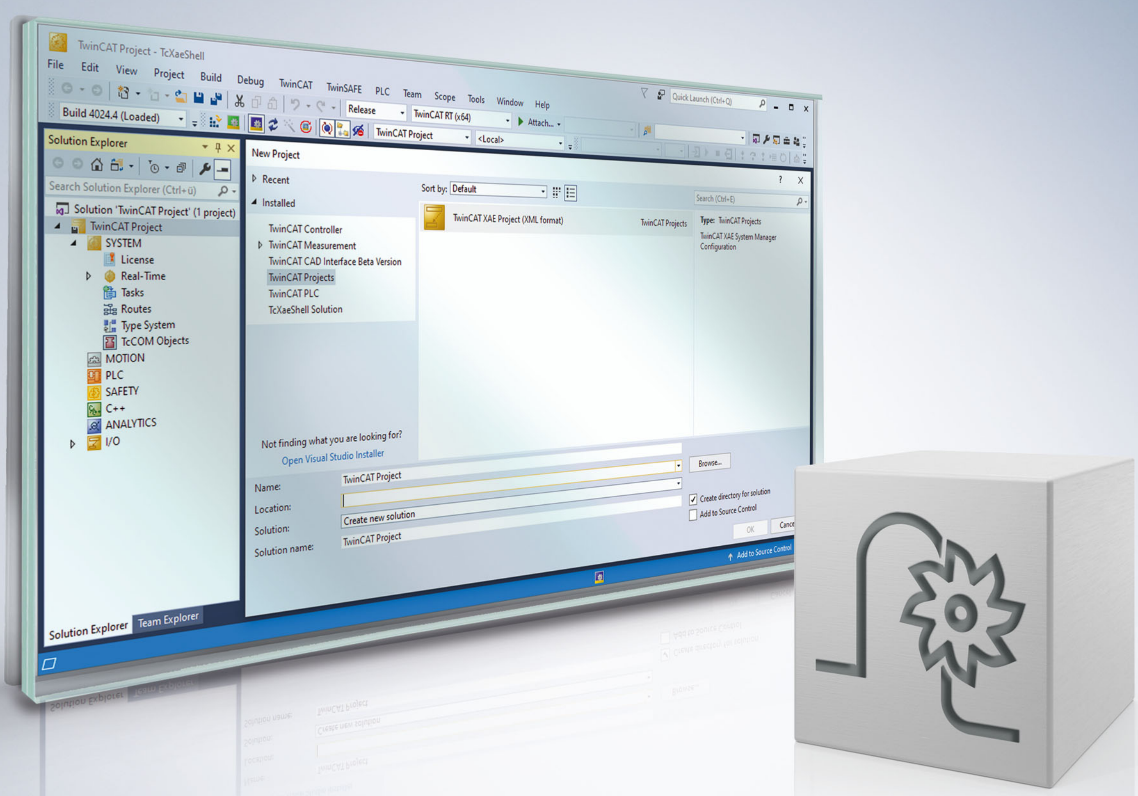
BECKHOFF New Automation Technology

Functional Description | EN

TF5264 | TwinCAT 3 CNC

Conveyor Tracking



3/27/2025 | Version: 1.01

Notes on the documentation

This description is only intended for the use of trained specialists in control and automation engineering who are familiar with the 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 patent protected, in particular by the following applications and patents:
EP1590927, EP1789857, EP1456722, EP2137893, DE102015105702
with corresponding applications or registrations in various other countries.

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

Copyright

General and safety instructions

Icons used and their meanings

This documentation uses the following icons next to the safety instruction and the associated text. Please read the (safety) instructions carefully and comply with them at all times.

Icons in explanatory text

1. Indicates an action.

⇨ Indicates an action statement.

⚠ DANGER

Acute danger to life!

If you fail to comply with the safety instruction next to this icon, there is immediate danger to human life and health.

⚠ CAUTION

Personal injury and damage to machines!

If you fail to comply with the safety instruction next to this icon, it may result in personal injury or damage to machines.

NOTICE

Restriction or error

This icon describes restrictions or warns of errors.

Tips and other notes

This icon indicates information to assist in general understanding or to provide additional information.

General example

Example that clarifies the text.

NC programming example

Programming example (complete NC program or program sequence) of the described function or NC command.

Specific version information

Optional or restricted function. The availability of this function depends on the configuration and the scope of the version.

Table of contents

List of figures

1 Overview

Task

This functionality permits the synchronisation of a linear conveyor system and working on workpieces moving on it.

This function is available as of CNC Build V4.20.0.

This function is an additional option requiring a license.

Effectiveness

Machining can take place with Cartesian and kinematic transformations and tool radius compensation.

Parameterisation

The necessary parameters are assigned both in the axis parameters [▸ 48] (conv_sync.*) and in the channel parameters [▸ 52] (conveyor_sync.*).

The parameter P-CHAN-00650 [▸ 55] must be assigned the value “FCT_DLM” in order to use the functionality.

Programming

There are special programming commands [▸ 26] such as **#SYNC IN** and **#SYNC OUT** for synchronisation and desynchronisation with a linear conveyor.

Mandatory note on references to other documents

For the sake of clarity, links to other documents and parameters are abbreviated, e.g. [PROG] for the Programming Manual or P-AXIS-00001 for an axis parameter.

For technical reasons, these links only function in the Online Help (HTML5, CHM) but not in pdf files since pdfs do not support cross-linking.

1.1 Definition of terms

Term	Description/explanation
T0 coordinate system	Coordinate system of the measuring unit relative to the machine coordinate system
Virtual conveyor axis	Axis that is virtual and parallel to the conveying direction created in the CNC for applications with machines with a static processing unit. This axis is only used by the CNC for synchronisation purposes and does not control a real drive.
Machine with co-moving processing unit	Machine set-up where the processing unit can be moved parallel to the conveying direction by the existing axes.
Machine with static processing unit	Machine set-up where the processing unit is static in the conveying direction and the workpieces are moved together with the conveyor system through the processing unit.
Conveyor system	Term that includes all conveyor units such as conveyor belts, automated guided vehicles or other conveyor systems.
MCS	Machine coordinate system

2 Description

The conveyor tracking functionality permits synchronisation with a linear conveyor system. This term covers all conveyor systems that move objects linearly within the synchronisation range. Conveyor tracking allows objects that are moving on a belt to be manipulated or machined.
Possible applications include woodworking, laser machining and workpiece handling, such as pick & place modules.

When workpieces are machined, e.g. by laser cutting, the conveyor tracking functionality offers the following benefits:

1. It permits the machining of throughfeed workpieces that are larger than the actual machine workspace. This means that the workspace can be designed smaller for the same workpiece size.
2. Throughput can be increased by machining carried out on a moving workpiece.
3. Material can be better utilised by machining sheet fed from a steel strip coil and waste is reduced compared to conventional sheet machining.

2.1 Operating principle

Synchronisation is always with the 1st axis in the channel (axis index = 0). Two coordinate systems of the conveyor system are used here.

- The fixed coordinate system PCS_0 is the reference system for synchronisation. The X axis of PCS_0 must point in the positive conveying direction of the linear conveyor. After synchronisation, the coordinate system co-moves with the conveyor system.
- The coordinate system PCS_1 is the reference system for programming the intersecting movement.

The conveyor can be offset and rotated in space as required in relation to the machine coordinate system. This position of the conveyor relative to the machine origin or to the machine coordinate system (MCS) must be configured using transformation T_0 (P-CHAN-00368 [▶ 53] to P-CHAN-00373 [▶ 54]). This describes the position of PCS_0 in the MCS.

Synchronisation is actuated by a trigger. For example, this can be done by a camera system or by an object moving through a light barrier. First, the current measured value of the conveyor system at the time of the trigger event is detected. At time t_0 of the measurement, the co-moving workpiece coordinate system PCS_1 is identical to the stationary coordinate system PCS_0. The coordinate system then co-moves with the conveyor system. Later, at time t_1, the coordinate system PCS_1 is then moved by the distance covered by the conveyor system between t_0 and t_1.

The figure below shows the relationship between these coordinate systems and the schematic structure of the conveyor tracking functionality:

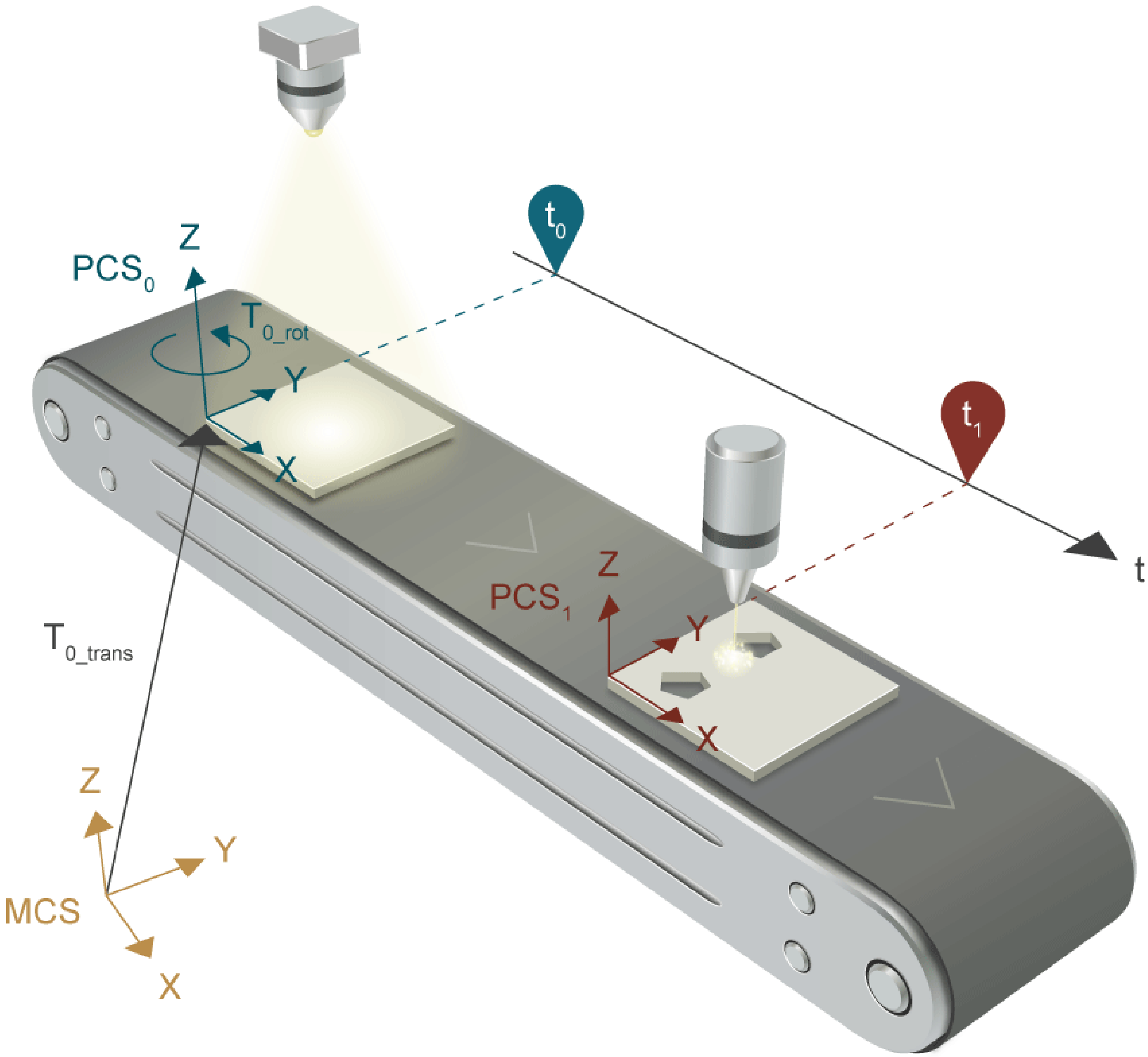


Fig. 1: Schematic structure

After synchronisation, the workpiece coordinate system PCS_1 is co-moved with the conveyor in the conveying direction until desynchronisation. The intersected path is then programmed in the co-moving coordinate system. After synchronisation is completed, other coordinate systems can be configured and selected based on PCS_1. Synchronisation can take place in the positive and negative conveying directions.

Here, the linear conveyor is the master axis and must be configured in the CNC as a spindle. It is also possible to synchronise multiple channels simultaneously with a physical conveyor system.

Synchronisation with the linear conveyor means that, depending on the conveyor velocity v_{conv}, the full dynamics are not available for the intersected movement.

The software limit switches continue to be monitored while synchronisation is active. In addition, contour intersections with workspace limits in the opposite conveying direction are monitored and adjusted by reducing the programmed velocity so that the end position is not overshot.

Kinematic transformations can be selected before synchronisation, in particular for handling tasks.

See the appropriate section for details on Programming [▶ 26] the conveyor tracking functionality.

2.1.1 Synchronisation with the conveyor system

The NC commands #SYNC IN [▶ 26] and #SYNC OUT [▶ 26] are provided for synchronising and desynchronising the coordinate system with the conveyor.

The synchronisation process first waits for the measurement of the actual position value of the master axis (conveyor system) at the time when the workpiece is measured. The measurement is sent over the configured axis measuring interface and is commanded by the PLCopen command MC_TouchProbe [▶ 63]. This command can be issued either by programming in the NC program or by the PLC. After the command is issued, the measurement is event-controlled based on the configured trigger. For example, the trigger may be issued by the PLC, e.g. a light barrier or a camera.

When the measurement is complete, the actual position at the time of measurement is used to couple the co-moving coordinate system PCS_1 with the conveyor belt.

The NC command **#SYNC IN** does not initially trigger a synchronisation movement with the conveyor. It only causes a change in the stationary basic coordinate system PCS_0 so that the co-moving coordinate system PCS_1 can be tracked on the conveyor. The synchronisation movement is only started in the NC program when the next movement on the workpiece contour is initiated.

The first movement after the NC command #SYNC IN is the synchronisation movement with the conveyor and must be of type G00 or G01.

Synchronisation takes place with the coordinates programmed in the motion block in the co-moving PCS_1 coordinate system. The PTP movement towards the programmed target point is started simultaneously in all axes. In general, the axes do not reach the target point at the same time.

This synchronisation movement only takes place after the measurement.

Additional coordinate systems can only be programmed after this synchronisation movement. It is important to always program the selection of a kinematic transformation outside #SYNC IN/ OUT.

The coupling to the workpiece on the conveyor is released when desynchronisation takes place with the NC command **#SYNC OUT**. As with #SYNC IN, this command does not yet cause a desynchronisation movement but only changes the programming coordinate system to the machine coordinate system (MCS). The desynchronisation movement is programmed in the first block after #SYNC OUT.

The desynchronisation movement must be programmed as an independent movement.

This independent movement with identifier “INDP_ASYN” is used to move the axes in PTP mode to the target coordinate programmed in the MCS after a #SYNC OUT command. This movement takes place as a single-axis movement, meaning that the axes do not follow a defined path. This synchronisation movement can be interrupted by a following #SYNC IN to permit an on-the-fly change to a new synchronisation process.

When #SYNC IN is programmed, the program automatically switches over to a jerk-limited slope. The acceleration profile cannot be switched over within the #SYNC IN/OUT function.

2.1.1.1 On-the-fly synchronisation/desynchronisation

The synchronisation/desynchronisation functionality allows the programmed target point of the synchronisation/desynchronisation movement to be approached at a relative velocity between the workpiece and the tool. This permits an on-the-fly change in the machining process or in the following movement. The relative velocity between the workpiece and the tool at the point of synchronisation is defined in the FEED_CONT parameter in either the #SYNC IN [▶ 26] or #SYNC OUT [▶ 26] NC command, and reduced as required.

The synchronisation movement always ends at acceleration 0.

2.1.1.2 Cross-program synchronisation/desynchronisation

There are 2 possible approaches for the continuous machining or handling of workpieces on a co-moving linear conveyor.

- In the first case, an NC program is started once. This program then carries out synchronisation and desynchronisation with the conveyor in a loop. In addition, the individual workpiece machine operations are started as subroutines of this program or are directly included in it.
- In the second case, a separate NC program containing the synchronisation function with the workpiece and the machining operations is started for each workpiece.

  In this case, the individual programs can be started either by a main program or a dispatcher program running in a separate NC channel, by the job manager [FCT-M10] or directly by the PLC using PLCopen Part 4 function blocks.

  In order to permit on-the-fly transition to the next NC program, the option for a fast program start "ProgStartFast [▶ 61]" must be set at program start. This suppresses the position request at program start. In addition, the program start option "SuppressProgStartInit [▶ 61]" must be set when the NC program is started. The synchronisation movement is fully completed without an on-the-fly transition if:
  - the program start option "ProgStartFast" is not set.
  - the program start option "SuppressProgStartInit" is not set.
  - A motion block with a relevant motion path is programmed before #SYNC IN.

2.1.1.3 Configuring the probing signal

Synchronisation is based on the measured value of the linear conveyor at the time of the positive edge of the measurement trigger signal.

The measured value is detected based on the configuration of the probing signal parameters P-AXIS-00516/ P-AXIS-00517 and P-AXIS-00518. The probe signal is initiated by the PLCopen command MC_TOUCH_PROBE [▶ 28] and the measured value is detected as soon as the configured trigger event occurs. The measured value is then detected directly by the CNC via the drive interface unless otherwise configured, but can be alternatively specified by the PLC.

2.1.2 Working on the conveyor system

Machining or handling takes place in the co-moving workpiece coordinate system PCS_1. The programmed path is intersected on the conveyor movement. This means that programming a feed rate of F=0 mm/min in the co-moving system PCS_1 results in a velocity that corresponds to the conveyor velocity v_{conv} in the stationary coordinate system PCS_0. In this case, the TCP moves at conveyor velocity.

This superimposition results in different maximum velocities on the workpiece and in the opposite conveying direction.

- In the conveying direction:

$$v_{max,workpiece} = v_{max,machine} - v_{conv}$$

- In the opposite conveying direction:

$$v_{max,workpiece} = v_{max,machine} + v_{conv}$$

Starting from PCS_1, other coordinate systems can be programmed using the #CS command. The workspace limits continue to be monitored as long as synchronisation with the conveyor is active. If a programmed contour crosses the workspace limit in the conveying direction, it can no longer be tracked and the program is aborted with an error message.

If the programmed contour crosses the workspace limit in the opposite conveying direction, the movement is adjusted so that the contour can be traversed without crossing the workspace limit.

See Monitoring the workspace limit in the opposite conveying direction [▶ 16].

2.1.3 Monitoring the workspace limit in the opposite conveying direction

NOTICE

This function may not be used to implement safety-related functions.

As soon as the synchronisation process is completed, the function for automatically maintaining the workspace limit in the opposite conveying direction is activated. This reduces the conveying velocity to prevent crossing the workspace limit in the opposite conveying direction. However, this type of intervention by reducing path velocity can only take place if the movement of the X axis (parallel to the conveyor) in the co-moving coordinate system PCS_1 takes place in the opposite conveying direction. The contour is checked for intersections with the workspace limit.

If an intersection is detected in an NC block, the path velocity in this NC block is reduced in the co-moving coordinate system PCS_1 to the velocity v_{conv} of the linear conveyor. As a result, the TCP is stationary in the stationary coordinate system PCS_0. Therefore, in this case, the movement of the X component along the contour of this NC block is solely implemented by the movement of the conveyor. The behaviour of this function is shown in the figure below.

The time taken to traverse the contour is increased by reducing the feed rate for the colliding NC segment. However, the increase in machining time can be optimised, as described in the next section.

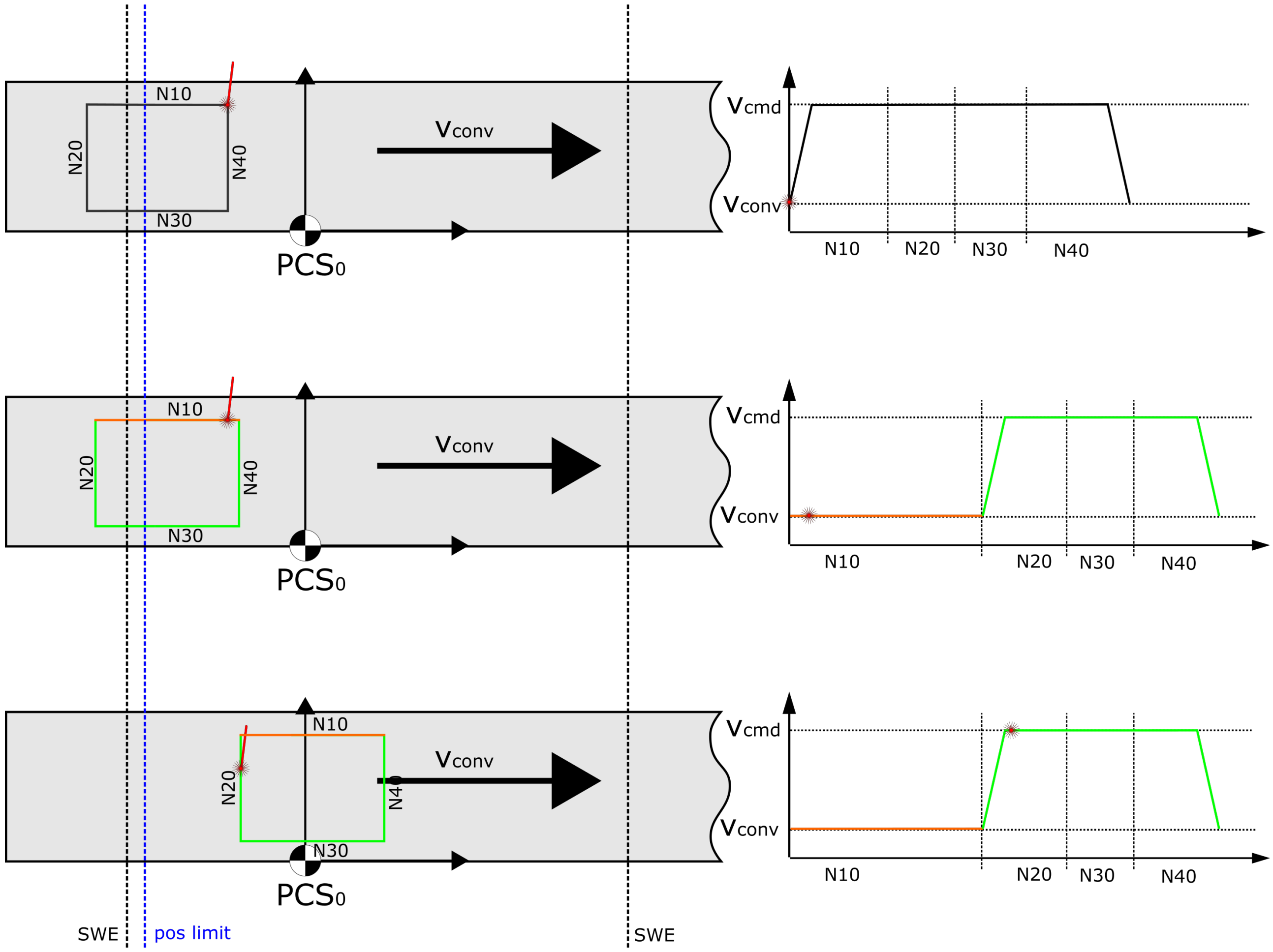


Fig. 2: Operating principle of the end position monitoring function

Optimised workspace limit monitoring in the opposite conveying direction

Optimised end position monitoring can be activated in the master axis by the axis parameter P-AXIS-00555 [▸ 50] (conv_sync_optim).

Here too, the programmed contour is monitored in the path preparation function for intersections with the configured workspace limit (P-CHAN-00374 [▸ 54]).

If an intersection is detected, the optimum braking instant and the corresponding braking profile are calculated so that path velocity reaches the exact conveyor velocity v_{conv} when the workspace limit is reached.

With linear blocks, this means that the tool comes to a standstill exactly at the workspace limit and remains there.

With circular blocks, the tool is pushed away from the workspace limit due to the tangential velocity on the circle.

Optimised end position monitoring reduces the time required to move along the colliding NC segment since the segment is traversed at the programmed feed rate until the workspace limit is reached.

The behaviour of optimised workspace limit monitoring is shown in the figure below. The star shows the calculated optimum point on the contour at which the conveyor velocity is reached exactly at the workspace limit.

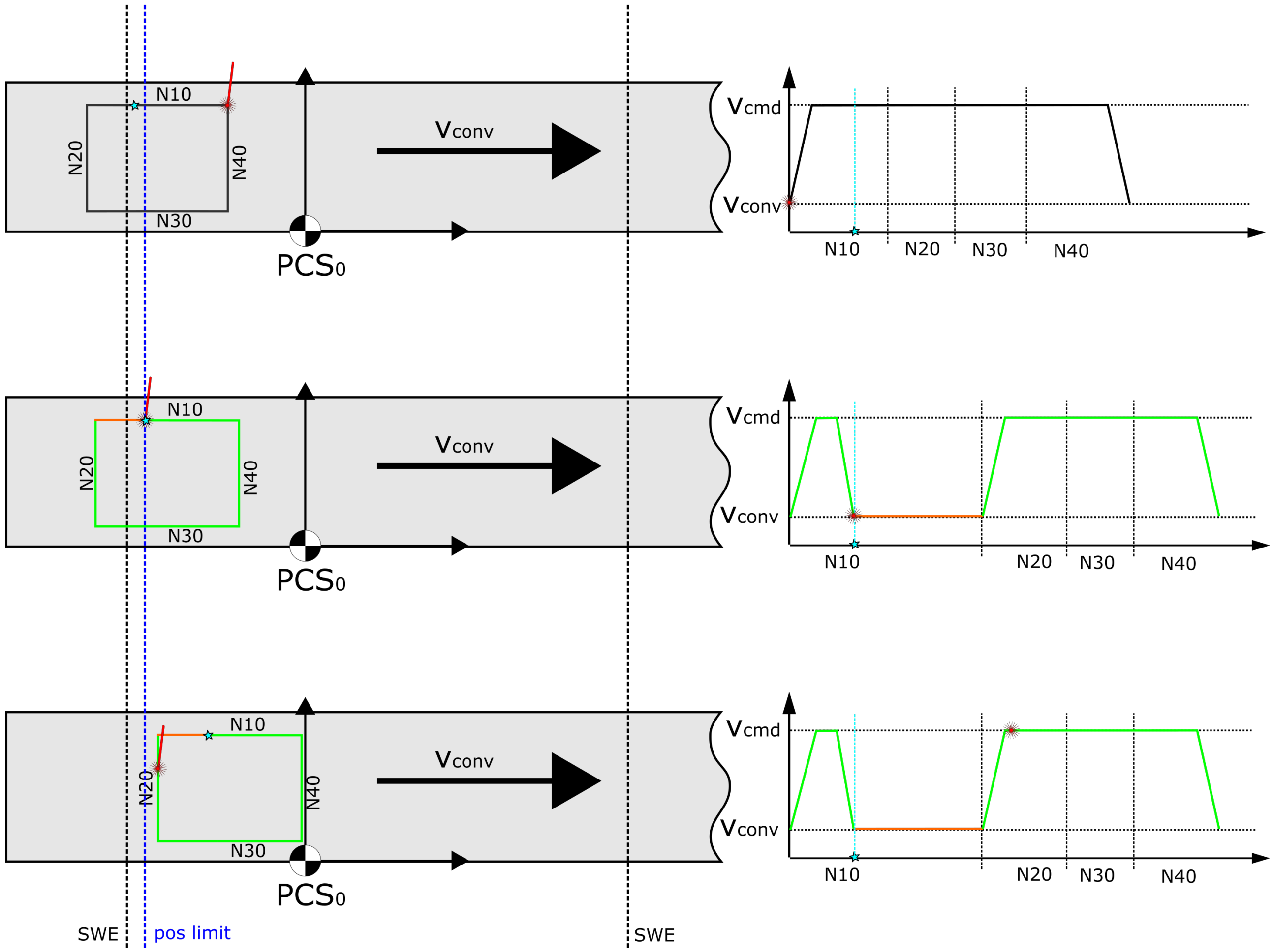


Fig. 3: Operating principle of optimised end position monitoring

Compared to simplified end position monitoring, optimised end position monitoring requires greater computing power and therefore calls for more powerful control hardware.

Fluctuations in conveyor velocities with optimised workspace limit monitoring

It is normal for the conveyor velocity to fluctuate. NC blocks with reduced feed rate due to workspace limit monitoring may cause the configured workspace limits to be crossed.

The parameter P-CHAN-00374 [▸ 54] (pos_limit) can be used to configure a position offset of the workspace limit to ensure that the workspace limit is reliably maintained even if fluctuations in conveyor velocity occur. This shifts the monitored workspace limit.

In addition, a safety factor (P-CHAN-00366 [▸ 53]) can be defined to reduce the velocity and acts directly on the velocity that reduces the feed rate of the colliding segments. The following applies:

v_{path}= P-CHAN-00366 * v_{conv}

The value range of the factor is between 0 and 1. A setting of 0.95 has proven to be most effective in practice.

2.1.4 Filtering the actual values of the conveyor axis

Synchronising and tracking the co-moving workpiece coordinate system PCS_1 are based on the actual values of the conveyor axis. The actual values are more or less extremely noisy depending on the design of the linear conveyor.

Due to the coupling, this noise has a negative effect on the contour error when synchronisation is active. This is why the actual values are smoothed by a filter. Two different filter configurations are available for this purpose and can be switched over depending on the operating state of the conveyor. Filtering also reduces the risk of exceeding the dynamics due to velocity jumps in the probing signal.

In normal mode at constant conveyor velocity, the strongly smoothed conveyor velocity is used for synchronisation. The longer filter length defined in P-AXIS-00623 [▶ 51] results is a greater time delay and is less dynamic. When the coupled system is in normal mode, this filter results in only very slight undesirable excitations in the manipulated variable caused by noise in the actual position signal of the conveyor axis.

The MCV_SetReducedFilter [▶ 65] function block in the PLC can be used to switch from strong to weak filtering to minimise the contour error in dynamic phases and especially in the event of a movement stop. This weak filter set-up consists of a sinc (ideal low-pass) filter (P-AXIS-00625 [▶ 51]) followed by an average value filter (P-AXIS-00624 [▶ 51]).

Both filters generate a longer or shorter delay time due to the set filter order. The weak filter reacts more dynamically due to the smaller filter width and also generates a shorter delay time. The filtered velocity is used for feedforward control of the delay time.

The individual filters can be configured in the axis parameters and, while the conveyor axis is in operation, the filter parameters can be adapted using the #MACHINE DATA command.

The figure below shows an example signal curve with a noisy probe signal and the filtered signal with a filter order of 10 (parameter conv_sync.order_v_filter = 10).

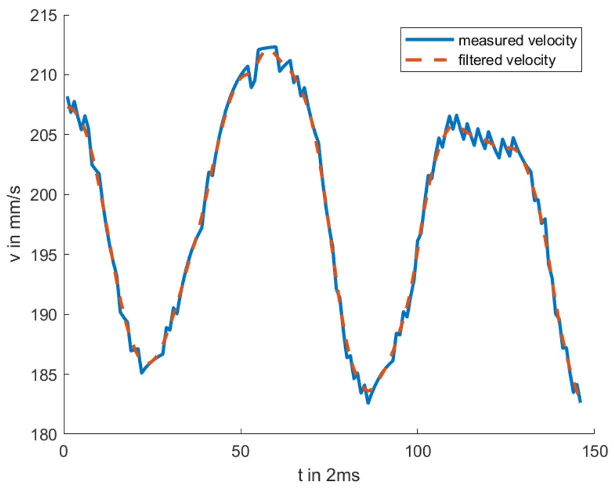


Fig. 4: Example of signal curve with a noisy probe signal and filtered signal

2.1.5 Delay time compensation

Delay times occur at various points between the latch moment of the actual position value and generation of command values for the machine.

Delays occur during transfer on the fieldbus:

- when the latch moment of the actual value is detected by the drive and
- when the command value is transferred to the drive

Delay times occur in the CNC due to

- the filters set for the actual values of the conveyor (see Filtering the actual values of the conveyor axis [▸ 19]) and also
- individual cycles may be delayed due to internal data transfer

Internal delays can be minimised by optimising the scheduling of “SWITCHED” in the CNC (see CNC scheduler).

The figure below shows a schematic diagram of the effects described and the resulting error in delay time by means of an example workpiece:

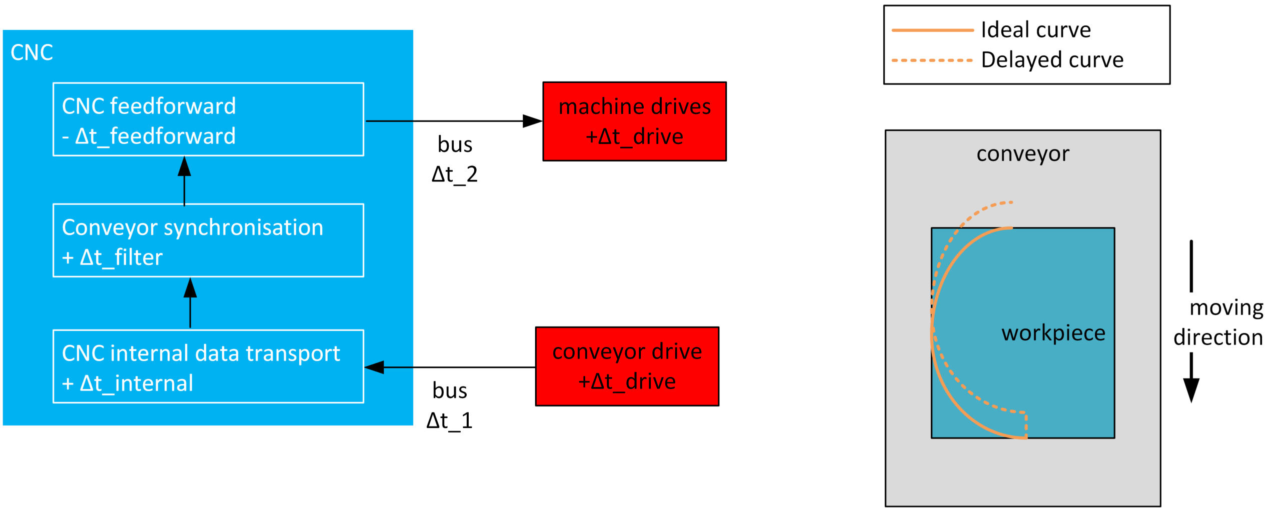


Fig. 5: Schematic diagram of delay times

The CNC has a feedforward control to compensate for errors caused by delay times. The axis parameter P-AXIS-00626 [▸ 51] defines the delay time compensation and the position uses the delay time for feedforward control at the current filtered velocity.

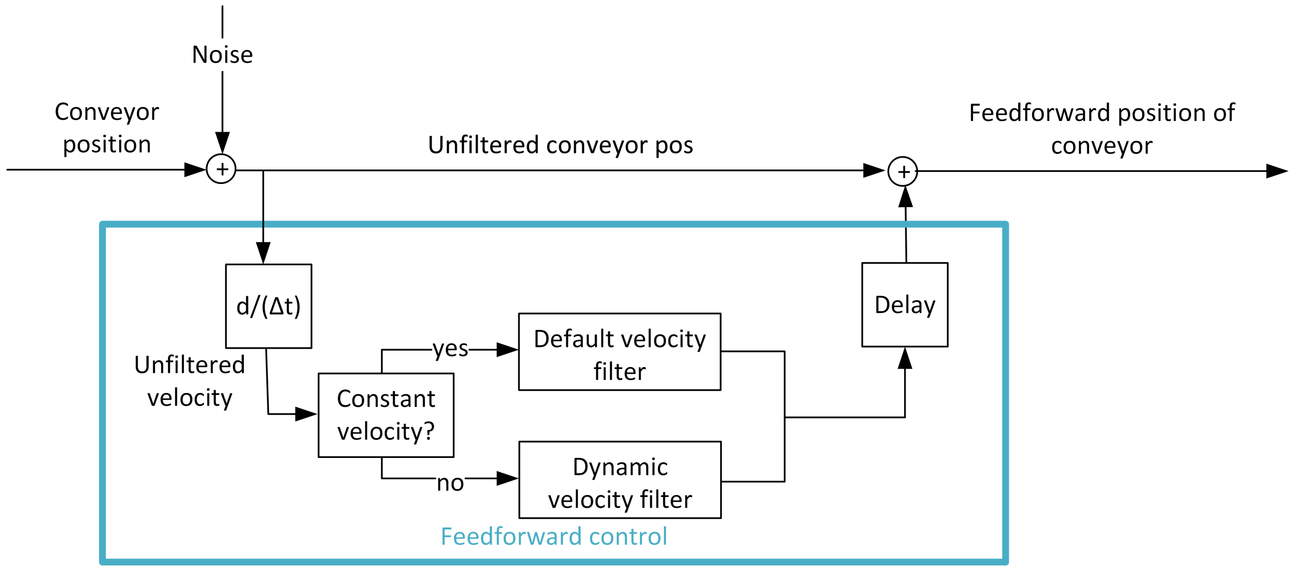


Fig. 6: Block diagram of feedforward control and velocity filter

Feedforward control assumes that the current filtered conveying velocity is constant. This is why conveyor velocities that are as constant as possible lead to better results. If the conveyor velocity fluctuates, the resulting contour error increases at higher conveyor velocities with the same delay time.

The CNC indicates the display variable synchro_lag to detect the delay time. This variable and suitable programs can measure the delay time at start-up [▶ 34].

2.1.6 Synchronisation without a parallel axis to the conveying direction

The conveyor tracking functionality offers the option of synchronisation for machines that have no axis in the direction of conveyor movement and therefore cannot co-move with the linear conveyor. For example, this is the case with machining units that can only be fed at right angles to the conveying direction. In this case, a virtual axis parallel to the conveying direction must be created as a simulation axis in the first main axis index (P-AXIS-00020). The absence of a parallel axis must be configured in the channel parameters using the conveyor_sync.x_virtual parameter (P-CHAN-00364 [▸ 52]).

Since the processing unit cannot move in the conveying direction, the relative movement between the tool and the workpiece is only carried out by the linear conveyor. This results in the following restrictions which must be taken into account when programming contours:

- The feed rate is dependent on the conveyor velocity and cannot be influenced independently.
- The programmed contour must be programmed completely in the opposite conveying direction.
- The programmed contour must be designed so that it does not overshoot the dynamic limits of the machine since it is not possible to reduce conveyor velocity to maintain these limits.
- The programmed contour must be twice continuously differentiable. Therefore, it may not contain any knee angles and must be continuously curved when the direction is changed. Smoothing methods such as polynomial contouring or spline interpolation can be used for this purpose.

The contour must also be continuous at the edges. To achieve this, the tool must be pre-positioned tangentially or the contour must be programmed so that the edges are tangential to the conveying direction.

2.1.7 Restrictions

The conveyor tracking functionality has the following restrictions:

- No coordinate systems may be selected during synchronisation or desynchronisation. This means that coordinate systems selected before synchronisation must be deselected before #SYNC IN by using #CS OFF. After synchronisation with the conveyor is terminated, coordinate systems may be selected for working on the moving workpiece. However, the coordinate systems must then be deselected using #CS OFF before desynchronisation.
- During the machining process, the conveyor velocity and programmed feed rate must be set so that machining is completed before the workspace limit is reached in the conveying direction.
- Synchronisation can only take place on linear conveyors or conveyor systems that move linearly within the synchronisation range.
- Only one axis can be configured as conveyor axis in an NC channel.
- The intersected movement uses a special jerk-limited slope type. It is not permitted to change the slope type using the #SLOPE command in the synchronised state. When synchronisation takes place, the slope type is automatically switched over to the jerk-limited slope. After desynchronisation, it is reset to the default slope type configured in the channel.
- When a kinematic transformation is used, such as with articulated arm, delta or SCARA robots, dynamic overshoots may occur when synchronisation with the conveyor belt is active. These overshoots may be caused by fluctuations in conveyor belt velocity during the movement. In this case, the belt velocity or the dynamic axis limits can be reduced.

2.2 Application examples

Possible applications for the conveyor tracking functionality are described in the subsections below. Each application example is supplied with an example configuration and an example NC program.

2.2.1 Laser machining on moving sheet

The figure below shows the schematic diagram for laser cutting using a Cartesian portal machine to cut a moving sheet unrolled from a coil.

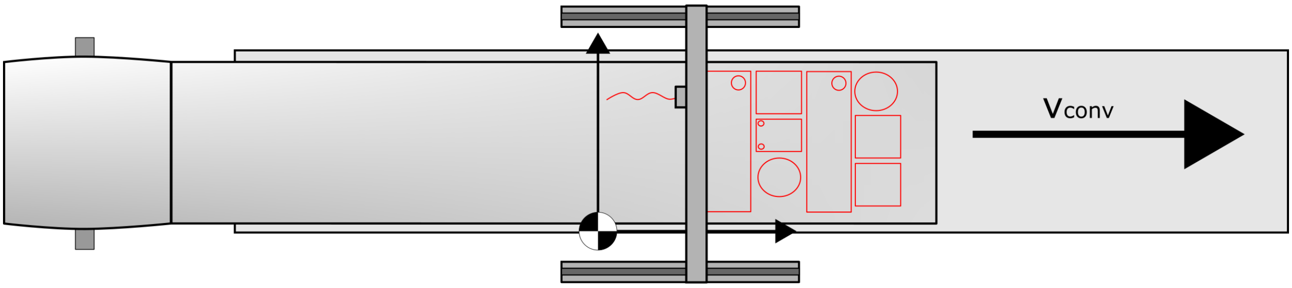


Fig. 7: Example application - laser cutting on a moving sheet

For configuration and program example, see Application example: Laser machining on moving sheet [▶ 38].

2.2.2 Pick & place

The figure below is a schematic diagram of a delta kinematic (X1-X2-X3) to pick & place objects or workpieces that are picked from the linear conveyor.

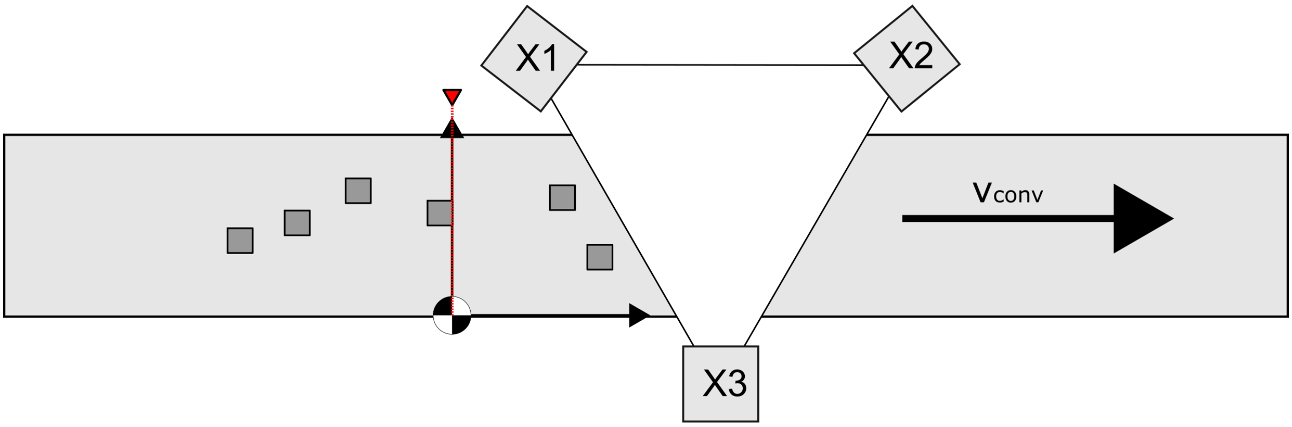


Fig. 8: Example application - pick & place on the moving workpiece

For configuration and program example, see Application example: Pick & place [▶ 40].

2.2.3 Machining without co-moving unit

The application example below shows the use of a band saw to saw objects conveyed on a linear conveyor. The aim of this application is to saw a curved workpiece contour.

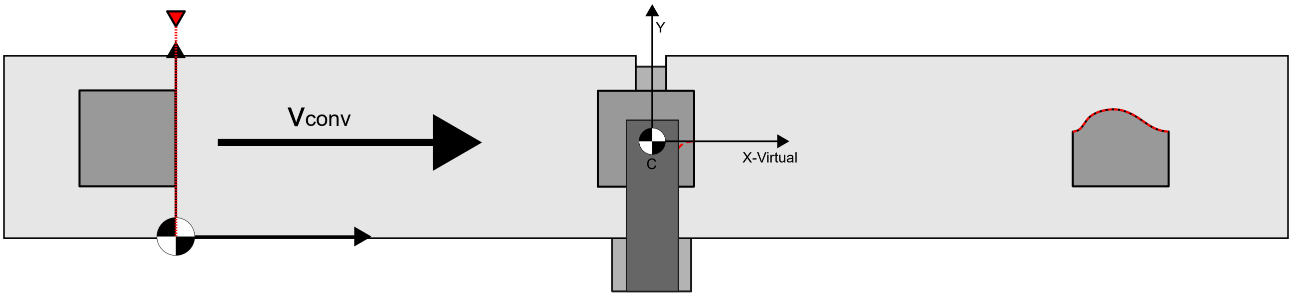


Fig. 9: Application example – machining without co-moving processing unit

The saw can be fed in the Y direction and the saw blade can be rotated about the Z axis (C axis). Since the machine has no axis in the conveying direction, the relative movement between the tool and the workpiece is only generated by the linear conveyor.

For configuration and program example, see Application example: Machining with static processing unit [▶ 43].

2.3 Behaviour in the event of an error

If an error occurs in the NC channel while synchronisation with the linear conveyor is active, the intersected path reacts depending on the error reaction class of the error that occurred. Synchronisation with the conveyor system is maintained. In this case, the velocity is the co-moving system PCS_1 is reduced to $v_{path}=0$. The machine then moves with the conveyor system at the conveying velocity.

If the intersected movement in the co-moving coordinate system is stopped due to an error, the machine moves at the conveying velocity in the stationary coordinate system until the limit switch position is reached.

If the linear conveyor is stopped by the PLC, the machine also comes to a standstill.

3 Programming

The **#SYNC IN** command indicates that the following machining operations are carried out in the moving coordinate system of the workpiece. The synchronisation movement is only started by the next motion block. Synchronisation takes place with the coordinates programmed in the motion block in the co-moving PCS_1.

The target point is simultaneously approached in multiple axes with a PTP motion. In general, the axes do not reach the target at the same time.

Syntax:

#SYNC IN [[CONVEYOR=..] CONV_VEL=.. [VEL_RESOLUTION=[<m/min>, <m/s>, <mm/min>, <mm/s>, <um/min>, <um/s>]] **[FEED_CONT=..] FEED_CONT_AUTO]**

CONVEYOR=..	The axis designated for synchronisation is defined by the CONVEYOR parameter. The axis name must be specified. If CONVEYOR is not specified, the axis with the logical axis number in P-CHAN-00362 [▸ 52] is used.
CONV_VEL=..	Programmed command velocity of the conveyor axis in [mm/min]. The velocity specified is used in the CNC to control whether the conveyor axis is moved at the required velocity. If the actual velocity is 10% higher than specified, error ID 50653 is output.. If a higher velocity than the maximum permissible velocity of the machine in the conveying direction is programmed, error ID 50587 is output.
VEL_RESOLUTION	Optional value of velocity resolution. The following resolutions are provided: m/min, m/s, mm/min, mm/s, um/min ad um/s. If the parameter is not specified, the velocity is interpreted as mm/min by default. This parameter only acts on the individual command and does not act globally for the following commands.
FEED_CONT=..	Definition of the relative velocity in [mm/min] used to terminate synchronisation and start contour machining. If FEED_CONT is not specified, synchronisation is performed at conveyor velocity. The relative velocity between the workpiece and the tool is 0.
FEED_CONT_AUTO	Identifier for synchronisation at the maximum permissible dynamic relative velocity.

The movements below after the #SYNC OUT command are listed in the machine co-ordinate system.

Syntax:

#SYNC OUT [[FEED_CONT=..] FEED_CONT_AUTO]

FEED_CONT=..	Definition of the relative velocity in [mm/min] used to terminate desynchronisation. If FEED_CONT is not specified, the desynchronisation movement terminates at velocity 0.
FEED_CONT_AUTO	Identifier for synchronisation at the maximum permissible dynamic relative velocity.

If you want to synchronise directly with a new workpiece on-the-fly without desynchronisation, the NC commands #SYNC IN/OUT must be programmed again with the appropriate parameters followed by a motion block.

Independent movement after #SYNC OUT

Syntax:

<axis_name> **[INDP_SYN | INDP_ASYN G00 | G01 POS=.. FEED=.. INTERRUPTIBLE]**

<axis_name>	Name of the axis to be moved in the MCS coordinate system after desynchronisation
INDP_SYN	Identifier for synchronous (blockwise) independent axis motion. The transition to the next block is only executed when all axes have reached their end positions. Must always be programmed as first keyword.
INDP_ASYN	Identifier for asynchronous (cross-block) independent axis motion. There is no synchronisation to end positions. Must always be programmed as first keyword.
G00/G01	Rapid traverse/linear interpolation
POS=::	Axis position in [mm, inch]
FEED2=..,	Axis-specific feed rate in [mm/min]
INTERRUPTIBLE	Identifier that this movement is interruptible.

INTERRUPTIBLE in conjunction with INDP_ASYN is ignored.

The program example below shows the full sequence for machining a moving workpiece. It includes reading the measured value for the machined workpiece, synchronisation, machining the workpiece followed by desynchronisation and positioning.

Machining a single workpiece with #SINC IN/OUT

```
%kontur_2
N0010 G0 X0 Y0 Z200
N0020 T1 D1                       ;Select tool
N0030 G00 G90 X0 Y0 Z50
N0040 S1 [MC_TouchProbe Channel=1] ;Read measured value
N0050 #SYNC IN [CONV_VEL 30]       ;Select synchronisation
N0060 G0 X50 Y0                    ;Position on moving workpiece
N0070 #CS ON[0,0,0,0,0,45]
N0080 G01 X50 F5000
N0090 G01 Y50
N0100 G01 X-50
N0110 G01 Y-50
N0120 #CS OFF
N0130 #SYNC OUT [FEED_CONT=5]      ;Select desynchronisation
N0140 X[INDP_SYN G1 POS0 FEED5 INTERRUPTIBLE]  ;Position in MCS
N0150 G00 X0 Y0 Z50
N0160 M30
```

3.1 MC_TouchProbe command

The MC_TouchProbe command records an axis position when a trigger event occurs. The measurement channel and method (rising or falling edge of the trigger signal) are defined via the reference for the trigger signal source.

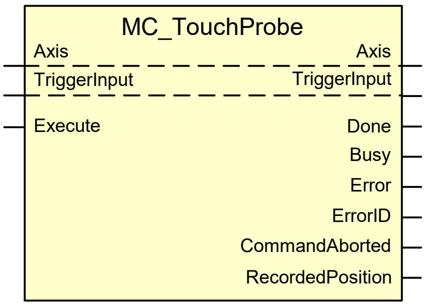
Optionally, the program can wait for the job to end (“WaitSyn” keyword) or can assign a Job ID (“Id” keyword) for later synchronisation.

```
Syntax of the NC command:
```

<*axis_name*>**[MC_TouchProbe Channel**=.. **[Mode**=..] [**Id**=..] [**WaitSyn**] { \ } **]**

Block diagram of the function block in PLCopen:

Input pin		Unit	
TriggerInput	Channel	----	
	Mode	---	If no mode is specified, probe signal evaluated acc. to P-AXIS-00518
		1	Probe signal on falling edge
		2	Probe signal on rising edge



MC_TouchProbe

```
S[MC_TouchProbe Channel=1 Mode=2]
```

4 Enabling

This section describes the steps required to enable an application using the conveyor tracking functionality.

You are advised to proceed step by step during the enabling process and to check the success of each step.

Due to the overall complexity, it is difficult to find the possible cause of errors if several or all of the steps are implemented at the same time.

4.1 Configuring the machine set-up

At the start of enabling, the machine must be set up in the channel depending on its axes or kinematics. A distinction can be made between two variants when setting up the machine:

The first variant includes machines that can co-move the processing unit TCP parallel to the conveying direction. Applications such as “Laser machining on moving sheet [▶ 23]” or “Pick & place [▶ 23]” can be implemented on this type of machine.

The second variant includes machines with a static processing unit that cannot be moved by the machine axes parallel to the conveying direction. In this case, the workpieces are only moved through the machine, see “Machining without co-moving unit [▶ 24]”.

4.1.1 First variant: co-moving processing unit

In the case of a machine with co-moving axes, it must be ensured that the axes have the necessary dynamics to synchronise with the linear conveyor and to machine the workpiece.

The maximum conveyor velocity must be determined for each application by testing. When a kinematic transformation is used, it must be ensured that the synchronisation movements are not planned at axis level but as TCP movements. This does not ensure that the configured axis dynamics are maintained. In this case, too, a suitable feed rate and axis configuration must be determined by testing.

In the case of a co-moving processing unit, a machine workspace must be configured on the linear conveyor; see “Configuring the workspace and monitoring the workspace limits [▶ 32]”.

4.1.2 Second variant: static processing unit

A virtual X axis must be created for a machine with a static processing unit.

This involves configuring a simulation axis with sufficient dynamics to cater for the conveyor dynamics. In order to use the virtual X axis, the parameter conveyor_sync.x_virtual (P-CHAN-00364 [▶ 52]) must be specified in the channel parameter list.

In the case of a static machine, the maximum conveyor velocity is heavily dependent on the dynamics of the axes involved in the machining operation and on the programmed contour. This is because the X axis cannot be controlled and therefore the machining axes can only follow the X axis.

If the position changes become too large due to the conveyor axis moving too fast, the CNC will output a dynamic error.

4.2 Configuring the linear conveyor

A linear conveyor is represented in the channel by a spindle axis. A distinction must be made here as to whether the linear conveyor is controlled by the CNC or by an external source.

- When the linear conveyor is controlled by the CNC, the axis must be configured to match the drive used.

- If the linear feeder is controlled externally, the spindle axis must be configured as a pure encoder axis using P-AXIS-00015 [▸ 48]

In both variants, the path resolution must be correctly parameterised using P-AXIS-00511 and P-AXIS-00512 so that the CNC can determine the subsequent workpiece position on the linear conveyor. The axis must then be parameterised as a linear conveyor axis using P-AXIS-00708 [▸ 52].

The spindle axis must still be identified in the channel as a conveyor axis together with its logical axis number using P-CHAN-00362 [▸ 52]. The parameter P-CHAN-00363 [▸ 52] is used to specify the conveying direction.

Checking the linear conveyor configuration

Finally, a check is made to ensure that the linear conveyor is correctly represented by the spindle axis. This is done by either starting the linear conveyor externally or using the PLCopen commands MC_MoveVelocity or MC_MoveAbsolute and acquiring the velocity and position display variables of the axis.

This data is available via the lr_state.current_rev_r and lr_state.current_position_acs_r control units.

After the spindle axis is verified, one option is to then configure the filter parameters for the spindle axis. To determine the correct filter values, they can be varied in an NC program using the #MACHINE DATA command and the display signals [▸ 58] of the unfiltered and filtered conveyor positions, velocities and accelerations can be acquired.

4.3 Configuring the measuring system

The measuring system is used by the CNC to locate the workpiece on the linear conveyor. The channel parameters P-CHAN-00368 [▸ 53] to P-CHAN-00373 [▸ 54] describe the location of the measuring system relative to the machine coordinate system.

In the application, a measured value corresponds to a position of the linear conveyor at a specific point in time. The CNC interprets this position as the position of the front edge of the workpiece at the measuring system position.

The parameter P-AXIS-00516 can be used to configure how the measured value for the spindle axis representing the linear conveyor is transferred to the CNC. Typically, three different measuring methods are recommended for this application:

1. The CNC controls the linear conveyor and a maximum of one measured value is always presented: direct measurement via the drive interface. In this variant, the PLC does not possess any measuring logic.
2. The linear conveyor is controlled externally and a maximum of one measured value is always presented: The measuring trigger is activated by the PLC (P-AXIS-00516 == PLC and lr_mc_control.probing_signal) and the measuring position is read by the drive interface.
3. The linear conveyor is controlled externally and measured values must be saved because new measured values occur and previous workpieces are not yet fully processed: The measuring trigger (P-AXIS-00516 == PLC and lr_mc_control.probing_signal) and the measuring position (lr_mc_control.probing_position) are specified by the PLC. In this case, the PLC must possess the logic to assign the various measured values and the appropriate NC program.

Checking the measuring system configuration

A measurement should be triggered to check whether the measuring system is correctly configured and the following variables are acquired:

- Measuring trigger: The variable lr_mc_control.probing_signal should be acquired when the PLC sends the trigger; the variable lr_state.probe_actuated_r should be acquired when the measurement is sent via the drive interface.
- Measured value: The variable lr_mc_control.probing_signal should be acquired when the PLC sends the trigger; the variable lr_state.probe_actuated_r should be acquired when the measurement is sent via the drive interface.

- Workpiece position: The variable bahn_state.conveyor_data.current_pos_master [▶ 58] indicates the current workpiece position on the linear conveyor. If a measuring system is installed that can also measure the actual workpiece position on the linear conveyor, this signal would also be useful.
- The workpiece position must be identical to the actual position after a measurement in order to verify that the measuring unit is configured correctly. This means that the workpiece position jumps to the configured X offset of the measuring unit (P-CHAN-00368 [▶ 53]) at the time of measurement. The position should then continue to move by the same position as the linear conveyor.

4.4 Configuring the workspace and monitoring the workspace limits

NOTICE
This function may not be used to implement safety-related functions.

While synchronisation with the conveyor system is active, the conveyor tracking functionality monitors the contour for intersections with the workspace limits. This requires the workspace to be configured for synchronisation. This workspace is only valid within the conveyor tracking function and is only monitored while synchronisation is active.

4.4.1 Configuring the workspace

The workspace for conveyor tracking is a cuboid. The cuboid is described in the machine coordinate system (MCS) using P-CHAN-00859 [▶ 56] to P-CHAN-00864 [▶ 57] as shown in the figure below.

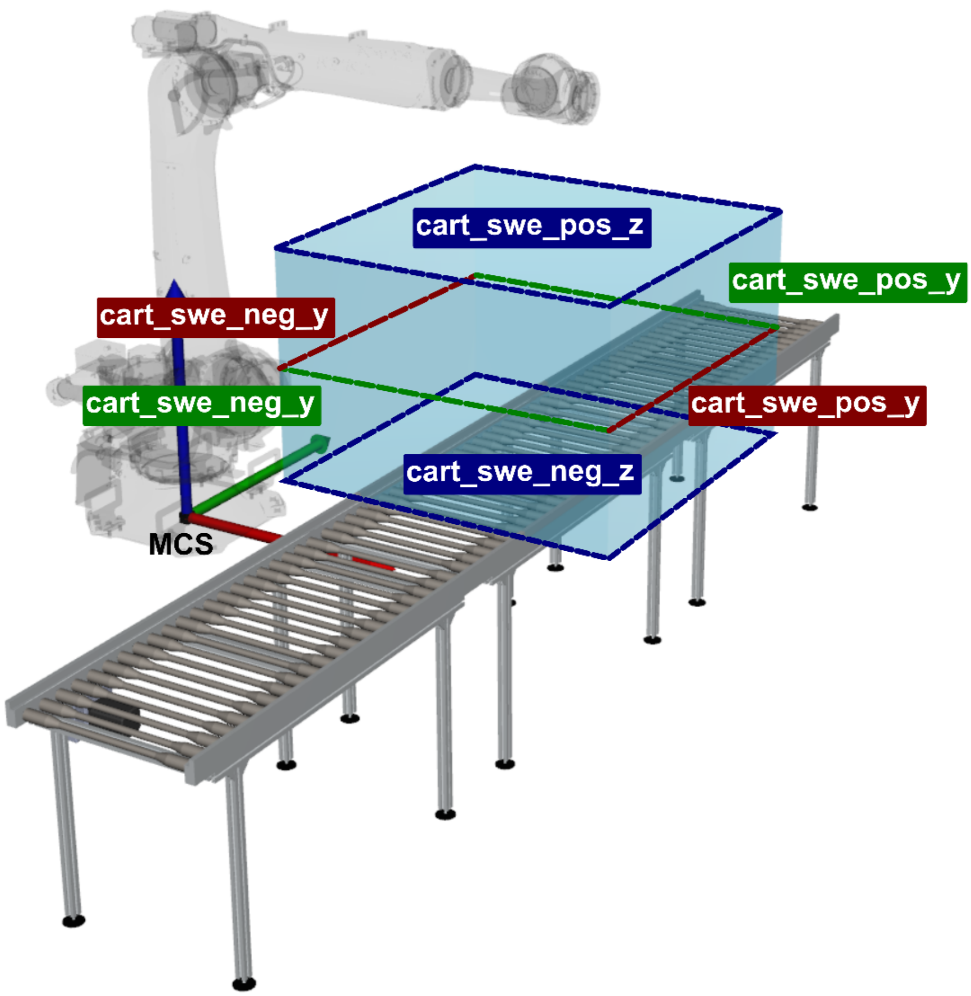


Fig. 10: Configuration and schematic diagram of a workspace monitoring system

The parameters describe the upper and lower workspace limits in the axes X, Y and Z of the MCS. The workspace cannot be rotated. This is why the workspace axes run parallel to the MCS axes.

If the parameters P-CHAN-00859 - P-CHAN-00864 are not configured, the configured software limit switches (P-AXIS-00177 [▸ 49] / P-AXIS-00178 [▸ 50]) of the first three axes are used to define the workspace.

This is why the workspace limits must always be configured when a kinematic transformation is used.

The function for monitoring workspace limits, see "Monitoring the workspace limit in the opposite conveying direction [▸ 16]". is only active while synchronisation with the conveyor system is active.

4.4.2 Configuring the workspace limit monitoring system

The parameters P-CHAN-00366 [▸ 53] (pos_limit) and P-CHAN-00374 [▸ 54] (hold_limit_vel_factor) must be set based on fluctuations in the conveyor velocity. An NC program can be used for this purpose. It runs parallel to the X axis and exceeds the workspace limit in the opposite conveying direction.

The two parameters must be set so that the workspace limits are not exceeded even if the conveying velocity fluctuates.
In practice, a setting of 0.95 has proven effective for parameter P-CHAN-00374 [▸ 54].

The optimised monitoring of the workspace can be configured using the parameter P-AXIS-00555 [▸ 50] (kenngr.conv_sync_optim) of the conveyor axis (axis index = 0). This requires a higher computing power. During enabling, make sure that the cycle time is not exceeded when this function is used.

4.5 Synchronising the workpiece on the linear conveyor

After the previous steps are completed, a workpiece can then be synchronised on the linear conveyor. An NC program describes in detail all the machining operations executed on a workpiece on the linear conveyor. The sequence for synchronising the workpiece is always identical:

- Workpiece is measured: Measuring unit scans the workpiece and stores the related position of the linear conveyor.
- Synchronising with the workpiece: The CNC moves the processing unit to the starting point on the workpiece. The processing unit velocity is then identical with the linear conveyor velocity.
- Machining the workpiece: The workpiece is machined on the linear conveyor by the CNC.
- Desynchronising with end position: After machining, the CNC moves the processing unit to the end position. At the end position, the processing unit has zero velocity.

If the workpiece is only measured via the drive interface, the CNC can synchronise itself with the workpiece independently and this only requires starting an NC program.

If the PLC has to trigger the measurement or even manage the measured values, the application consists of two concurrent processes. The two processes have two synchronisation times:

1. Before machining a workpiece, the CNC waits until a valid measurement is presented.
2. Before the next measurement, the PLC must wait until the current machining operation is completed.

To test the CNC synchronisation, it is recommended to program only a waiting period for the machining operation.

This can be implemented using the program example below:

Synchronising the workpiece and waiting time on the linear conveyor

```
N0030 G00 G90 X0 Y0 Z0
(- Start conveyor, only required if CNC controls the conveyor )
N030 S1[MC_MoveVelocity Velocity=300000 (500000=30m/min) Acceleration=2000 Deceleration=2000
Jerk=750000 Direction=1]
N0040 S1 [MC_TouchProbe Channel=1] ;Read measured value
N0050 #SYNC IN [CONV_VEL 30]       ;Select synchronisation
N0060 G0 X0 Y0                    ;Position on moving workpiece
N0070 G04 10                       ;Waiting time instead of machining
N0130 #SYNC OUT [FEED_CONT=5]      ;Select desynchronisation
N0140 X[INDP_SYN G1 POS0 FEED5 INTERRUPTIBLE]  ;Position in the MCS
N0160 M30
```

After the machine is successfully synchronised, the delay time compensation can be optimally set using the parameter P-AXIS-00626 [▶ 51] and the display variable synchro_lag [▶ 59], see Filtering the actual values of the conveyor axis [▶ 19].

Below is an example of the curve with an inconstant feed rate (approximate sinusoidal curve) and a delay time in the control section:

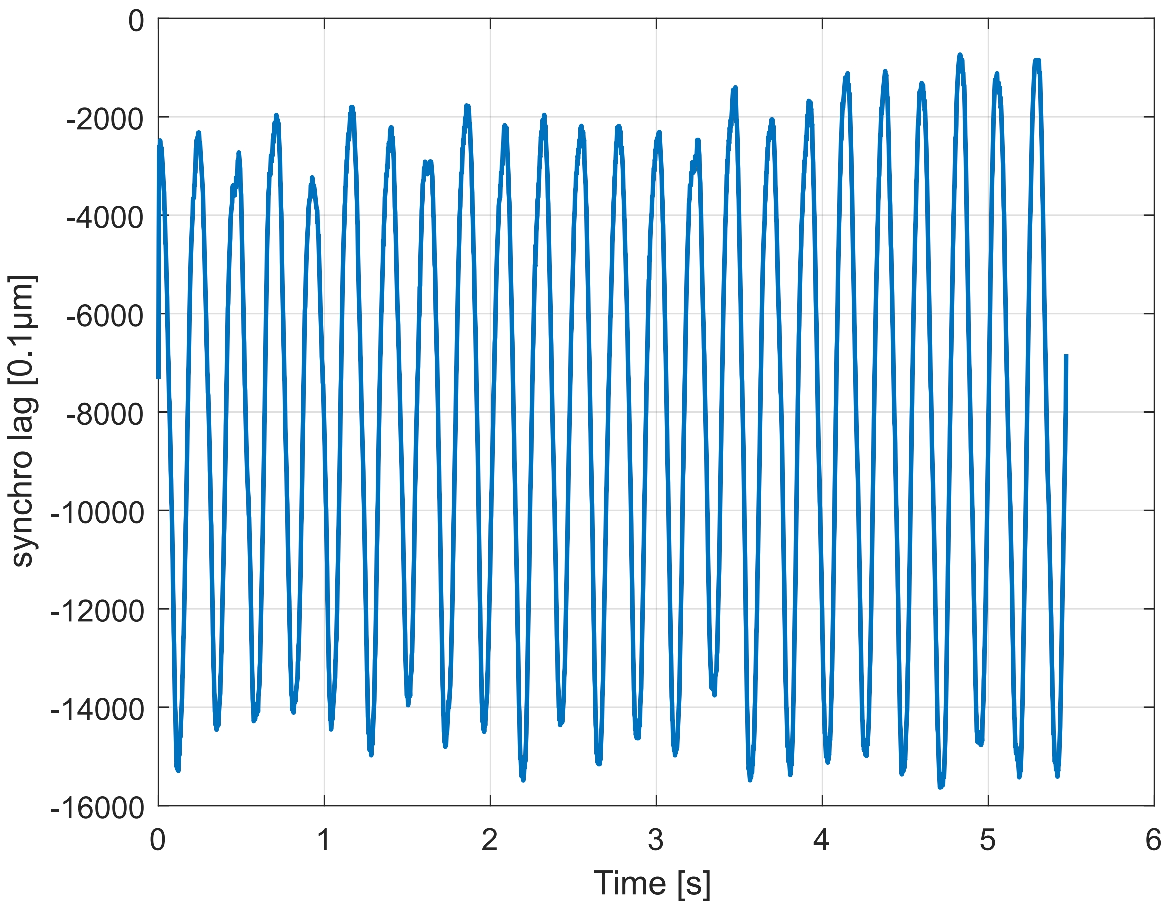


Fig. 11: Time curve with inconstant conveying velocity

Since the position feedforward control for delay time compensation assumes that the velocity remains constant, the assumed sinusoidal signal also results in a sinusoidal position deviation.

A static error can also be seen in the signal since the sine does not oscillate about 0. This static error can be improved to 0 by correctly setting the delay time using P-AXIS-00626 [▶ 51].

The synchro_lag variable is only correct if programmed in a specific way. A distinction must be made between a machine with a co-moving processing unit and a machine with a static processing unit.

Two program examples are presented below:

Static processing unit

With a static processing unit, a contour must be programmed where the target points in the second axis (Y) coincide with the target points of the virtual axis. The example shows the correct calculation of the synchro_lag variable in block N080.

```
N030 #CONTOUR MODE [DEV]
N040 G17 G161
; Request measurement from conveyor
N050 S1 [MC_TouchProbe Channel=1]

; Synchronise with conveyor
N060 #SYNC IN [CONVEYOR=S1, CONV_VEL=200 VEL_RESOLUTION=mm/s FEED_CONT=0]
N070 G0 G90 X50 Y0 F100
N070 G01 G90 X25 Y0 F100 G61
N070 G01 G90 X0 Y0 F100 G61
; Linear motion moves in Y in the same way as X moves
N080 G01 X-550 Y-550 G61
N085 G01 X-600 Y-550 G61
; End sequence
N090 #SYNC OUT [FEED_CONT=5]
N100 X[INDP_SYN G0 G90 POS0 INTERRUPTIBLE] Y[INDP_SYN G0 G90 POS0 INTERRUPTIBLE]
```

Co-moving processing unit

With a co-moving processing unit, the unit must wait for 0 after synchronisation with the X axis. The example shows the correct calculation of the synchro_lag variable in block N090.

```
; Waiting position
N040 G0 G90 X0 Y400 Z0

; Request spindle measured value
N050 S1 [MC_TouchProbe Channel=1]

; Synchronise with the belt
N060 #SYNC IN [CONVEYOR=S1, CONV_VEL=18000 FEED_CONT=0]
N080 G0 G90 X0 Y450 Z160
N090 G04 2
N100 #SYNC OUT [FEED_CONT=5]

N110 X[INDP_SYN G0 G90 POS0 INTERRUPTIBLE] Y[INDP_SYN G0 G90 POS400 INTERRUPTIBLE] Z[INDP_SYN G0 G90
POS0 INTERRUPTIBLE]
```

4.6 Machining the moving workpiece

After completing the static synchronisation with the workpiece, the actual machining process can be executed. Apart from the requirements regarding measured values (see Configuring the measuring system [▶ 30]), the CNC makes a distinction between two different variants:

Machining operations on workpieces are always identical

When machining operations are identical, an endless loop can be programmed in an NC program. The endless loop always waits for the measured value of the next workpiece before starting the machining operation. At the end of the machining operation, the loop is restarted and the system waits for the measured value of the next workpiece.

Workpiece machining is a singular operation and dependent on the measurement

If a machining operation depends on the measurement of the current workpiece, an NC program template can be created for the machining operation. The differing sections in the NC program can then be described by V.E. variables which the PLC then correctly assigns after measuring the current workpiece. The template program is then started for each individual workpiece.

4.7 Optimising the application

Machining accuracy on a moving workpiece can be increased by carrying out the following measures:

1. A highly constant conveying velocity results in optimised feedforward control. By contrast, strongly fluctuating feed rates lead to greater position errors since feedforward control assumes a constant velocity.
2. The smaller the system delay time, the smaller the influence of feed rate fluctuations on position accuracy, see also CNC scheduling. Delay time can be defined using the display variable synchro_lag [▶ 59].

4.8 Application example: Laser machining on moving sheet

The application example for laser machining involves a Cartesian machine set-up with a parallel X axis to the conveying direction. Therefore, three machine axes must be configured in the channel. In the case of the X axis, please ensure that the dynamics are set appropriately so that the axis can synchronise with the linear conveyor. The machine workspace and the behaviour on reaching the limit must be configured in the channel and in the X axis.

In this example, the linear conveyor represented by the spindle axis is controlled by the CNC. When the spindle axis speed is positive, the linear conveyor moves in the positive X direction of the machine.

The measuring unit in this example is located exactly at the origin of the machine coordinate system.

4.8.1 Parameterisation example

The application example below lists the key parameters in the axis and channel lists:

Axis 1

```
kopf.log_achs_name                    X
kopf.achs_nr                          1
kenngr.swe_pos                        40000000 ( P-AXIS-00178
kenngr.swe_neg                        -2000000 ( P-AXIS-00177
kenngr.conv_sync_optim                1        ( P-AXIS-00555
```

Axis 2

```
kopf.log_achs_name                    Y
kopf.achs_nr                          2
kenngr.swe_pos                        25000000 ( P-AXIS-00178
kenngr.swe_neg                        -2000000 ( P-AXIS-00177
```

Axis 3

```
kopf.log_achs_name                    Z
kopf.achs_nr                          3
kenngr.swe_pos                        3000000  ( P-AXIS-00178
kenngr.swe_neg                        -1000000 ( P-AXIS-00177
```

Spindle

```
kopf.log_achs_name                    S
kenngr.achs_typ                       4 ( type == 4 spindle
kenngr.measure.signal                 PLC_FIRST_EVENT
conv_sync.is_master                   1
conv_sync.enable_filter               1
conv_sync.type_pos_filter             1
conv_sync.order_pos_filter            4
conv_sync.order_v_filter              10
conv_sync.order_post_v_filter         6
conv_sync.order_v_filter_dyn          10
conv_sync.delay_time                  17455
```

The required entries in the channel parameter list are as follows:

```
configuration.interpolator.function   FCT_DLM    ( P-CHAN-00650
conveyor_sync.log_number_master       3 #spindle ( P-CHAN-00362
conveyor_sync.move_direction          0          ( P-CHAN-00363
conveyor_sync.sync_in_tolerance       100        ( P-CHAN-00365
conveyor_sync.hold_limit_vel_factor   950        ( P-CHAN-00366
conveyor_sync.hold_limit_tolerance    5.000e+004 ( P-CHAN-00367
conveyor_sync.cart_t0_shift_x         0          ( P-CHAN-00368
conveyor_sync.cart_t0_shift_y         0          ( P-CHAN-00369
conveyor_sync.cart_t0_shift_z         0          ( P-CHAN-00370
conveyor_sync.cart_t0_rot_a           0          ( P-CHAN-00371
conveyor_sync.cart_t0_rot_b           0          ( P-CHAN-00372
conveyor_sync.cart_t0_rot_c           0          ( P-CHAN-00373
conveyor_sync.pos_limit               -1900000   ( P-CHAN-00374
```

4.8.2 Programming example

Laser machining on moving sheet

Linear conveyor CNC controlled

```
% laser

; Start conveyor
N010 S1[MC_MoveVelocity Velocity=20000 Acceleration=2000 Deceleration=2000 Jerk=750000 Direction=1]

; Waiting position
N020 G0 G90 X0 Y400 Z0

; Request spindle measured value
N030 S1 [MC_TouchProbe Channel=1]

; Synchronise
N040 #SYNC IN [CONVEYOR=S1, CONV_VEL=6000 FEED_CONT=0]

N050 G0 G90 X0 Y450 Z160
$FOR P2 = 0, 24, 1
; Switch on laser
N060 M50
; Plunge
N70 G01 G91 G261 X0 Y0 Z20 F10000
N80 G01 G91 X0 Y-50 Z0
N90 G01 G91 X-50 Y0 Z0
N100 G01 G91 X0 Y50 Z0
N110 G01 G91 X50 Y0 Z0
; Switch off laser
N120 M51
N130 G00 G91 G260 X0 Y100 Z-20
$ENDFOR

; Cut out
N140 G0 G90 X-100 Y350
N150 M50
N160 G01 G91 X150 F100000
N170 G01 G91 Y2600
N180 G01 G91 X-150
N190 G01 G91 Y-2600
N200 M51

N210 #SYNC OUT [FEED_CONT=5]

N220 X[INDP_SYN G0 G90 POS0 INTERRUPTIBLE] Y[INDP_SYN G0 G90 POS400 INTERRUPTIBLE] Z[INDP_SYN G0 G90
POS0 INTER-RUPTIBLE]

; Program end
N230 M30
```

4.9 Application example: Pick & place

The example here shows the application taken from "Pick & place [▶ 23]".

4.9.1 Parameterisation example

This section lists the key parameters in the axis and channel lists for the application example.

The required parameters of the axes concerned are listed below: The only thing you should make sure about is that the axes X, Y, Z and C are identical to the robot axes. In this example, the linear conveyor is controlled and the current workpiece is measured directly via the drive interface. This means that no PLC code is required in this example.

Axis 1

```
kopf.log_achs_name                    X
kopf.achs_nr                          1
```

Axis 2

```
kopf.log_achs_name                    Y
kopf.achs_nr                          2
```

Axis 3

```
kopf.log_achs_name                    Z
kopf.achs_nr                          3
```

Axis 4

```
kopf.log_achs_name                    C
kopf.achs_nr                          4
```

Spindle

```
kopf.log_achs_name                    S1
kenngr.achs_typ                       4 ( type == 4 spindle
kenngr.messachse                      1
kenngr.measure.signal                 DRIVE_TYPE_DEFAULT
conv_sync.is_master                   1
conv_sync.enable_filter               1
conv_sync.type_pos_filter             1
conv_sync.order_pos_filter            4
conv_sync.order_v_filter              10
conv_sync.order_post_v_filter         6
conv_sync.order_v_filter_dyn          10
conv_sync.delay_time                  17455
```

The required entries in the channel parameter list are as follows:

```
configuration.interpolator.function   FCT_DLM ( P-CHAN-00650
spdl_anzahl                           1
#
main_spindle_ax_nr                    5
main_spindle_name                     S1
#
trafo[0].id                           37
trafo[0].param[0]                     0
trafo[0].param[1]                     0
trafo[0].param[2]                     0
trafo[0].param[4]                     2000000
trafo[0].param[3]                     480000
trafo[0].param[5]                     10500000
trafo[0].param[6]                     4655440
trafo[0].param[7]                     0
trafo[0].param[8]                     0
trafo[0].param[9]                     0
trafo[0].param[10]                    -1800000
trafo[0].param[11]                    1800000
trafo[0].param[12]                    0
#
conveyor_sync.log_number_master       5
```

```
conveyor_sync.move_direction         0
conveyor_sync.x_virtual              0
conveyor_sync.sync_in_tolerance      1.0e+02
conveyor_sync.hold_limit_vel_factor  950
conveyor_sync.hold_limit_tolerance   5.0e+04
conveyor_sync.cart_t0_shift_x        0
conveyor_sync.cart_t0_shift_y        0
conveyor_sync.cart_t0_shift_z        0
conveyor_sync.cart_t0_rot_a          0.0
conveyor_sync.cart_t0_rot_b          0.0
conveyor_sync.cart_t0_rot_c          0.0
conveyor_sync.pos_limit              0
conveyor_sync.cart_swe_pos_x         5000000
conveyor_sync.cart_swe_pos_y         5000000
conveyor_sync.cart_swe_pos_z         5000000
conveyor_sync.cart_swe_neg_x         -5000000
conveyor_sync.cart_swe_neg_y         -5000000
conveyor_sync.cart_swe_neg_z         -5000000
```

4.9.2 Programming example

Pick & place

The example below contains a program for the case that all workpieces are picked at the same position on the workpiece and placed at the same location in the machine coordinate system. An endless loop is programmed for this reason and the CNC waits again for the next workpiece immediately after placing a workpiece. The linear conveyor is controlled by the CNC (N030) and it is recognisable that a delta robot is used because kinematic 37 is selected at the start.

```
%pick_and_place

N010 #KIN ID[37]
N020 G0 G90 X34.9485 Y-18.0290 Z34.0535
N020 #TRAFO ON

(- Start conveyor)
N030 S1[MC_MoveVelocity Velocity=300000 (500000=30m/min) Acceleration=2000 Deceleration=2000
Jerk=750000 Direction=1]

N040 $WHILE [TRUE]

; Waiting position
N050 G01 X-100 Y0 Z-700 F350000

(- Request spindle measured value-)
N060 S1 [MC_TouchProbe Channel=1]

(- Synchronise with the current workpiece -)
N070 #SYNC IN [CONVEYOR=S1, CONV_VEL=18000 FEED_CONT=0]

(- Grip the workpiece -)
N075 G01 X-50 Y0 Z-700 F350000
N080 G01 X-50 Y0 Z-800 F350000
N090 G01 X-50 Y0 Z-750 F350000

(- Desynchronise and place the workpiece -)
N100 #SYNC OUT [FEED_CONT=5]

(- Move to place location -)
N110 X[INDP_SYN G1 G90 POS300 FEED350000 INTERRUPTIBLE] Y[INDP_SYN G1 G90 POS500 FEED350000
INTERRUPTIBLE] Z[INDP_SYN G1 G90 POS-750 FEED350000 INTERRUPTIBLE]

(- Place -)
N120 G01 Z-790 F350000
N130 G01 Z-800 F20000
N140 G01 Z-790 F350000

N150 $ENDWHILE

(-Program ended-)
N160 M30
```

4.10 Application example: Machining with static processing unit

The example here shows the application taken from “Machining without co-moving unit [▶ 24]”. As opposed to the application examples above, the machine set-up contains a static processing unit. There is no parallel axis in the conveying direction to synchronise the processing unit with the linear conveyor movement. In this machine, the X axis must be configured as a simulation axis and also as a virtual X axis using P-CHAN-00364 [▶ 52].

Since the saw blade must always be aligned tangentially during machining, the "Automatic axis tracking" function is also used in this application. In this case, the tracking axis may not be located in a main axis index (axis 1-3). This means that an additional simulation axis must be created on axis index 2.

When programming, make sure there is no inconsistency in the position and orientation of the saw blade at the start of machining. This is ensured in the example by pre-positioning the Y and C axes or by a tangential approach movement. An additional contouring mode is used to ensure that the contour is C2-continuous.

4.10.1 Parameterisation example

The required parameters of the axes concerned are listed below:

Axis 1

```
kopf.log_achs_name                   X
kopf.achs_nr                         1
kenngr.antr_typ                      4          ( P-AXIS-00020
```

Axis 2

```
kopf.log_achs_name                   Y
kopf.achs_nr                         2
```

Axis 3

```
kopf.log_achs_name                   Z
kopf.achs_nr                         3
kenngr.swe_pos                       550000     ( P-AXIS-00178
kenngr.swe_neg                       -550000    ( P-AXIS-00177
kenngr.antr_typ                      4          ( P-AXIS-00020
```

Axis 4

```
kopf.log_achs_name C
kopf.achs_nr 4
```

Spindle

```
kopf.log_achs_name                   S
kenngr.achs_typ                      4          ( type == 4 spindle
kenngr.measure.signal                PLC_FIRST_EVENT
conv_sync.is_master                  1
conv_sync.enable_filter              1
conv_sync.type_pos_filter            1
conv_sync.order_pos_filter           4
conv_sync.order_v_filter             10
conv_sync.order_post_v_filter        6
conv_sync.order_v_filter_dyn         10
conv_sync.delay_time                 17455
```

The required entries in the channel parameter list are as follows:

```
configuration.interpolator.function  FCT_DLM    ( P-CHAN-00650
conveyor_sync.log_number_master      5          ( P-CHAN-00362 - spindle
conveyor_sync.move_direction         0          ( P-CHAN-00363
conveyor_sync.x_virtual              1          ( P-CHAN-00364
conveyor_sync.sync_in_tolerance      100        ( P-CHAN-00365
conveyor_sync.hold_limit_vel_factor  950        ( P-CHAN-00366
conveyor_sync.hold_limit_tolerance   5.000e+004 ( P-CHAN-00367
conveyor_sync.cart_t0_shift_x        -10000000  ( P-CHAN-00368
conveyor_sync.cart_t0_shift_y        0          ( P-CHAN-00369
conveyor_sync.cart_t0_shift_z        0          ( P-CHAN-00370
conveyor_sync.cart_t0_rot_a          0          ( P-CHAN-00371
```

```
conveyor_sync.cart_t0_rot_b          0          ( P-CHAN-00372
conveyor_sync.cart_t0_rot_c          0          ( P-CHAN-00373
conveyor_sync.pos_limit              -1900000   ( P-CHAN-00374
log_number_tracking_axis             4          ( P-CHAN-0095
```

4.10.2 Programming example

Machining with static processing unit

The programming example below is conceived for one workpiece only. If all the workpieces are to be machined identically, an endless loop could be inserted before block N040 and after block N190. In this example, the linear conveyor started in line N020 is controlled by the CNC. This line is omitted if the linear conveyor is controlled externally. In this program, blocks N130 - N160 represent the actual contour on the workpiece. If the contour depends on the current measurement, the position used there could be replaced by V.E. variables assigned by the PLC:

```
%
(- Home the axes )
N010 G74 Y1 C1

(- Start conveyor )
N020 S1[MC_MoveVelocity Velocity=100000 Acceleration=2000 Deceleration=2000 Jerk=750000 Direction=1]

N030 G0 G90 Y0 C0

(- Request spindle measured value)
N040 S1 [MC_TouchProbe Channel=1]

(- Synchronise with conveyor )
N050 #SYNC IN [CONVEYOR=S1, CONV_VEL=18000 FEED_CONT=0]
N060 G0 G90 X0 Y0 C0

(- Activate automatic tracking )
N070 #CAXTRACK ON [AX=C, ANGLIMIT 3, OFFSET 180, ROTMODE=1]

(- Feed rate is ignored and corresponds to the conveying velocity )
N080 F100
N090 #SPLINE TYPE BSPLINE
N100 #SPLINE ON
(- Start of tangential movement )
N110 G1 G90 X-0 Y0
N120 G1 G90 X-50 Y0
N130 G1 G90 X-200 Y-30
N140 G1 G90 X-350 Y30
N150 G1 G90 X-550 Y0
(- End of tangential movement )
N160 G1 G90 X-600 Y0
N170 #SPLINE OFF

N180 #SYNC OUT [FEED_CONT=5]
N190 X[INDP_SYN G0 G90 POS0 INTERRUPTIBLE] Y[INDP_SYN G0 G90 POS0 INTERRUPTIBLE]

N200 #CAXTRACK OFF

N210 M30
```

5 Parameter

If the conveyor is not controlled as an axis by the CNC, a spindle axis must be configured with the operation mode ACHSMODE_COUNTER [P-AXIS-00015] so that the CNC can read out the measured values for the moving workpieces.

5.1 Overview

Axis parameter

ID	Parameter	Description
P-AXIS-00015	kenngr.achs_mode	Operation mode of an axis
P-AXIS-00177	kenngr.swe_neg	Negative software limit switch
P-AXIS-00178	kenngr.swe_pos	Positive software limit switch
P-AXIS-00555	kenngr.conv_sync_optim	Method to monitor the workspace of conveyor systems
P-AXIS-00620	conv_sync.enable_filter	Activate actual position value filter for the conveyor axis
P-AXIS-00623	conv_sync.order_v_filter	Order of the velocity actual value filter of the conveyor axis
P-AXIS-00624	conv_sync.order_post_v_filter	Order of the downstream velocity actual value filter in the case of a conveyor axis feedhold
P-AXIS-00625	conv_sync.order_v_filter_dyn	Order of the velocity actual value filter in the case of a conveyor axis feedhold
P-AXIS-00626	conv_sync.delay_time	Delay time between the latch moment of the actual values and generating the command value
P-AXIS-00708	conv_sync.is_master	Definition of the conveyor axis

Channel parameters

ID	Parameter	Description
P-CHAN-00362	conveyor_sync.log_number_master	Logical axis number of the linear conveyor
P-CHAN-00363	conveyor_sync.move_direction	Conveying direction of the linear conveyor
P-CHAN-00365	conveyor_sync.sync_in_tolerance	Tolerance window for synchronisation
P-CHAN-00366	conveyor_sync.hold_limit_vel_factor	Factor for reducing the velocity when limiting movement to the end position
P-CHAN-00368	conveyor_sync.cart_t0_shift_x	X offset of the Cartesian basic coordinate system
P-CHAN-00369	conveyor_sync.cart_t0_shift_y	Y offset of the Cartesian basic coordinate system
P-CHAN-00370	conveyor_sync.cart_t0_shift_z	Z offset of the Cartesian basic coordinate system
P-CHAN-00371	conveyor_sync.cart_t0_rot_a	A rotation of the Cartesian basic coordinate system
P-CHAN-00372	conveyor_sync.cart_t0_rot_b	B rotation of the Cartesian basic coordinate system
P-CHAN-00373	conveyor_sync.cart_t0_rot_b	C rotation of the Cartesian basic coordinate system
P-CHAN-00374	conveyor_sync.pos_limit	End position of the X axis in the co-moving coordinate system
P-CHAN-00650	configuration.interpolator.function	Activate the conveyor tracking function
P-CHAN-00859	conveyor_sync.cart_swe_pos_x	X position of the upper end position of the workspace
P-CHAN-00860	conveyor_sync.cart_swe_pos_y	Y position of the upper end position of the workspace

P-CHAN-00861	conveyor_sync.cart_swe_pos_z	Z position of the upper end position of the workspace
P-CHAN-00862	conveyor_sync.cart_swe_neg_x	X position of the lower end position of the workspace
P-CHAN-00863	conveyor_sync.cart_swe_neg_y	Y position of the lower end position of the workspace
P-CHAN-00864	conveyor_sync.cart_swe_neg_z	Z position of the lower end position of the workspace

5.2 Description

5.2.1 Axis parameter

P-AXIS-00015	Axis mode	
Description	Axes can be traversed in different operating modes.	
Parameter	kenngr.achs_mode	
Data type	UNS32	
Data range	0x00000001 - 0x10000000	
Axis types	T, R, S	
Dimension	T: ----	R,S: ----
Default value	0x00000001	
Drive types	----	
Remarks		

The following operation modes can be parameterised(1):

Value	Meaning		Axis type	Interpolator type	
	Description	ACHSMODE_		Path	Spindle
0x00000001	The axis is operated as a linear axis; no modulo calculation is executed in the channel. For example, rotary axis with restricted motion range; must be set as default for linear axes.	..LINEAR(1)	T, R	X	
0x00000004	A modulo calculation always takes place after the destination position has been reached. Regardless of the operation mode selected for rotary axes, a modulo calculation is always executed in the position controller. In this way, modulo circle compensation can be executed if required.	..MODULO(1)	R	X	X
0x00000040	Axis is used as a face turning axis (turning functions).	..PLANDREHEN	T	X	
0x00000080	Axis is used as a longitudinal turning axis (turning functions).	..LAENGSDREHEN	T	X	
0x00000100	For a spindle an automatic homing before positioning the spindle can be pre-vented. This is only relevant if the axis is not referenced. The function is drive-dependent.	..KEINE_AUTO_RPF	R		X
0x00000200	Axis for kinematic 'C axis' transformation.	..CAX	R	X	X
0x00000400	Modulo calculation for linear axis. (Example: conveyor belt with drive motor where the position on the belt is programmed in mm).	..MODULO_LINEAR	R	X	
0x00000800	Axis is released for mechanical blocking by the PLC. This axis mode is not available for TwinCAT systems.	..CLAMPABLE	T, R	X	
0x00001000	Axis carries a rotary workpiece table.	..ROT_TABLE	T, R	X	
0x00008000	Monitoring of axis collision.	..COLL_CHECK	T	X	

Value	Meaning		Axis type	Interpolator type	
0x00010000	Master axis of gantry coupling.	..GANTRY_MASTER	T, R	X	
0x00020000	Slave axis of gantry coupling.	..GANTRY_SLAVE	T, R	X	
0x00040000	Identifier for PLC spindle with axis interface	..SPINDLE_EXT_CTRL	R		X
0x00080000	Input axis for additional external position command values (e.g. distance control).	..EXT_CTRL_INPUT	T	X	
0x00100000	Pure encoder axis, only to display actual values (e.g. conveyor belt).	..COUNTER	T, R	X	X
0x00200000	Lead axis in combination with single feed axis and G194 (contouring with DIST_MASTER)	..LEAD_AXIS	T, R	X	
0x00400000	The resolution (wegaufz/wegaufn) of this axis can be changed.	..ALLOW_RESOLUTION_CHANGE(2)	T, R	X	X
0x00800000	Path-dependent dynamic weighting for this axis is possible.	..DYNAMIC_WEIGHTING	T, R	X	
0x02000000	Path axis for tool centre point path	..PATH_LENGTH_TCP	R	X	
0x04000000	Path axis for contour path	..PATH_LENGTH_CONTOUR	R	X	
0x08000000	Virtual lead axis for path interpolation	..VIRT_LEAD_AXIS	R	X	
0x10000000	Axis carries the pressure roller for edge bending.	..LAH_OFFSET_AXIS	R	X	

(1) One of the following two axis modes **must** always be defined:

- ACHSMODE_LINEAR or
- ACHSMODE_MODULO

All other bits of the parameter *achs_mode* are additional specifications. For example, the ACHSMODE_MODULO_LINEAR bit is only useful if combined with the ACHSMODE_MODULO bit.

(2) Changing certain axis parameters while the controller is running may be critical, e.g. path resolution. Therefore, this parameter can be enabled by setting the bit ALLOW_RESOLUTION_CHANGE in axis mode. Otherwise, these parameters (P-AXIS-00234, P-AXIS-00233) can no longer be changed after controller start-up.

When the bit ALLOW_RESOLUTION_CHANGE is set, a check is first made whether the axis is interpolated, also when other critical parameters change. If the axis is currently in motion, this parameter update is rejected.

P-AXIS-00177	**Negative software limit switch**	
Description	The parameter defines the possible traverse range in the negative direction (negative software limit switch position). The programmed command positions are always checked on 'kenngr.swe_neg', the actual positions on 'kenngr.swe_neg - kenngr.swe_toleranz'.	
Parameter	kenngr.swe_neg	
Data type	SGN32	
Data range	MIN(SGN32) < swe_neg < P-AXIS-00178	
Axis types	T, R	
Dimension	T: 0.1µm	R: 0.0001 °
Default value	-100000000	
drive types.	----	
Remarks	The value of the parameter is adopted on reset, mode change and axis replacement from the axis record.	

P-AXIS-00178	**Positive software limit switch**	
Description	The parameter defines the possible traverse range in the positive direction (positive software limit switch position). The programmed command positions are always checked on 'kenngr.swe_pos', the actual positions on 'kenngr.swe_pos + kenngr.swe_toleranz'.	
Parameter	kenngr.swe_pos	
Data type	SGN32	
Data range	P-AXIS-00177 < swe_pos < MAX(SGN32)	
Axis types	T, R	
Dimension	T: 0.1µm	R,S: 0.0001 °
Default value	100000000	
drive types.	----	
Remarks	The value of the parameter is adopted on reset, mode change and axis replacement from the axis record.	

P-AXIS-00555	**Select method to monitor the workspace when using the conveyor tracking functionality.**	
Description	This parameter influences the monitoring of the workspace when workpieces are machined on a linear conveyor. With machines that have the conveyor tracking functionality, the tool should remain as close as possible to the *end position* defined in P-CHAN-00374 in order to utilise the workspace to the full extent. In the event of a movement in the opposite conveying direction, the controller brakes the tool accordingly to prevent the position from being exceeded. When contour elements are fed through in the opposite conveying direction at a higher feed rate than the conveyor belt velocity, a check must be made whether the workspace is exited in the opposite conveyor belt direction. This requires a more complex check of the workspace and can be activated by this parameter.	
Parameter	kenngr.conv_sync_optim	
Data type	BOOLEAN	
Data range	0: Simplified method with low computing time requirement. It is only suitable when the conveyor velocity is significantly higher than the programmed feed rate. 1: The programmed contour is checked for collision. In this method, the tool travels to the required *end position* if possible.	
Axis types	T	
Dimension	T: ----	
Default value	0	
Drive types	----	
Remarks	Available as of CNC Build V4.20.0	

P-AXIS-00620	**Enable position actual value filter for the conveyor axis**	
Description	This parameter enables and disables the filter for the position actual values of the conveyor axis.	
Parameter	conv_sync.enable_filter	
Data type	BOOLEAN	
Data range	0: Position actual value filter off 1: Position actual value filter on	
Axis types	T, R, S	
Dimension	T: ----	R, S: ----
Default value	0	
Drive types	----	
Remarks	Available as of CNC Build V4.20.0	

P-AXIS-00623	**Order of the velocity actual value filter of the conveyor axis**	
Description	This parameter defines the order (number of filtered values) of the velocity actual value filter of the conveyor axis. The filter is off in the case of order 0. An FIR low-pass filter is used automatically.	
Parameter	conv_sync.order_v_filter	
Data type	UNS32	
Data range	0 ≤ order_v_filter ≤ 1000	
Axis types	T, R, S	
Dimension	T: ----	R, S: ----
Default value	20	
Drive types	----	
Remarks	Available as of CNC Build V4.20.0	

P-AXIS-00624	**Order of the downstream velocity actual value filter in the case of feedhold on the conveyor axis**	
Description	This parameter defines the order (number of filtered values) of the downstream velocity actual value filter of a feedhold is defined on the conveyor axis. The filter is off in the case of order 0. An average value filter is used automatically.	
Parameter	conv_sync.order_post_v_filter	
Data type	UNS32	
Data range	0 ≤ order_post_v_filter ≤ 1000	
Axis types	T, R, S	
Dimension	T: ----	R, S: ----
Default value	6	
Drive types	----	
Remarks	Available as of CNC Build V4.20.0	

P-AXIS-00625	**Order of the velocity actual value filter in the case of feedhold on the conveyor axis**	
Description	This parameter defines the order (number of filtered values) of the velocity actual value filter if a feedhold is defined on the conveyor axis. The filter is off in the case of order 0. An FIR low-pass filter is used automatically.	
Parameter	conv_sync.order_v_filter_dyn	
Data type	UNS32	
Data range	0 ≤ order_v_filter_dyn ≤ 1000	
Axis types	T, R, S	
Dimension	T: ----	R, S: ----
Default value	10	
Drive types	----	
Remarks	Available as of CNC Build V4.20.0	

P-AXIS-00626	**Delay time between the latch moment of the actual values and generating the command value**	
Description	This parameter defines the delay time between the read-in actual value of the conveyor axis and the output command value of the machine axis. This delay time is used for feedforward control of the conveyor axis velocity.	
Parameter	conv_sync.delay_time	
Data type	UNS32	
Data range	0 ≤ delay_time ≤ MAX(UNS32)	
Axis types	T, R, S	
Dimension	T: [μs]	R, S: [μs]
Default value	0	

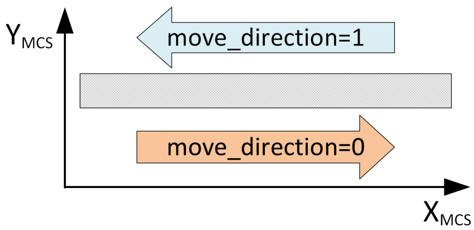
Drive types	----
Remarks	Available as of CNC Build V4.20.0

P-AXIS-00708	**Definition of the conveyor axis**	
Description	This parameter defines the axis representing the linear conveyor.	
Parameter	conv_sync.is_master	
Data type	BOOLEAN	
Data range	0: Axis is not a conveyor axis 1: Axis is the conveyor axis	
Axis types	T, R, S	
Dimension	T: ----	R, S: ----
Default value	0	
Drive types	----	
Remarks	Available as of CNC Build V4.20.0	

5.2.2 Channel parameters

P-CHAN-00362	**Logical axis number of the linear conveyor**
Description	This parameter defines the represented axis of the linear conveyor when the conveyor tracking functionality is used.
Parameter	conveyor_sync.log_number_master
Data type	UNS16
Data range	1 ≤ conveyor_sync.log_number_master ≤ MAX(UNS16)
Dimension	----
Default value	0
Remarks	Available as of CNC Build V4.20.0

P-CHAN-00363	Conveying direction of the linear conveyor
Description	This parameter defines the conveying direction of the linear conveyor.



Parameter	conveyor_sync.move_direction
Data type	UNS16
Data range	0: The linear conveyor moves at positive velocity in the positive X direction 1: The linear conveyor moves at negative velocity in the positive X direction
Dimension	----
Default value	0
Remarks	Available as of CNC Build V4.20.0

P-CHAN-00364	**Virtual X axis when the conveyor tracking functionality is used**
Description	This parameter must be set to 0 if a machine is to be synchronised with a linear conveyor (condition: the machine can move in the conveying direction). This parameter must be set to 1 for a machine that cannot move in the conveying direction.
Parameter	conveyor_sync.x_virtual

Data type	BOOLEAN
Data range	0/1
Dimension	----
Default value	0
Remarks	Available as of CNC Build V4.20.0

P-CHAN-00365	**Tolerance window for synchronisation with a linear conveyor**
Description	This parameter defines the tolerance window for synchronisation with a linear conveyor. The movement is considered 'synchronised' when the machine TCP is within this tolerance during synchronisation.
Parameter	conveyor_sync.sync_in_tolerance
Data type	REAL64
Data range	0.0 ≤ sync_in_tolerance ≤ MAX(REAL64)
Dimension	0.1µm
Default value	0.0
Remarks	Available as of CNC Build V4.20.0

P-CHAN-00366	**Factor for reducing the velocity when limiting movement to the end position**
Description	This parameter defines a safety factor for reducing the path velocity. It takes effect as soon as the limiting function towards the end position is active in the opposite conveying direction. The factor is usually set to 0.95, which corresponds to a reduction in velocity to 95%. The machine TCP is then taken to a safe distance from the end position. Factors that are too small are not optimal since they can increase machining time. If the value defined is too large, the warning ID 21483 is output and the value is limited to the maximum value
Parameter	conveyor_sync.hold_limit_vel_factor
Data type	UNS16
Data range	0 ≤ hold_limit_vel_factor ≤ 1000
Dimension	[0.1%]
Default value	1000
Remarks	Available as of CNC Build V4.20.0

P-CHAN-00368	**X offset of the Cartesian basic coordinate system when the conveyor tracking functionality is used**
Description	X offset of the Cartesian basic coordinate system for synchronisation. This value should correspond to the offset of the measuring unit in the X direction.
Parameter	conveyor_sync.cart_t0_shift_x
Data type	REAL64
Data range	MIN(REAL64) ≤ cart_t0_shift_x ≤ MAX(REAL64)
Dimension	[0.1 µm]
Default value	0.0
Remarks	Available as of CNC Build V4.20.0

P-CHAN-00369	**Y offset of the Cartesian basic coordinate system when the conveyor tracking functionality is used**
Description	Y offset of the Cartesian basic coordinate system for synchronisation. This value should correspond to the offset of the measuring unit in the Y direction.
Parameter	conveyor_sync.cart_t0_shift_y
Data type	REAL64
Data range	MIN(REAL64) ≤ cart_t0_shift_y ≤ MAX(REAL64)
Dimension	[0.1 µm]

Default value	0.0
Remarks	Available as of CNC Build V4.20.0

P-CHAN-00370	**Z offset of the Cartesian basic coordinate system when the conveyor tracking functionality is used**
Description	Z offset of the Cartesian basic coordinate system for synchronisation. This value should correspond to the offset of the measuring unit in the Z direction.
Parameter	conveyor_sync.cart_t0_shift_z
Data type	REAL64
Data range	MIN(REAL64) ≤ cart_t0_shift_z ≤ MAX(REAL64)
Dimension	[0.1 µm]
Default value	0.0
Remarks	Available as of CNC Build V4.20.0

P-CHAN-00371	**A rotation of the Cartesian basic coordinate system when the conveyor tracking functionality is used**
Description	A rotation of the Cartesian basic coordinate system for synchronisation. This value should correspond to the rotation of the measuring unit of the machine coordinate system about the A axis.
Parameter	conveyor_sync.cart_t0_rot_a
Data type	REAL64
Data range	0.0 ≤ cart_t0_rot_a < 3600000.0
Dimension	[0.1 mdeg]
Default value	0.0
Remarks	Available as of CNC Build V4.20.0

P-CHAN-00372	**B rotation of the Cartesian basic coordinate system when the conveyor tracking functionality is used**
Description	B rotation of the Cartesian basic coordinate system for synchronisation. This value should correspond to the rotation of the measuring unit of the machine coordinate system about the B axis.
Parameter	conveyor_sync.cart_t0_rot_b
Data type	REAL64
Data range	0.0 ≤ cart_t0_rot_b < 3600000.0
Dimension	[0.1 mdeg]
Default value	0.0
Remarks	Available as of CNC Build V4.20.0

P-CHAN-00373	**C rotation of the Cartesian basic coordinate system when the conveyor tracking functionality is used**
Description	C rotation of the Cartesian basic coordinate system for synchronisation. This value should correspond to the rotation of the measuring unit of the machine coordinate system about the C axis.
Parameter	conveyor_sync.cart_t0_rot_c
Data type	REAL64
Data range	0.0 ≤ cart_t0_rot_c < 3600000.0
Dimension	[0.1 mdeg]
Default value	0.0
Remarks	Available as of CNC Build V4.20.0

P-CHAN-00374	**End position of the X axis in the co-moving coordinate system**

Description	This parameter defines the position that must not be crossed in the X direction opposite to the conveying direction. The specified position refers to the co-moving coordinate system.
Parameter	conveyor_sync.pos_limit
Data type	REAL64
Data range	0.0 ≤ pos_limit ≤ MAX(REAL64)
Dimension	[0.1 μm]
Default value	0.0
Remarks	Available as of CNC Build V4.20.0

P-CHAN-00650	**Definition of interpolator functionalities**
Description	This parameter defines individual functionalities and the size of the look-ahead buffer in the interpolator, i.e. it defines the number of blocks required to calculate deceleration distance and dynamic planning.
Parameter	configuration.interpolator.function
Data type	STRING
Data range	See Channel parameters [▶ 55]
Dimension	----
Default value	FCT_IPO_DEFAULT
Remarks	

Interpolation function table

Identifier	**Description**
FCT_IPO_DEFAULT	FCT_LOOK_AHEAD_STANDARD
FCT_LOOK_AHEAD_LOW	30 blocks
FCT_LOOK_AHEAD_STANDARD	120 blocks
FCT_LOOK_AHEAD_HIGH	190 blocks
FCT_LOOK_AHEAD_CUSTOM	Number of look-ahead blocks in the interval [10; P-CHAN-00653].
FCT_SYNC	Synchronising an axis in coordinated motion Example: FCT_IPO_DEFAULT | FCT_SYNC
FCT_LOOK_AHEAD_OPT	The path velocity curve can be further improved for HSC machining by additional calculations. This generally reduces machining time. The additional calculations place greater demands on the controller hardware.
FCT_LIFT_UP_TIME	Automatic lifting/lowering of an axis (time-based coupling). Example: FCT_IPO_DEFAULT | FCT_LIFT_UP_TIME
FCT_SHIFT_NCBL	Path-controlled offset of M functions (dwell time). Example: FCT_IPO_DEFAULT | FCT_SHIFT_NCBL
FCT_CALC_STATE_AT_T	Calculation of path velocity at a time in the future. Function only available in combination with HSC slope and only as of V3.1.3057.0 Example: FCT_IPO_DEFAULT | FCT_CALC_STATE_AT_T
FCT_CALC_TIME	Calculation of interpolation time to next feed block (G01,G02,G03). Example: FCT_IPO_DEFAULT | FCT_CALC_TIME
FCT_CONTOUR_LAH	Contour look-ahead: advance output of motion blocks to the PLC as of V3.1.3104.07
FCT_DYN_POS_LIMIT	Dynamic limitation of axis positions
FCT_EXTENSION_EQUIDIST	Die-sinking EDM Orbiting
FCT_CALC_POS_V_0	Calculating and supplying the braking distance on the path until velocity and acceleration are 0. Supplying CNC objects on the path and assigned PCS and ACS axis positions at the end of this deceleration process. as of V3.01.3081.7 or V3.1.3114.0

FCT_DLM	Activate the conveyor tracking function. as of V4.20.0

The look-ahead buffer size values specified above apply as of CNC Builds V2.11.2800 and higher; the following settings apply to CNC Build V2.11.20xx:

FCT_LOOK_AHEAD_LOW	30 blocks
FCT_LOOK_AHEAD_STANDARD	70 blocks
FCT_LOOK_AHEAD_HIGH	120 blocks

P-CHAN-00859	**X position of the upper end position of the workspace**
Description	This parameter defines the X coordinate of the upper end position in the Cartesian workspace. The other coordinates of the upper end position are defined using P-CHAN-00860 [▸ 56] and P-CHAN-00861 [▸ 56]; the coordinates of the lower end position are defined using P-CHAN-00862 [▸ 57]/ P-CHAN-00863 [▸ 57]/ P-CHAN-00864 [▸ 57].
Parameter	conveyor_sync.cart_swe_pos_x
Data type	REAL64
Data range	
Dimension	[0.1 µm]
Default value	0.0
Remarks	Available as of CNC Build V4.20.0

P-CHAN-00860	**Y position of the upper end position of the workspace**
Description	This parameter defines the Y coordinate of the upper end position in the Cartesian workspace. The other coordinates of the upper end position are defined using P-CHAN-00859 [▸ 56] and P-CHAN-00861 [▸ 56]; the coordinates of the lower end position are defined using P-CHAN-00862 [▸ 57]/ P-CHAN-00863 [▸ 57]/ P-CHAN-00864 [▸ 57].
Parameter	conveyor_sync.cart_swe_pos_y
Data type	REAL64
Data range	
Dimension	[0.1 µm]
Default value	0.0
Remarks	Available as of CNC Build V4.20.0

P-CHAN-00861	**Z position of the upper end position of the workspace**
Description	This parameter defines the Z coordinate of the upper end position in the Cartesian workspace. The other coordinates of the upper end position are defined using P-CHAN-00859 [▸ 56] and P-CHAN-00860 [▸ 56]; the coordinates of the lower end position are defined using P-CHAN-00862 [▸ 57]/ P-CHAN-00863 [▸ 57]/ P-CHAN-00864 [▸ 57].
Parameter	conveyor_sync.cart_swe_pos_z
Data type	REAL64
Data range	
Dimension	[0.1 µm]
Default value	0.0
Remarks	Available as of CNC Build V4.20.0

P-CHAN-00862	X position of the lower end position of the workspace
Description	This parameter defines the X coordinate of the upper end position in the Cartesian workspace. The other coordinates of the lower end position are defined using P-CHAN-00863 [▶ 57] and P-CHAN-00864 [▶ 57]; the coordinates of the upper end position are defined using P-CHAN-00859 [▶ 56]/ P-CHAN-00860 [▶ 56]/ P-CHAN-00861 [▶ 56].
Parameter	conveyor_sync.cart_swe_neg_x
Data type	REAL64
Data range	
Dimension	[0.1 µm]
Default value	0.0
Remarks	Available as of CNC Build V4.20.0

P-CHAN-00863	Y position of the lower end position of the workspace
Description	This parameter defines the Y coordinate of the lower end position in the Cartesian workspace. The other coordinates of the lower end position are defined using P-CHAN-00862 [▶ 57] and P-CHAN-00864 [▶ 57]; the coordinates of the upper end position are defined using P-CHAN-00859 [▶ 56]/ P-CHAN-00860 [▶ 56]/ P-CHAN-00861 [▶ 56].
Parameter	conveyor_sync.cart_swe_neg_y
Data type	REAL64
Data range	
Dimension	[0.1 µm]
Default value	0.0
Remarks	Available as of CNC Build V4.20.0

P-CHAN-00864	Z position of the lower end position of the workspace
Description	This parameter defines the Z coordinate of the lower end position in the Cartesian workspace. The other coordinates of the lower end position are defined using P-CHAN-00862 [▶ 57] and P-CHAN-00863 [▶ 57]; the coordinates of the upper end position are defined using P-CHAN-00859 [▶ 56]/ P-CHAN-00860 [▶ 56]/ P-CHAN-00861 [▶ 56].
Parameter	conveyor_sync.cart_swe_neg_z
Data type	REAL64
Data range	
Dimension	[0.1 µm]
Default value	0.0
Remarks	Available as of CNC Build V4.20.0

6 Interfaces

6.1 Display data

The structure gpCh[channel_idx]^.bahn_state.**conveyor_data** supplies the state signals for this functionality via the HLI.

The variables current_pos_master, current_v_master, current_a_master indicate the unfiltered position, velocity and acceleration values for the linear conveyor with the logical axis number log_master_axis_nr.

The variables pos_master_comp, v_master_comp, a_master_comp indicate the filtered values.

Logical axis number of the linear conveyor	
Description	Indicates the logical axis number of the linear conveyor.
Signal flow	CNC → PLC
ST path	gpCh[*channel_idx*]^.bahn_state.conveyor_data.**log_master_axis_nr**
Data type	UINT
Unit	[-]
Value range	[1, MAX_UNS16]
Access	PLC is reading
Special feature	Available as of CNC Build V4.20.0

Current unfiltered actual position of the workpiece in the currently selected PCS coordinate system	
Description	This variable indicates the unfiltered actual position of the workpiece in the currently selected PCS coordinate system. This variable is only valid after a measured value is received for the current workpiece. The variable considers the measured value as well as the offset of the measuring system by the configured T_0 coordinate system.
Signal flow	CNC → PLC
ST path	gpCh[channel_idx]^.bahn_state.conveyor_data.**current_pos_master**
Data type	DINT
Unit	[0.1 μm]
Value range	[MIN_SGN32, MAX_SGN32]
Access	PLC is reading
Special feature	Available as of CNC Build V4.20.0

Current unfiltered actual velocity of the linear conveyor	
Description	This variable indicates the unfiltered actual velocity of the linear conveyor.
Signal flow	CNC → PLC
ST path	gpCh[channel_idx]^.bahn_state.conveyor_data.**current_v_master**
Data type	DINT
Unit	[0.1 μm/s]
Value range	[MIN_SGN32, MAX_SGN32]
Access	PLC is reading
Special feature	Available as of CNC Build V4.20.0

Current unfiltered actual acceleration of the linear conveyor	
Description	This variable indicates the unfiltered actual acceleration of the linear conveyor.
Signal flow	CNC → PLC
ST path	gpCh[channel_idx]^.bahn_state.conveyor_data.**current_a_master**
Data type	DINT

Unit	[0.1 µm/s²]
Value range	[MIN_SGN32, MAX_SGN32]
Access	PLC is reading
Special feature	Available as of CNC Build V4.20.0

Current filtered actual position of the workpiece in the currently selected PCS coordinate system	
Description	This variable indicates the filtered actual position of the workpiece in the currently selected PCS coordinate system. This variable is only valid after a measured value is received for the current workpiece. The variable considers the measured value as well as the offset of the measuring system by the configured T_0 coordinate system.
Signal flow	CNC → PLC
ST path	gpCh[channel_idx]^.bahn_state.conveyor_data.**pos_master_comp**
Data type	DINT
Unit	[0.1 µm]
Value range	[MIN_SGN32, MAX_SGN32]
Access	PLC is reading
Special feature	Available as of CNC Build V4.20.0

Current filtered actual velocity of the linear conveyor.	
Description	This variable indicates the filtered actual velocity of the linear conveyor.
Signal flow	CNC → PLC
ST path	gpCh[channel_idx]^.bahn_state.conveyor_data.**v_master_comp**
Data type	DINT
Unit	[0.1 µm/s]
Value range	[MIN_SGN32, MAX_SGN32]
Access	PLC is reading
Special feature	Available as of CNC Build V4.20.0

Current filtered actual acceleration of the linear conveyor	
Description	This variable indicates the filtered actual acceleration of the linear conveyor.
Signal flow	CNC → PLC
ST path	gpCh[channel_idx]^.bahn_state.conveyor_data.**a_master_comp**
Data type	DINT
Unit	[0.1 µm/s²]
Value range	[MIN_SGN32, MAX_SGN32]
Access	PLC is reading
Special feature	Available as of CNC Build V4.20.0

Position difference: Actual position of the conveyor and machine position.

Description	If programmed correctly (see section Synchronising the workpiece on the linear conveyor [▶ 34]), the variable indicates the current difference between the actual position of the machine axis and the feedforward controlled actual position of the workpiece. When synchronisation is ideal, the position difference should be zero. However, deviations may result from delays on the control section or an inconstant velocity signal. These variable is used to determine the deviation and therefore the delay time between the latch moment of the linear conveyor actual values and generating the command value for the machine axis. The delay time is parameterised by the parameter conv_sync.delay_time (P-AXIS-00626 [▶ 51]). If this variable has an average value unequal to zero, set the delay time as follows: - synchro_lag > 0 -> delay_time set too high - synchro_lag > 0 -> delay_time set too low
Signal flow	CNC → PLC
ST path	gpCh[channel_idx]^.bahn_state.conveyor_data.**synchro_lag**
Data type	DINT
Unit	[0.1 μm]
Value range	[MIN_SGN32, MAX_SGN32]
Access	PLC is reading
Special feature	Available as of CNC Build V4.20.0

6.2 CNC objects

Name	mc_active_execution_mode_r		
Description	This object reads the active channel operation mode. See value range Channel operation modes [▶ 61]		
Task	COM (Port 553)		
Index group	0x120101	**Index offset**	0x41
Data type	SGN32	**Length/byte**	4
Attributes	read	**Unit**	
Remarks			

Name	mc_command_execution_mode_r		
Description	This object reads the commanded channel operation mode. See value range Channel operation modes [▶ 61]		
Task	COM (Port 553)		
Index group	0x120101	**Index offset**	0x40
Data type	SGN32	**Length/byte**	4
Attributes	read	**Unit**	[-]
Remarks			

Name	mc_command_execution_mode_w		
Description	This object defines and enables the channel operation mode. See value range Channel operation modes [▶ 61]		
Task	COM (Port 553)		
Index group	0x120101	**Index offset**	0x3F
Data type	SGN32	**Length/byte**	4
Attributes	write	**Unit**	[-]
Remarks			

Channel operation mode	
Description	Selects a special channel operation mode, e.g. syntax check or machining time calculation
Data type	MC_CONTROL_SGN32_UNIT, see description of Control Unit
Access	PLC reads request_r + state_r and writes command_w + enable_w
ST path	gpCh[*channel_idx*]^.decoder_mc_control.**execution_mode**
Commanded, requested and return values	
ST element	**.command_w** **.request_r** **.state_r**
Data type	DINT

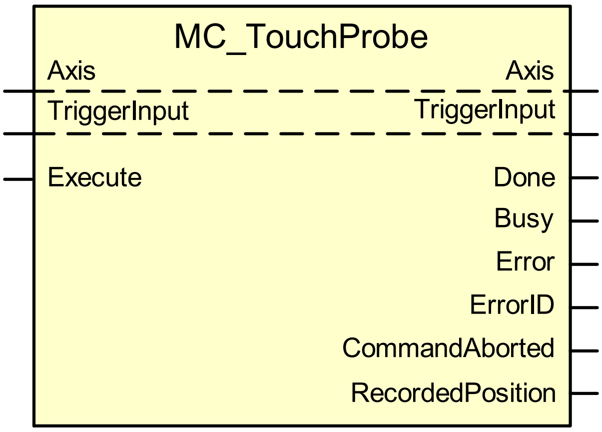
Value range	Value	Constant	Meaning
	0x0000	ISG_STANDARD	Normal mode
	0x0001	SV	Block search
	0x0002	SOLLKON	Nominal contour visualisation simulation with output of visualisation data
	0x0802	SOLLKON_SUPRESS_OUTPUT & SOLLKON	Nominal contour visualisation simulation without output of visualisation data
	0x0004	ON_LINE	Online visualisation simulation
	0x0008	SYNCHK	Syntax check simulation
	0x0010	PROD_TIME	Simulation machining time calculation (in TwinCAT without function)
	0x0020	ONLINE_PROD_TIME	Simulation of online machining time calculation
	0x0040	MACHINE_LOCK	Dry run without axis motion
	0x0080	ADD_MDI_BLOCK	Extended manual block mode: the end of a manual block is not evaluated as a program end. It permits the commanding of further manual blocks.
	0x0100	KIN_TRAFO_OFF	Overwrites automatic enable for kinematic transformations by a characteristic parameter defined in the channel parameters (sda_mds*.lis).
	0x1000	BEARB_MODE_SCENE	When SCENE mode is enabled, the output of #SCENE commands is activated on the interface (see also [FCT-C17// Scene contour visualisation]). An additional client is linked to this output via DataFactory / CORBA.
	0x2000	SUPPRESS_TECHNO_OUTPUT	Without output of technology functions (M/H/T). Set implicitly in connection with syntax check.
	0x10000	SUPPRESS_POSITION_REQUEST	Fast program start without position request at program start
	0x20000	SUPPRESS_PROG_START_INIT	Suppress program start sequence for machining on the belt
Redirection			
ST element	.enable_w		

6.3 PLCopen blocks

6.3.1 MC_TouchProbe

The MC_TouchProbe FB records an axis position when a trigger event occurs. The measurement channel and method (rising or falling edge of the trigger signal) are defined via the reference for the trigger signal source.

Blockdiagramm



Parameters of the FB

VAR_IN_OUT		
Variable name	**Data type**	**Description**
axis	AXIS_REF	Axis reference
TriggerInput	TRIGGER_REF	Reference for the trigger signal source (see chapter entitled Data structure TRIGGER_REF).

VAR_INPUT		
Variable name	**Data type**	**Description**
Execute	BOOL	Starts the command on the rising edge.

VAR_OUTPUT		
Variable name	**Data type**	**Description**
Done	BOOL	Trigger event was detected.
Busy	BOOL	Is TRUE as long as the input/output waits for the trigger event.
Error	BOOL	Indicates TRUE if an error occurs in the FB.
ErrorID	WORD	Error code.
CommandAborted	BOOL	Command is aborted by another command of an FB MC_AbortTrigger.
RecordedPosition	LREAL	Axis position when the trigger event occurred. Default unit [0.1µm or 10^{-4} °]

Behaviour of the FB

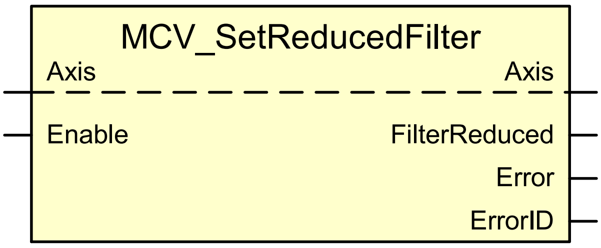
- A measurement command is issued only when the axis is in the "Standstill" or "Continuous Motion" or "Discrete Motion" or "Synchronized Motion" or "Stopping" state. Moreover, no measurement command must be active in the measurement channel specified by "TriggerInput". The measurement channel must therefore be in the "TP_IDLE" state. If this is not the case, the "error" output is set to FALSE and "error_id" indicates an error value that designates the state the axis is currently in.

- If a measurement command in a measurement channel is active, no further measurement command in relation to this measurement channel can be sent. Therefore, the FB is not retriggerable. If a command is still sent, the “Error“ output becomes TRUE and signals the error P-ERR-44009 (ERR_PO_AX_TPROBE_RETRIG) at the “ErrorID“ output.
- Errors are also generated when the axis reference, the measuring channel or the trigger event of the measurement process were changed while a measurement command is active. The following error may occur in these cases:
- P-ERR-44001 (ERR_PO_AX_REF_CHG_WHILE_ACTIVE)
- P-ERR-44011 (ERR_PO_AX_INPREF_TPCH_CHW_ACTIVE)
- P-ERR-44012 (ERR_PO_AX_INPREF_TPMD_CHW_ACTIVE).

6.3.2 MCV_SetReducedFilter

The “MCV_SetReducedFilter” FB influences the filtering of actual values of an axis and is usually used in combination with an axis configured as an encoder axis (see P-AXIS-00015) which is not controlled by the motion controller. The FB is only effective if a slave axis was configured for this encoder axis.

Blockdiagramm



Parameters of the FB

VAR_IN_OUT		
Variable name	**Data type**	**Description**
axis	AXIS_REF	Axis reference

VAR_INPUT		
Variable name	**Data type**	**Description**
enable	BOOL	TRUE, the parameter for actual value filtering is activated for a reduced number of values.

VAR_OUTPUT		
Variable name	**Data type**	**Description**
FilterReduced	BOOL	TRUE; Actual value filtering is then executed by the reduced number of values which were parameterised.
Error	BOOL	Indicates TRUE if an error occurs in the FB.
ErrorID	WORD	Error code.

Behaviour of the FB

- To parameterise the filter effect, the parameters of the conv_sync group must be configured accordingly in the axis parameter list ([AXIS]).

7 Support and Service

Beckhoff and their partners around the world offer comprehensive support and service, making available fast and competent assistance with all questions related to Beckhoff products and system solutions.

Download finder

Our download finder contains all the files that we offer you for downloading. You will find application reports, technical documentation, technical drawings, configuration files and much more.

The downloads are available in various formats.

Beckhoff's branch offices and representatives

Please contact your Beckhoff branch office or representative for local support and service on Beckhoff products!

The addresses of Beckhoff's branch offices and representatives round the world can be found on our internet page: www.beckhoff.com

You will also find further documentation for Beckhoff components there.

Beckhoff Support

Support offers you comprehensive technical assistance, helping you not only with the application of individual Beckhoff products, but also with other, wide-ranging services:

- support
- design, programming and commissioning of complex automation systems
- and extensive training program for Beckhoff system components

Hotline:	+49 5246 963-157
e-mail:	support@beckhoff.com

Beckhoff Service

The Beckhoff Service Center supports you in all matters of after-sales service:

- on-site service
- repair service
- spare parts service
- hotline service

Hotline:	+49 5246 963-460
e-mail:	service@beckhoff.com

Beckhoff Headquarters

Beckhoff Automation GmbH & Co. KG

Huelshorstweg 20
33415 Verl
Germany

Phone:	+49 5246 963-0
e-mail:	info@beckhoff.com
web:	www.beckhoff.com

Index

More Information:
www.beckhoff.com/TF5264

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