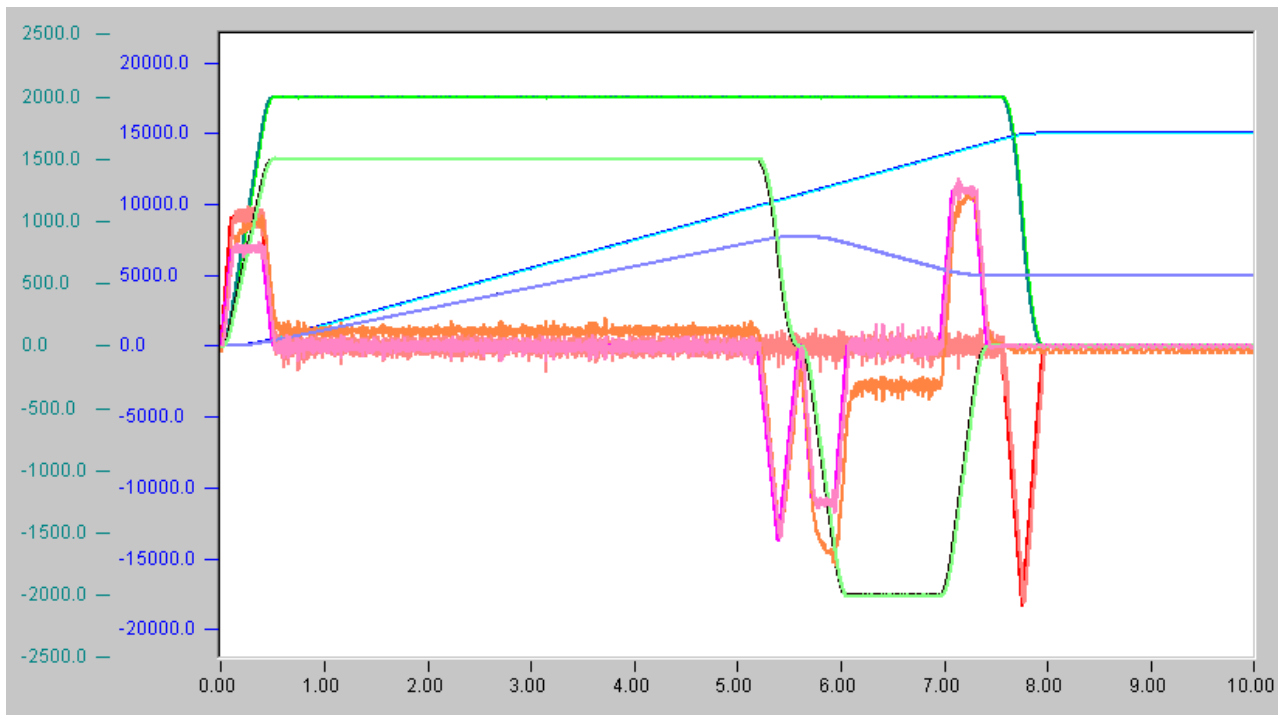
Online decoupling and new end position



Coupling factor 0.8, decoupling and aim for new end position in +ve direction of travel.



Coupling factor -0.8 decoupling and aim for new end position in -ve direction of travel.



Coupling factor 0.75, decoupling and aim for new end position against direction of travel.

Interfaces

System Manager

PLC function blocks

3.2.4 Position compensation

In the following sections, the dynamics and positioning specified by the coupling to the respective master will be referred to as basic dynamics and **basic positioning**. Position compensation is an additional positioning that is additively superimposed on the basic positioning. Over a certain distance (compensation length L), additional positioning is carried out by a desired distance (compensation difference D). The aim of slave axis position compensation is generally spatial synchronization with other axes.

Position compensation may be aborted via a compensation stop (continuous).

Position compensation of slave axes is a variant of [master axis position compensation](#) [► 20].

⚠ WARNING

If the compensation difference D is negative (deceleration), (double) motion reversal is possible.

General notes:

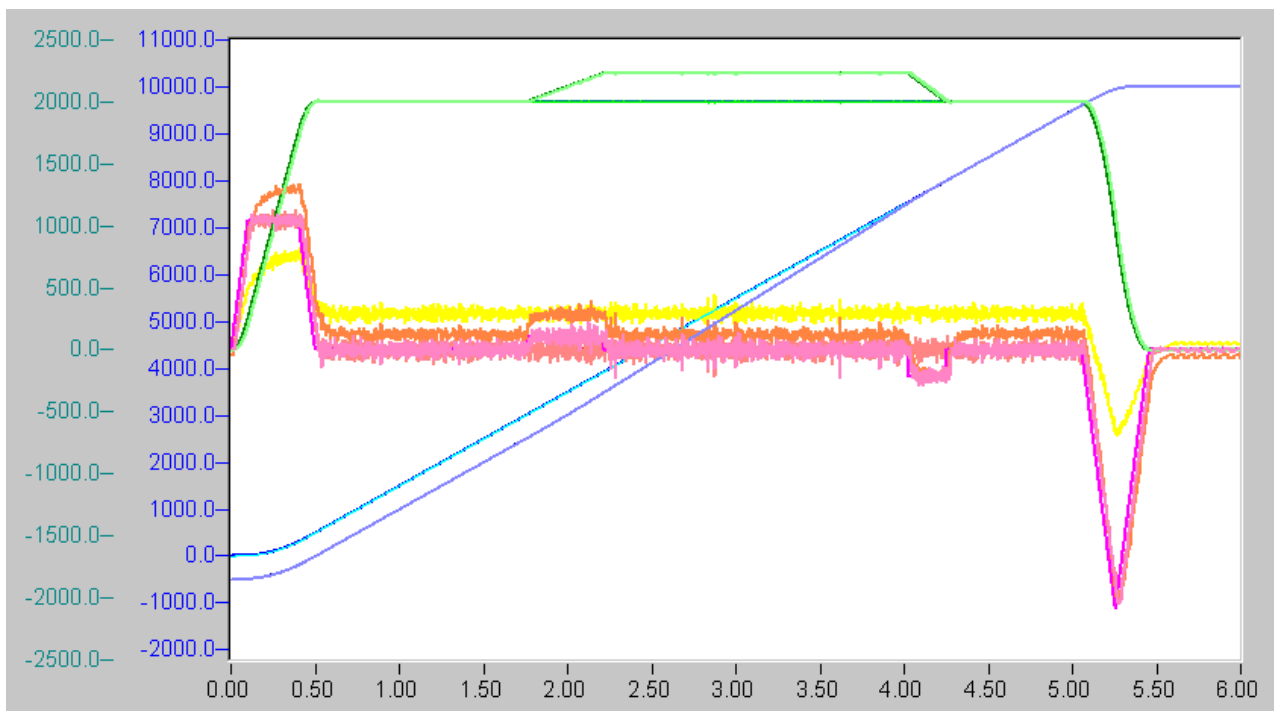
1. D may be negative. In this case, the overall positioning lags behind the basic positioning.
2. Compensation is complete once the distance $L \pm D$ has been travelled.
3. Compensation is a **linear path control**, i.e. during compensation the slave axis will follow all the changes in basic dynamics (e.g. override 0.0).
4. During compensation, axis compensation start is locked, i.e. there is no compensation iteration.
5. End of path during axis compensation stops the compensation.

Special notes for slave axes:

1. If a slave axis is coupled to a virtual axis as master axis, the slave axis can only start and carry out a compensation if the velocity filter time (of the virtual axis) is large enough for the axis not to change direction randomly.
2. Position compensation is possible independently on all axes of a cascade (including master axis).
3. Within a cascade, compensation is passed on to the subsequent slave axes. In case of negative (relative to the axis direction) compensation, movement reversal may occur.
4. In case of several compensations within a cascade, the compensation velocities will be added to each other, and the accelerations will be multiplied to each other.
5. In case of uncoupling of a slave axis during slave or master axis compensation, or of a control axis between master axis and the slave axis to be uncoupled, the basic dynamics of the master axis is maintained and the velocity and the position is carried out continuously differentiable, i.e. the compensation velocity at the uncoupling time is maintained, and the compensation is never completed.
6. Compensation takes place independent of the slave axis coupling factor.

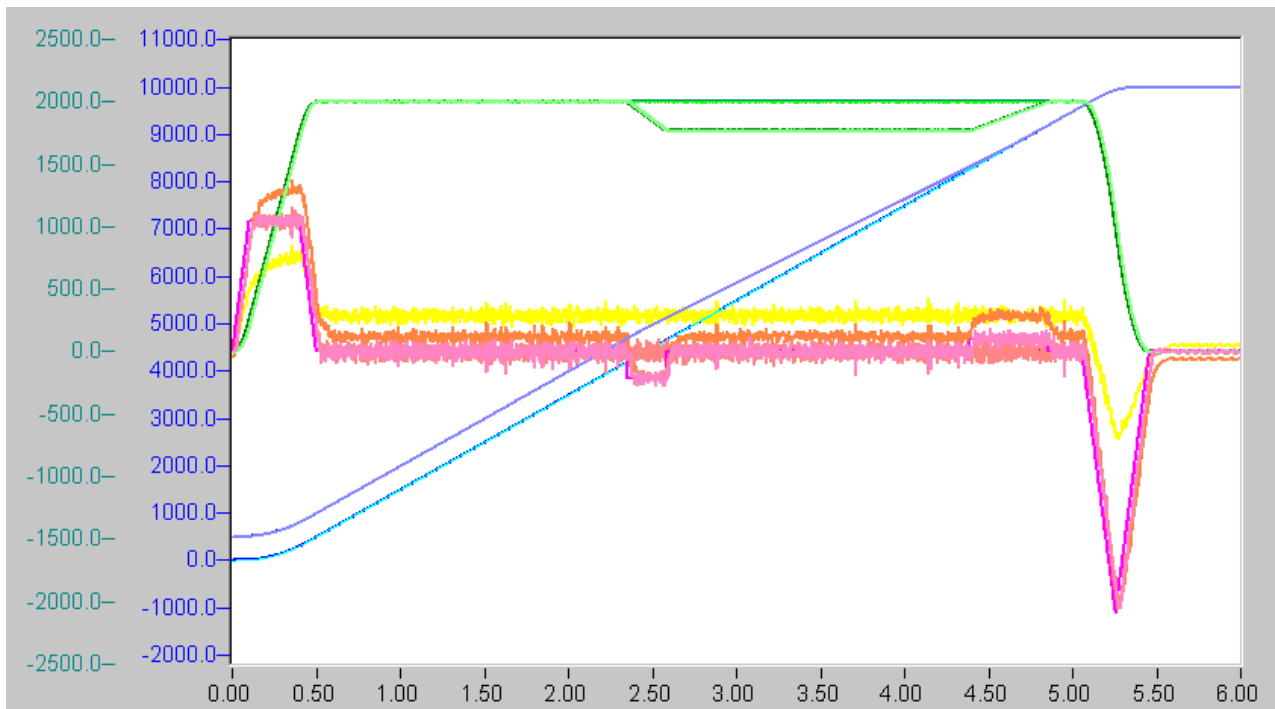
Algorithm

After checking the compensation start parameters, a calculation will be carried out to check whether compensation of the difference D over distance L is feasible with the compensation start parameters. If this is not the case, the maximum possible compensation difference is calculated and activated for D . In any case, the parameters are chosen in such a way that compensation occurs **over the whole compensation length**. The setpoint generation of the compensation is a non-jerk-limited 3-phase-algorithm (**trapezoidal velocity profile**), whereby different ramp slopes are possible.



Compensation: slave is making up fully. Master start from 0.0 mm to 10,000.0 mm with $V = 2000.0$ mm/s. Slave start position -500.0 mm, coupling factor 1.0.

Compensation: $A^+ = 500.0$ mm/s/s, $A^- = 1000.0$ mm/s/s, $V = 300.0$ mm/s, $VP = 2000.0$ mm/s, $D = 500.0$ mm. $L = 5000.0$ mm.



Compensation: slave slows down.

Master start from 0.0 mm to 10,000.0 mm with $V = 2000.0$ mm/s. Slave start position 5000.0 mm, coupling factor 1.0.

Compensation: $A+ = 500.0$ mm/s/s, $A- = 1000.0$ mm/s/s, $V = 300.0$ mm/s, $VP = 2000.0$ mm/s, $D = -500.0$ mm. $L = 5000.0$ mm.

Start parameters

Position compensation: start parameters

Start parameters	Meaning and boundary conditions
Compensation length L	Distance, over which additional positioning occurs ($L > 0.0$)
Compensation difference D	Additional travel distance ($D \neq 0.0$)
Basic process velocity VP	Velocity, to which the velocity increase refers (generally axis target velocity) ($VP > 0.0$)
Velocity increase VK	Permissible velocity increase / reduction relative to the basic process velocity ($VK > 0.0$)
Acceleration $AK+$	Required acceleration increase relative to the global axis acceleration ($AK+ > 0.0$)
Deceleration $AK-$	Required deceleration increase relative to the global axis deceleration ($AK- > 0.0$)

Notes on start parameters:

1. The compensation difference D may be positive (making up) or negative (deceleration). If $D < 0.0$, motion reversal is possible.
2. The compensation refers to a **basic process velocity** VP , which is generally reached by starting the master axis to a target velocity with an override of $x\%$, so that: $v = x / 100$. It is assumed that the master will have reached this velocity when compensation is requested.
3. If the preset basic process velocity VP does not coincide with the amount of the real current local process velocity v , the following applies. If $VP > v$, the actual compensated distance (absolute) will be less than the required one (or the one calculated at the start). If $VP < v$, the actual compensated dis-

tance will generally be equal the required one (or the one calculated at the start). This means that for stop or override 0.0 , the actual compensated distance (absolute) will be less than the required one (or the one calculated at the start).

4. The values $AK+$ and $AK-$ are increases in path acceleration/path deceleration values, i.e. if $D > 0.0$, compensation starts with the acceleration increase $AK+$, whilst, if $D < 0.0$, compensation starts with the deceleration increase $AK-$.

Special notes for **compensation within slave cascade**.

During simultaneous compensation of all axes, let A_{m+} be the acceleration of the cascade master axis, AK_{m+} the compensation acceleration of the master axis, and AK_{sk+} the compensation acceleration of the k -th slave axis, $1 \leq k \leq n$. Then the following applies (Π is the product of all k)

$$A_{m+} \times AK_{m+} \Pi AK_{sk+} \leq A_{max+}$$

where A_{max+} is the acceleration of the master axis that would be movable without slaves. (If an axis does not take part in the compensation, $AK_{sk+} = 1$ is to be used for its acceleration in this equation). Analogous considerations apply for $A-$.

Compensation stop

The function stop terminates the compensation of an axis, if the axis is in compensation mode, but not in the braking phase of compensation, by suitably moving the (temporal) phase limits of the 3-phase algorithm, recalibrating the length of the compensation length L , recalibrating the differential speed and replacing the compensating distance D at the end the compensation by the actual compensated distance.

Interfaces

Compensation start

System Manager: compensation parameters

System Manager: start compensation

PLC function blocks: compensation start

Compensation stop

System Manager

PLC function blocks

3.3 Cyclic Slave Axes

A slave axis, whose position setpoints are determined from

1. the position setpoints of another axis (master axis) through an affine transformation and
2. a coupling factor set through the cyclic PLC-NC interface within the PLC cycle,

is called **cyclic slave axis**. If the coupling factor changes constantly, e.g. because it depends on an external signal, then the slave algorithm is linear in each PLC cycle, i.e. locally linear and globally non-linear.

The following applies: slave set position = coupling factor(PLC) * master set position + offset, where the offset results from the coupling position and the coupling factor is set again and again.

While the coupling of the slave to the master is done offline (master and slave are stationary), the decoupling can be done online: the slave is thereby converted into a master, which continues to move as if it were still a slave, but which can also be stopped independently of the former master.

Each online change of the coupling factor causes a discontinuity in the velocity and thus a load on the dynamics of the axis. In order to avoid excessive strain, the coupling factor arriving via the cyclic interface is therefore smoothed (= filtered). Smoothing is parameterized by limiting the amount of the slave axis acceleration.

Parameters

The parameter `AccBound` in mm/s^2 is the desired acceleration limit of the slave axis. In order to avoid the algorithm becoming instable, a large enough limit must be chosen. Stability is ensured, if $AccBound \geq \max(|Cf(t)|) * \max(A+, -A-)$ is selected, where $Cf(t)$ is the coupling factor at time t and $A+$ and $A-$ are the acceleration limits of the master axis (if the master axis itself is a slave axis, an appropriate calculation must be carried out).

Interfaces

The interfaces consist of the System Manager, PLC blocks and the cyclic NC-PLC axis interface.

Decoupling and new end position

Decoupling is possible in every dynamic state of the master axis - the slave axis is converted into a master axis and continues to travel as if it was still a slave axis, although it can be stopped individually. The coupling flag is reset.

During offline decoupling (decoupling at standstill), the slave axis is converted into a PTP master axis.

When decoupling online (decoupling in motion), the slave axis is converted back into a PTP master axis. A new end position in the direction of movement may be aimed for.

4 Examples

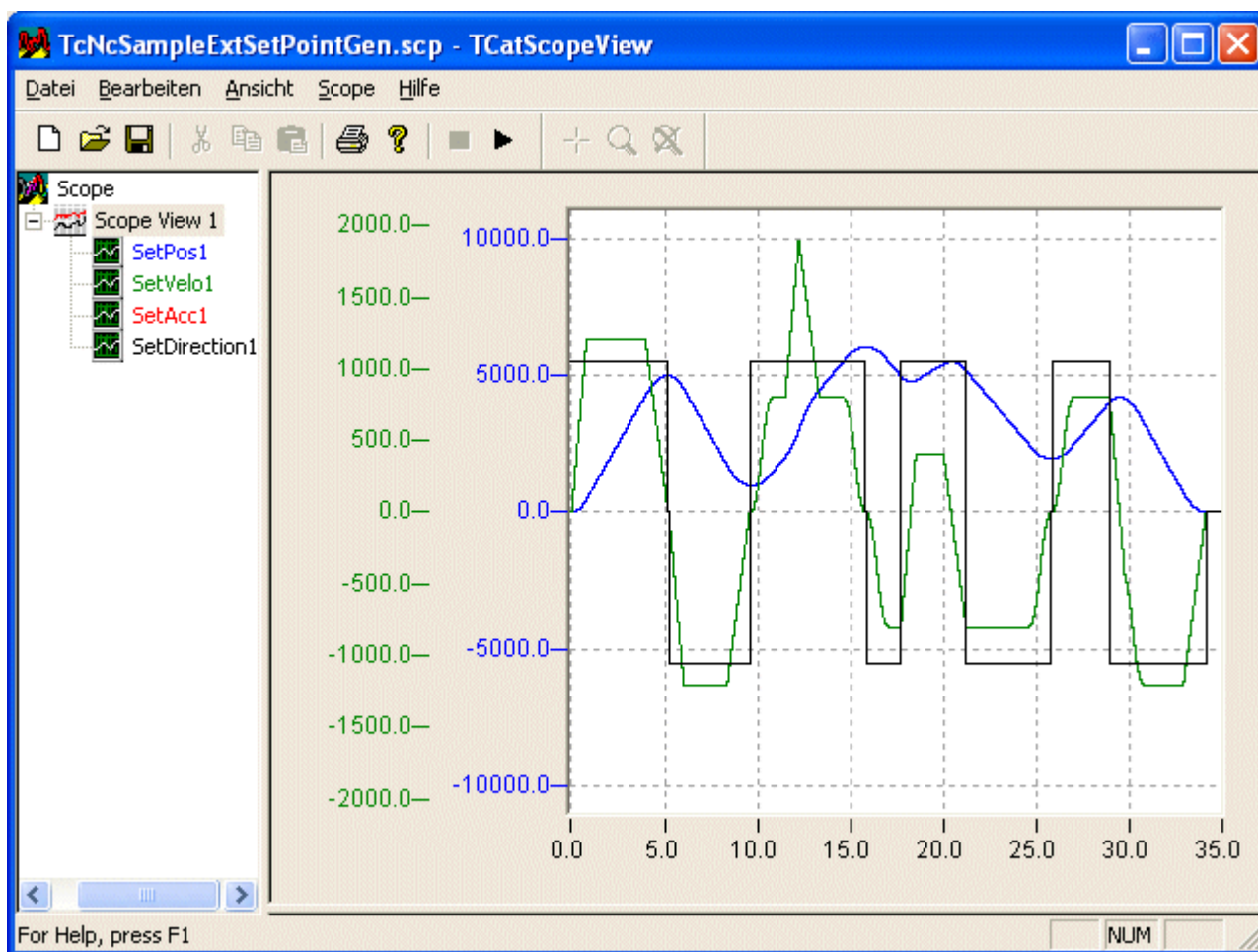
4.1 External Set value generator

TwinCAT NC PTP offers the possibility to program own setpoint generators to move axes. This example shows how a PLC setpoint generator can be integrated into the system. The program implements a 3-phase generator with rectangular acceleration profile.

Further information about external setpoint generator [▶ 23].

The procedure for saving and installing the sample program is similar to the [Axis movement example \[▶ 41\]](#). The program is executable from TwinCAT version 2.9.

- Sample program 1 External setpoint generator: <https://infosys.beckhoff.com/content/1033/tcncptp/Resources/3456014603/.exe>
- Sample program 2 External setpoint generator: <https://infosys.beckhoff.com/content/1033/tcncptp/Resources/3456016011/.exe>



4.2 Drive Controlled Homing (SERCOS)

A SERCOS drive offers the homing function as a drive-controlled sequence. In this case, referencing is therefore not performed by the TwinCAT NC system, but by the drive itself. This sample shows how drive-controlled referencing is initialized and triggered by the TwinCAT PLC.

The procedure for saving and installing the sample program is similar to the [Axis movement example \[▶ 41\]](#).

- Sample program Drive-controlled homing of a SERCOS axis: <https://infosys.beckhoff.com/content/1033/tcncptp/Resources/3456320267/.zip>

The sample program uses the following special error codes:

16#FFFFFF - The sample program does not support parameter scaling in S-0-0044.

16#FFFFFFE - The SERCOS drive has aborted the referencing process with error.

16#FFFFFFD - The referencing process has been completed but the drive has not set the reference flag in S-0-0403.

4.3 Move Axis

TwinCAT NC PTP offers the PLC programmer simple function blocks for the operation of NC axes. This example describes the integration and operation of an axis within the TwinCAT system.

- <https://infosys.beckhoff.com/content/1033/tcncptp/Resources/3456017419/.exe>

The example program uses simulation axes (axes with simulated encoders), and can thus be run on any Windows NT PC without additional hardware. The file *TcNcsample PTPmove.wsm* is loaded into the TwinCAT System Manager, and the system is started with these settings.

- Load the WSM file
- Register the setting via the menu *Actions - Save To Registry*
- Start the TwinCAT system via the gear wheel symbol

TwinCAT Scope is used to be able to examine the result. The supplied setting, *TcNcsample PTPmove.scp*, allows the axis motion to be recorded.

- Load the SCP file into TwinCAT ScopeView
- Start recording via the menu or via the F5 key. (The Scope configuration contains a trigger condition so that, after starting the process with F5, recording will only commence once the axis has started)

Finally, the example program *TcNcsample PTPmove.pro* is loaded into TwinCAT PLC Control, compiled, and started.

- Load the PRO file
- Compile the PLC project via the menu *Project - Rebuild All*.
- Load the PLC project via the menu *Online - Login* into the runtime system
- Start the program via the menu *Online - Run*

Once the program has started, the axis is moved in simulation mode, and recording commences. The image shows position, velocity, and acceleration of the axis.

TcNcsample PTPmove.wsm - TwinCAT System Manager

File Edit Actions View Options Help

General Settings Global Dynamics Online Functions Coupling Compensation

0.0000 Setpoint Position: [mm] 0.0000

Lag Distance (min/max): [mm] 0.0000 (0.000, 0.000) Actual Velocity: [v/s] 0.0000 Setpoint Velocity: [v/s] 0.0000

Override: [%] 0.0000 % Total / Control Output: [%] 0.00 / 0.00 % Error: 0 (0x0)

Status (log.) Status (phys.) Enabling

Ready NOT Moving Coupled Mode Controller Set

Calibrated Moving Fw In Target Pos. Feed Fw

Has Job Moving Bw In Pos. Range Feed Bw

Controller Kv-Factor: [mm/s/mm] 0.1 Reference Velocity: [mm/s] 2200

Target Position: [mm] 0 Target Velocity: [mm/s] 0

F1 F2 F3 F4 F5 F6 F8 F9

Ready Local (172.16.5.71.1.1)

TwinCAT PLC Control - TcNcsample PTPmove.pro - [MAIN (PRG-ST)]

File Edit Project Insert Extras Online Window Help

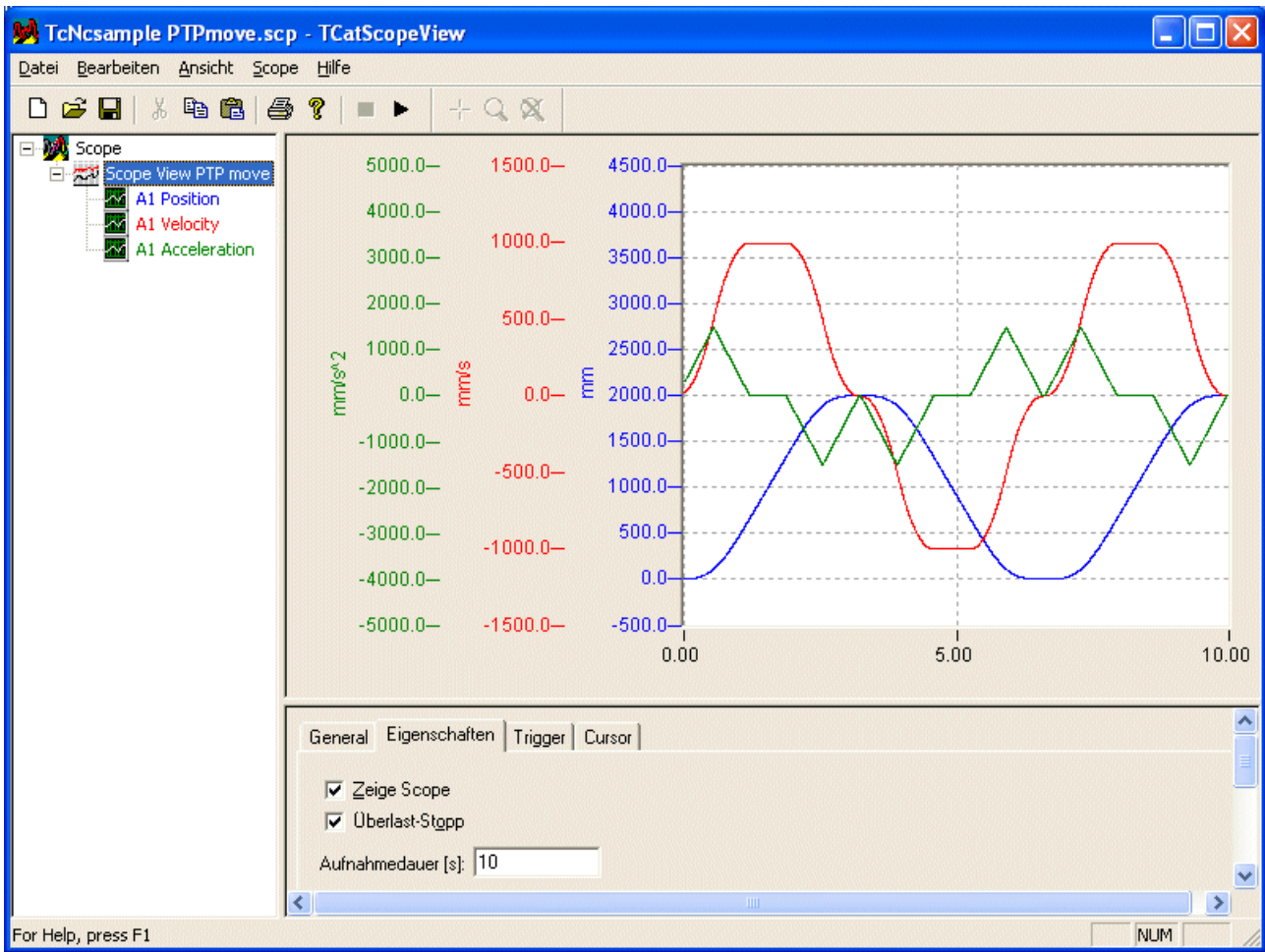
POUs

- AxisEnable (PRG)
- MAIN (PRG)
- MoveAction

```

0001 PROGRAM MAIN
0002 VAR
0003   Position: LREAL;
0004   StartAxis: BOOL;
0005   MoveAxis: MC_MoveAbsolute;
0006
0007   (* enable the axis *)
0008   AxisEnable;
0009
0010   (* move the axis forward and backwards *)
0011   CASE state OF
0012     MOVESTATE_INIT : (* initialisation *)
0013       StartAxis := FALSE;
0014       IF AxisEnable.Ready THEN
0015         state := MOVESTATE_FORWARD;
0016       END_IF
0017
0018     MOVESTATE_FORWARD : (* start to first position *)
0019       Position := 2000;
0020       StartAxis := TRUE;
0021       IF MoveDone THEN
0022         StartAxis := FALSE;
0023         state := MOVESTATE_BACKWARDS;
0024       END_IF
0025
0026     MOVESTATE_BACKWARDS : (* move back to initial position *)
0027       Position := 0;
0028       StartAxis := TRUE;
0029       IF MoveDone THEN
0030         StartAxis := FALSE;
0031       END_IF
0032   END_CASE
0033 END_PROGRAM
    
```

Lin.: 1, Col.: 1 ONLINE [OV] [READ]



More Information:
www.beckhoff.com/tx1250

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

