

Manual | EN

# NC Flying Saw

TwinCAT 2





# Table of contents

<b>1 Foreword</b> .....	<b>5</b>
1.1 Notes on the documentation.....	5
1.2 Safety instructions .....	6
<b>2 General</b> .....	<b>7</b>
<b>3 Interfaces</b> .....	<b>9</b>
<b>4 Procedure</b> .....	<b>10</b>
<b>5 Synchronisation Conditions</b> .....	<b>12</b>
<b>6 Stop</b> .....	<b>14</b>
<b>7 Position compensation</b> .....	<b>16</b>
<b>8 Diagonal Flying Saw</b> .....	<b>18</b>
<b>9 Flying Saw and Flying Diagonal Saw at Velocity</b> .....	<b>19</b>
<b>10 Flying Saw with Online Conversion to Master PTP Axis</b> .....	<b>20</b>
<b>11 TcNcFlyingSaw</b> .....	<b>21</b>
11.1 Classical Flying Saw.....	21
11.1.1 FlyingSawCoupleOnPositon .....	21
11.1.2 FlyingSawCoupleOnVelocity .....	22
11.1.3 FlyingSawStop.....	23
11.1.4 Get_TcNcFlyingSaw_Version.....	23
11.2 PLCopen Function blocks.....	24
11.2.1 MC_GearInPos .....	24
11.2.2 MC_GearInVelo .....	26
11.2.3 MC_ReadFlyingSawCharacteristics .....	27
11.2.4 The ST_SyncMode data structure .....	28



# 1 Foreword

## 1.1 Notes on the documentation

This description is only intended for the use of trained specialists in control and automation engineering who are familiar with applicable national standards.

It is essential that the documentation and the following notes and explanations are followed when installing and commissioning the components.

It is the duty of the technical personnel to use the documentation published at the respective time of each installation and commissioning.

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

### Disclaimer

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

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

### Trademarks

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

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

### Patent Pending

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

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

## EtherCAT®

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

### Copyright

© Beckhoff Automation GmbH & Co. KG, Germany.

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

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

## 1.2 Safety instructions

### Safety regulations

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

### Exclusion of liability

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

### Personnel qualification

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

### Description of symbols

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

#### **DANGER**

##### **Serious risk of injury!**

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

#### **WARNING**

##### **Risk of injury!**

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

#### **CAUTION**

##### **Personal injuries!**

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

#### **NOTE**

##### **Damage to the environment or devices**

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



##### **Tip or pointer**

This symbol indicates information that contributes to better understanding.

## 2 General

The **Flying Saw** (parallel slave axis) is a slave axis, moving in parallel with the master axis, whose velocity is synchronised, at a particular position, to the master velocity of the moving master axis. It is understood here that in coupling the slave axis to the master axis, the master axis is moving with a (constant) set velocity, and that the slave axis is at its base (coupling) position. The slave axis starts automatically when the master axis reaches the dynamic coupling position. The whole system is partially implemented as a linear path controller (slave acceleration phase (jerk restricted) and slave parallel travel) and partly under time control (slave delay phase (jerk restricted) and slave stationary).

The **diagonal flying saw** (diagonal slave axis) is a slave axis that is velocity synchronised to the moving master axis at a given position diagonally (with a constant angle) in such a way that the velocity component of the slave axis in the direction of the master's motion following the synchronisation is equal to the master velocity.

### Interfaces

[Interfaces \[► 9\]](#)

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

### Procedure

[Procedure \[► 10\]](#)

A procedure consists of these phases:

1. Logical coupling to the master axis (the slave axis remains stationary).
2. Dynamic coupling to the master axis: acceleration phase.
3. Movement in parallel with the master axis.
4. Stop: dynamic uncoupling from the master axis: braking phase.
5. Stationary.
6. Logical uncoupling.

### Synchronisation Conditions

[Synchronisation Conditions \[► 12\]](#) The synchronisation conditions for the jerk restricted linear acceleration algorithm implies **forced conformity between the coupling positions** of the master and slave and the **synchronisation positions** of master and slave.

### Start and Stop

[Start and Stop \[► 14\]](#)

When the axis is started, and after the dynamic coupling position has been reached, the start parameters (master synchronisation position and slave synchronisation position) and the general dynamic parameters are used to generate a jerk restricted acceleration profile with the current master velocity as the set velocity with every call of the block execution task. This is used to generate this set values for linear path control of the slave axis.

To stop, the general dynamic data is used to calculate a time-controlled braking profile, the slave axis is dynamically uncoupled from the master, and braked in accordance with the braking profile.

### Position compensation

[Position compensation \[► 16\]](#)

Online position correction of the flying saw, during which the slave axis is moved along a specified path by a certain additional or reduced distance compared with the coupling to the master axis. The compensation dynamics must be parameterised. If the desired compensation cannot be carried out with the specified

compensation start data, the maximum possible compensation will automatically be carried out. Compensation may be terminated prematurely via a compensation stop. The compensation is only to be used during the synchronisation phase.

**Warning:**

If the compensation difference  $D$  is negative (deceleration), (double) **movement reversal** is possible.

**Diagonal Flying Saw**

[Diagonal Flying Saw \[► 18\]](#)

If  $v$  is the current set velocity of the parallel slave axis, then the velocity of the diagonal slave axis is  $v/\sin(\theta)$ , where  $\theta$ ,  $0.0^\circ < \theta \leq 90.0^\circ$ , indicates the complementary angle between the master direction and the slave direction.

**Flying Saw and Flying Diagonal Saw at Velocity**

[Flying Saw and Flying Diagonal Saw at Velocity \[► 19\]](#)

The **Flying Saw at Velocity (parallel slave axis)** is a slave axis moving in parallel with a moving master axis, whose velocity is synchronised as quickly as possible to the velocity of the master axis. It is understood here that in coupling the slave axis to the master axis, the master axis is moving with a (constant) set velocity, and that the slave axis is at its base (coupling) position. The whole system is partially implemented as a linear path controller (slave acceleration phase (jerk restricted) and slave parallel travel) and partly under time control (slave delay phase (jerk restricted) and slave stationary).

The **diagonal flying saw at velocity (diagonal slave axis)** is a slave axis whose velocity is synchronised to the moving master axis as quickly as possible diagonally (with a constant angle) in such a way that the velocity component of the slave axis in the direction of the master's motion following the synchronisation is equal to the master velocity.

**Flying Saw with Online Conversion to Master PTP Axis**

[Flying Saw with Online Conversion to Master PTP Axis \[► 20\]](#)

This versatile slave type is started as a flying saw. It is coupled at the synchronisation point to the velocity of the master axis. It can then be locally synchronised again by means of position compensation in the synchronous phase, after which it can be uncoupled online in the synchronous phase from the master and can itself be converted into a PTP master axis. At this point force can be removed, or it can continue to operate as an endless axis at a velocity stated during the coupling or uncoupling instruction. Finally a "new end position" instruction can be issued to move it to its final destination.



## 3 Interfaces

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

### PLC Document

There is special PLC documentation for the flying saw.

### System Manager

The System Manager offers access to all the functionalities of the flying saw:

- Logical coupling, stating the master axis and setting the synchronisation positions.
- Stop.

### PLC Blocks

The PLC Library offers access to the complete slave axis functionality

- Logical coupling, stating the master axis and setting the synchronisation positions.
- Stop.

### NC-PLC-Interface

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

## 4 Procedure

A procedure consists of these phases:

1. Logical coupling to the master axis (the slave axis remains stationary).
2. Dynamic coupling to the master axis: Acceleration phase.
3. Movement in parallel with the master axis.
4. Stop: dynamic uncoupling from the master axis: braking phase.
5. Stationary
6. Logical uncoupling.

### Logical Coupling

Nearly all the functionality of the slave axis is blocked by (logical) coupling. The axis is in its base position, and waits for the master axis to pass through the dynamic coupling position.

### Dynamic Coupling

When the master axis passes through the dynamic coupling position, the slave axis starts **automatically**. Every time the block execution task is called, a full jerk restricted acceleration profile from velocity  $0.0$  to the **current master velocity** is calculated. The latest value is passed to the slave axis as a set value via a linear path controller.

### Acceleration phase

The acceleration phase is designed as a **linear path controller**. This means that the slave axis follows the current data for the master axis, both in terms of velocity and of position, so that it follows all the changes in velocity and motion inversions of the master axis. When the master axis passes through its synchronisation position, the slave axis has also reached its synchronisation position. It has thereby reached the current master velocity, and the parallel phase begins.

### The Parallel Phase

In the parallel phase, the flying saw is coupled to the master in a condition very similar **to that of the linear slave axis** (with coupling factor  $1.0$ ).

### Dynamic Uncoupling

The user stops the slave to end the parallel travel. In this process, a jerk restricted time controlled dynamic profile is generated from the current dynamic state (acceleration, velocity) down to a velocity of  $0.0$ . The slave is dynamically uncoupled from the master, and braked according to this calculated dynamic profile.

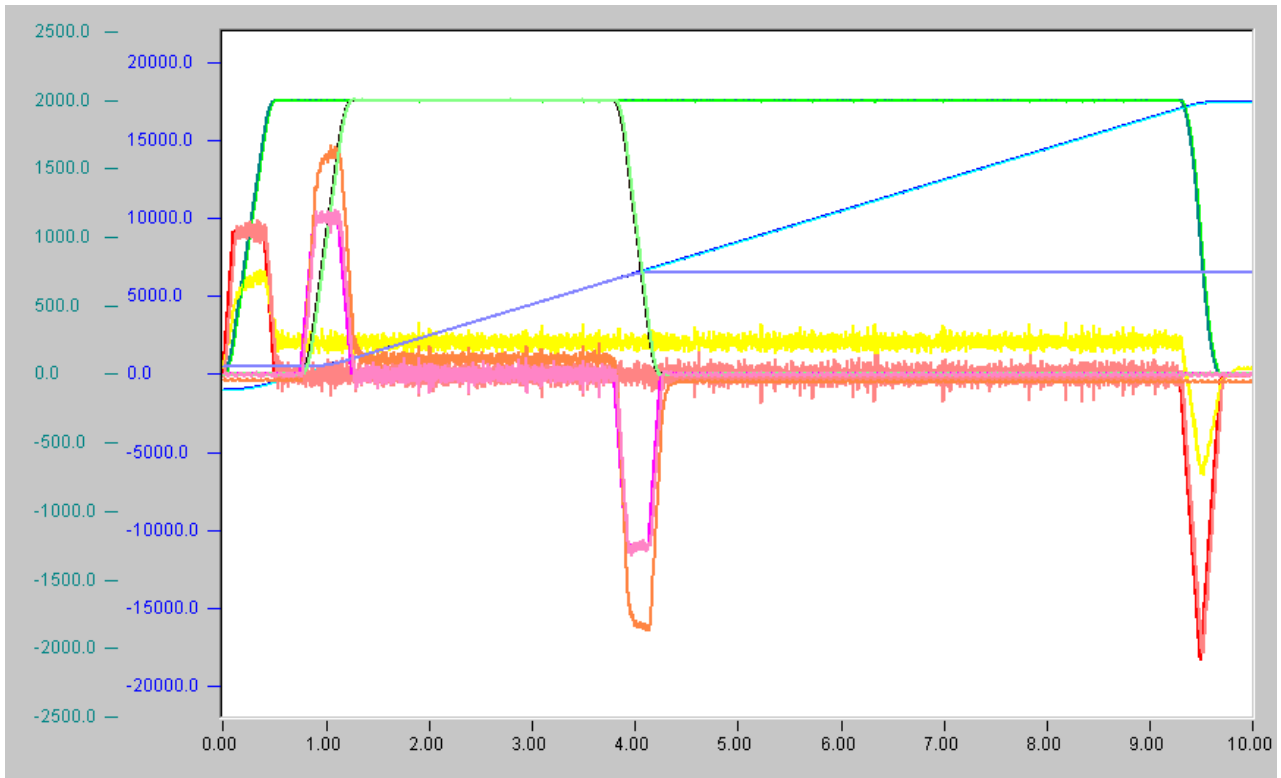
### Logical Uncoupling

The position at which the slave becomes stationary depends, in addition to the velocity, very much on the acceleration state of the axis (at the moment when the stop function is called). Once it is stationary, the slave axis can be uncoupled from the master axis.

Logical uncoupling is possible at any time, but uncoupling during motion does result in a following error.

### Reverse motion stop

In the linear path controlled phases (acceleration, parallel travel) the slave axis is synchronised to the master axis in such a way that it follows every reverse motion. To prevent the slave axis from travelling beyond its base position (support) in the event of a large reverse movement, a reverse motion stop is provided to prevent the reverse motion from extending beyond the base position. This stop functions instantaneously (following error). When the master moves again through its old coupling position in the original direction, the slave automatically starts up again.



Sequence: master start, slave start, synchronisation, synchronisation phase, slave stop, master stop.

## 5 Synchronisation Conditions

The fundamental axis synchronisation for the flying saw depends crucially on the acceleration algorithm of the slave axis. We elucidate here the axis synchronisation conditions for the jerk restricted 3-phase algorithm. The synchronisation conditions are based on the following axis variables (it should be noted that the master and the slave may have **different co-ordinate systems** using the same units):

1. Slave synchronisation position,
2. Slave base position = logical slave coupling position = dynamic slave coupling position,
3. Master synchronisation position,
4. Logical master coupling position (position of the master axis for master-slave coupling),
5. Dynamic master coupling position (position of the master axis for dynamic slave start).

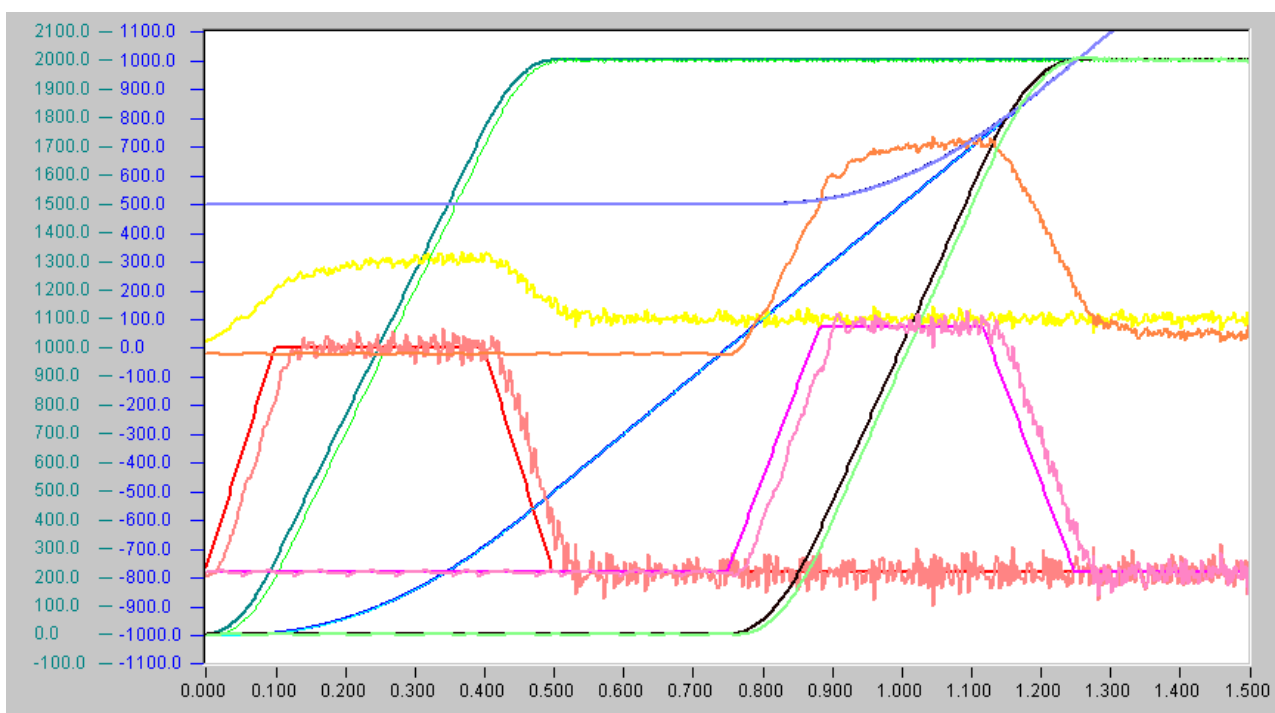
### Synchronisation Conditions for the Jerk Restricted Acceleration Algorithm

The slave axis requires a certain time to pass from its initial condition (base position, velocity  $0.0$ , acceleration  $0.0$ ) to its final condition (synchronisation position, master velocity, master acceleration). During this time the master axis covers the distance from the dynamic master coupling position to the master synchronisation position. This means that three of the four positions ( slave synchronisation position, slave coupling position, master synchronisation position, dynamic master coupling position ) are independent, and the fourth is then given. The fundamental algorithm implies the following relationship between the positions:

**[dynamic master coupling position - master synchronisation position] = 2 × [slave coupling position - slave synchronisation position].**

It follows that the distance between the dynamic master coupling position and the master synchronisation position must be **greater** than twice the distance between the slave coupling position and the slave synchronisation position. At start, the two synchronisation positions are stated explicitly and the slave coupling position is known internally, so that the dynamic master coupling position can be calculated from these. If the synchronisation cannot be executed because the master is already too close to the coupling position, the coupling is aborted with an error.

Synchronisation conditions for the flying diagonal saw [► 18].



Synchronisation conditions:  $|0.0 - 1000.0| = 2.0 \times |5000.0 - 1000.0|$

Master start at -1000.0 mm, master coupling position-logical -200.0 mm, master coupling position-dynamic 0.0 mm, master synchronisation position 1000.0 mm.

Slave waiting position = slave coupling position-logical = slave coupling position-dynamic 500.0 mm, slave synchronisation position = 1000.0 mm.

It is sufficient if the master coupling position-logical is definitely smaller than the master coupling position-dynamic.

## 6 Stop

After the dynamic coupling position has been reached, the start parameters (master synchronisation position and slave synchronisation position) and the general dynamic parameters are used to generate a jerk restricted acceleration profile with the current master velocity as the set velocity with every call of the block execution task. This is used to generate this set values for linear path control of the slave axis.

To stop, the general dynamic data is used to calculate a time-controlled braking profile, the slave axis is dynamically uncoupled from the master, and braked in accordance with the braking profile.

### Coupling Parameters

$P_{ms}$  (master synchronisation position in the master co-ordinate system)

$P_{ss}$  (slave synchronisation position in the slave co-ordinate system)

$\theta$  Diagonal Angle

cf Coupling factor to master velocity. The coupling factor must not change the direction of the axis.

Internal parameters: ( $P_{sc}$  slave coupling position,  $P_{mcd}$  dynamic master coupling position)

There is **no explicit start**. The start data is calculated internally, and the start is **automatically** initiated.

Table 1: Start parameters

Start parameters	Meaning and boundary conditions
<b>Master synchronisation position</b> $P_{ms}$	Position at which the master should be located (in its co-ordinate system) when the slave is to be a synchronised to it, $ P_{ms} - P_{mcd}  = 2 \times  P_{ss} - P_{sc} $
<b>Slave synchronisation position</b> $P_{ss}$	Position at which the slave should be located (in its co-ordinate system) when it is to be a synchronised to the master, $ P_{ms} - P_{mcd}  = 2 \times  P_{ss} - P_{sc} $
<b>Coupling factor</b> cf	Coupling factor $cf \neq 0.0$
<b>Diagonal Angle</b> $\theta$	Complementary angle between master direction and slave direction, $0.0^\circ < \theta \leq 90.0^\circ$ , parallel slave: $\theta = 90.0^\circ$

### Interfaces

System Manager

PLC Blocks

### Stop

Stop has the **highest priority**, and must function in all dynamic phases except the stop phase itself (not in the preliminary phase and not in the stationary phase). Stop also functions if reverse motion occurs. If the slave axis is subject to acceleration, an additional phase is inserted in front of the 3-phase algorithm for the stop dynamics unless this would entail an inversion of the dynamics. During this additional phase, the slave axis changes from its current acceleration to an acceleration of 0.0.

**Attention:**the stop of the flying saw must be finished while the master axis is running otherwise the stop is locked.

There are **no explicit stop parameters**. These (delay, jerk) are taken from the global dynamic data for the axis.

### Interfaces

System Manager

PLC Blocks

## 7 Position compensation

Online position correction of the flying saw, during which the slave axis is moved along a specified path by a certain additional or reduced distance compared with the coupling to the master axis. The compensation dynamics must be parameterised. If the desired compensation cannot be carried out with the specified compensation start data, the maximum possible compensation will automatically be carried out. Compensation may be terminated prematurely via a compensation stop. The compensation is only to be used during the synchronisation phase.

### Warning:

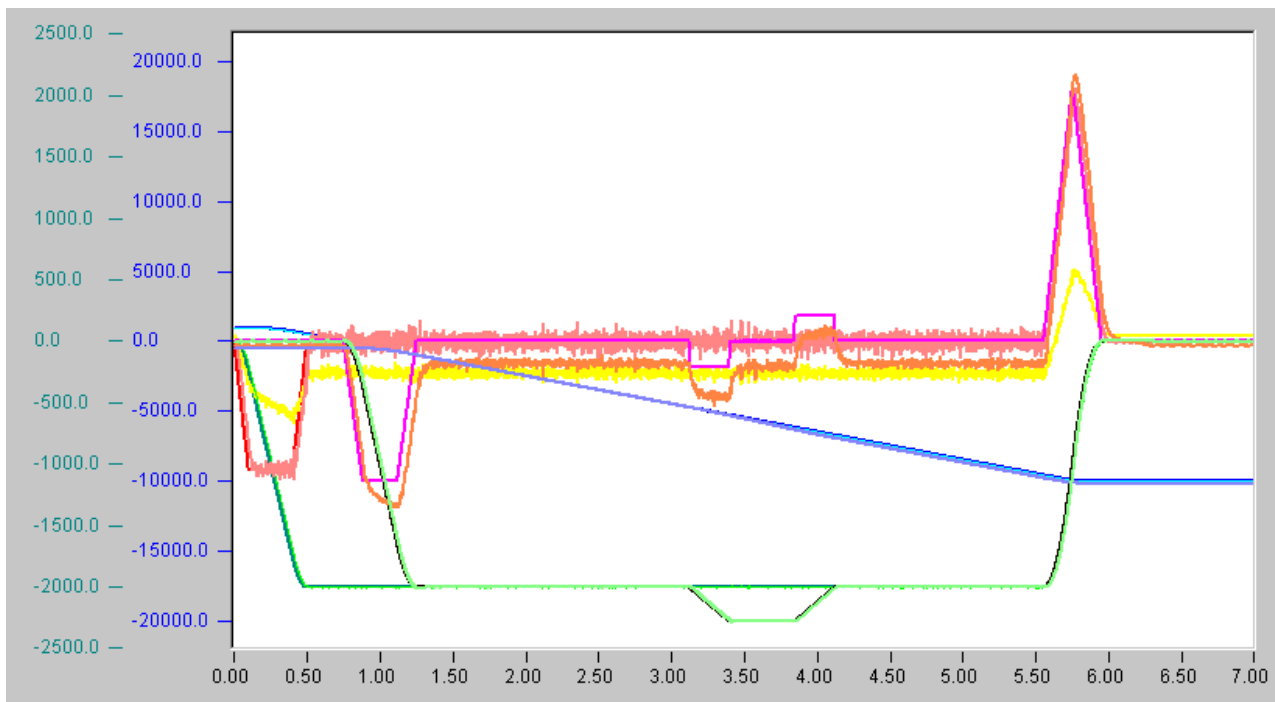
If the compensation difference  $D$  is negative (deceleration), (double) **movement reversal** is possible.

### General notes

Position compensation for slave axes is a variation of position compensation for linear slave axes .

### Special features

1. The compensation should only be performed after successful synchronisation (i.e. following the end of the acceleration phase).
2. The compensation should be complete before the braking phase begins.
3. The compensation takes the positional extension of the diagonal saw into account.

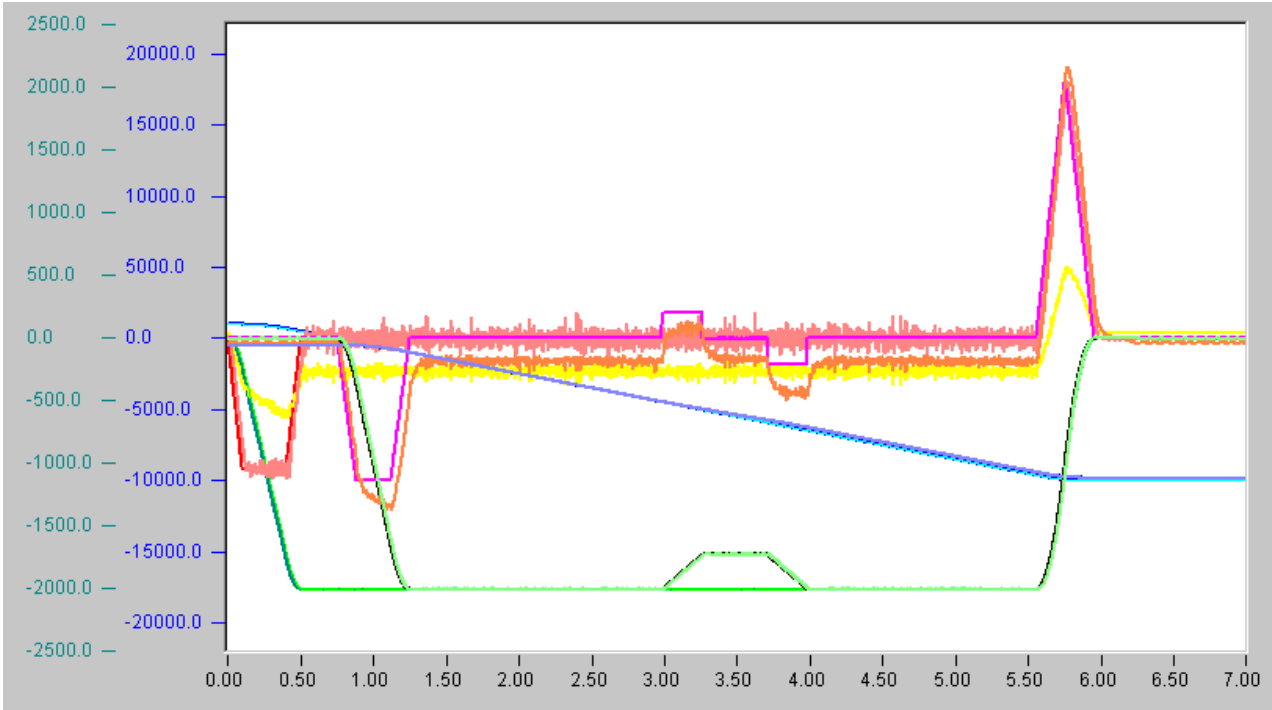


Master data: Start from 1000.0 to -10,000.0 with synchronisation position -1000.0

Slave start position -500.0 with slave synchronisation position -1000.0

Compensation data:  $A_{\pm} = 1000.0$  mm/sec/sec,  $V = 500.0$  mm/sec,  $VP = 2000.0$  mm/sec, compensation difference  $\pm 200.0$  mm, compensation length 2000.0 mm.





## 8 Diagonal Flying Saw

### Overview

The **diagonal flying saw (diagonal slave)** is a slave axis that is velocity synchronised to the moving master axis diagonally (with a constant angle  $\theta$ ) in such a way that the velocity component of the slave axis in the direction of the master's motion is equal to the master velocity.

### Geometry: Diagonal Angle $\theta$

If  $v$  is the current set velocity of a parallel slave axis, then the velocity of the diagonal slave axis is  $v/\sin(\theta)$ , where  $\theta$  ( $0.0^\circ < \theta \leq 90.0^\circ$ ), indicates the **complementary angle between the master direction and the slave direction**. In the synchronisation phase, therefore, the slave axis is coupled to the master axis with a coupling factor of  $1/\sin(\theta)$ .

Diagonal saw: geometry

### Synchronisation Conditions for the Jerk Restricted/Linear Diagonal Acceleration Algorithm

The diagonal slave axis requires a certain time to pass from its initial condition (base position, velocity  $0.0$ , acceleration  $0.0$ ) to its final condition (synchronisation position, master velocity, master acceleration). During this time the master axis covers the distance from the dynamic master coupling position to the master synchronisation position. The basic algorithms require that:

**$|\text{Slave coupling position} - \text{slave synchronisation position}| = 0.5 \times |\text{dynamic master coupling position} - \text{master synchronisation position}| / \sin(\theta)$ .**

It follows that the distance between the dynamic master coupling position and the master synchronisation position must be **greater** than  $2.0 \times \sin(\theta)$  times the distance between the slave coupling position - slave synchronisation position.

### Dynamics

The velocity of the slave axis once it has reached the parallel phase is equal to  $1.0/\sin(\theta)$  times the velocity of the master axis at that time. This velocity is greater than the velocity of the master axis, because the slave axis, in addition to the parallel component (which is the master velocity) must also have an orthogonal component.

# 9 Flying Saw and Flying Diagonal Saw at Velocity

## Overview

The **Flying Sawat Velocity (parallel slaveaxis)** is a slave axis moving in parallel with a moving master axis, whose velocity is synchronised as quickly as possible to the velocity of the master axis. It is understood here that in coupling the slave axis to the master axis, the master axis is moving with a (constant) set velocity, and that the slave axis is at its base (coupling) position. The whole system is partially implemented as a linear path controller (slave acceleration phase (jerk restricted) and slave parallel travel) and partly under time control (slave delay phase (jerk restricted) and slave stationary).

The **diagonal flying sawat velocity (diagonal slaveaxis)** is a slave axis whose velocity is synchronised to the moving master axis as quickly as possible diagonally (with a constant angle) in such a way that the velocity component of the slave axis in the direction of the master's motion following the synchronisation is equal to the master velocity.

## General

Apart from the way in which the coupling is executed, the flying (diagonal) saw at velocity is identical in its algorithms to the flying (diagonal) saw to position (and velocity). The direction of the slave axis is here specified as the direction of the master axis. If, at the time of coupling, the master axis does not have any direction (e.g. an encoder axis that happens to be stationary), the coupling is aborted.

## Coupling Parameters

$\theta$  Diagonal Angle

The additional start data is calculated internally, and the start is **automatically** initiated straight away.

Table 2: Start parameters

Start parameters	Meaning and boundary conditions
Diagonal Angle $\theta$	Complementary angle between master direction and slave direction, $0.0^\circ < \theta \leq 90.0^\circ$ , parallel slave: $\theta = 90.0^\circ$

## Interfaces

System Manager

PLC Blocks

The slave axis couples on the fly as quickly as possible to velocity.

## 10 Flying Saw with Online Conversion to Master PTP Axis

This versatile slave type is started as a flying saw. It is coupled at the synchronisation point to the velocity of the master axis. It can then be locally synchronised again by means of position compensation in the synchronous phase, after which it can be uncoupled online in the synchronous phase from the master and can itself be converted into a PTP master axis. At this point force is not applied, after which all the master functions are available.

### Requirements

The logical precondition for this conversion is that (i) the flying saw is started as a parallel slave, and that (ii) the current direction of the flying saw agrees with the logical direction at the start. It is most dynamically favourable if the flying saw is converted during the synchronous phase.

The dynamic precondition for a conversion is that the new dynamic parameters are such that the new master axis can be stopped at any time. One implication of this is that the new dynamic parameters (those of the new master axis) must not be weaker than the old dynamic parameters (in the synchronous phase, these are those of the old master axis for the flying saw, otherwise those of the old master axis of the flying saw) multiplied by the coupling factor.

The physical precondition for a smooth dynamic conversion is that the velocity of the flying saw does not fluctuate too much at the time of the conversion.

### Condition Following Conversion

Following the conversion, the flying saw is a PTP master axis with full functionality. The instruction given to the axis at that time implies that, following the conversion, the axis will initially operate without force (that is with an acceleration of 0), and that it then retains the velocity that has been reached in that way and drives to the target position  $\pm\infty$ , whereby the direction is maintained at the time of the transformation.

### Functionality Following Conversion

All master functions are available: override, stop, position compensation, 'new end position' and 'new velocity and new end position'.

Table 3: Start parameters

Start parameters	Meaning and boundary conditions
Master synchronisation position $P_{ms}$	Position at which the master should be located (in its co-ordinate system) when the slave is to be a synchronised to it, $ P_{ms} - P_{mcd}  = 2 \times  P_{ss} - P_{sc} $
Slave synchronisation position $P_{ss}$	Position at which the slave should be located (in its co-ordinate system) when it is to be a synchronised to the master, $ P_{ms} - P_{mcd}  = 2 \times  P_{ss} - P_{sc} $
Diagonal Angle $\theta$	Parallel slave: $\theta = 90.0^\circ$
Coupling factor $cf$	Coupling factor $cf \neq 0.0$

### Interfaces

System Manager

PLC Blocks

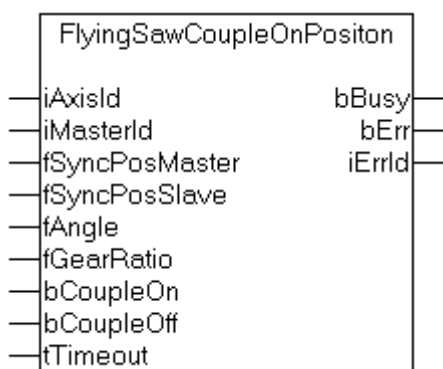
# 11 TcNcFlyingSaw

In many plants workpieces undergo machining operations while being transported. For this purpose it is necessary to synchronize the position and the speed of tool and workpiece, so that the tool can then be applied as if to a stationary workpiece. One example of such an application is a saw that during the transport process cuts through the material that is being transported (flying saw). In order to implement this kind of application, TwinCAT provides the flying saw.

The TwinCAT PLC library **TcNcFlyingSaw.lib**, available as an *additional product*, provides easy management of the flying saw. An example program using the flying saw makes use of this library.

## 11.1 Classical Flying Saw

### 11.1.1 FlyingSawCoupleOnPositon



**FlyingSawCoupleOnPositon** couples a slave axis to a master axis as a flying saw. The synchronisation speed is achieved precisely at the synchronisation point.

#### Interface

```

VAR_INPUT
    iAxisId      : UINT; (* Flying saw axis ID *)
    iMasterId    : UINT;
    fSyncPosMaster : LREAL; (* absolute master synchronous position *)
    fSyncPosSlave  : LREAL; (* absolute slave synchronous position *)
    fAngle       : LREAL; (* parallel = 90 degrees *)
    fGearRatio   : LREAL; (* 0.0 is invalid, then default is 1.0 *)
    bCoupleOn    : BOOL; (* couple *)
    bCoupleOff   : BOOL; (* decouple *)
    tTimeout     : TIME;
END_VAR
VAR_OUTPUT
    bBusy      : BOOL;
    bErr       : BOOL;
    iErrId     : UDINT;
END_VAR

```

**iAxisId**: Axis ID of the slave axis that is to be coupled as a flying saw

**iMasterId**: Axis ID of the master axis that is to be coupled to.

**fSyncPosMaster** : Synchronization position of the master axis

**fSyncPosSlave** : Synchronization position of the slave axis

**fAngle** : Angle of the flying saw (this is 90° for a parallel axis)

**fGearRatio** : Gearing factor. A gearing factor of 0 is invalid, and will be interpreted as 1

**bCoupleOn**: Edge-triggered signal for execution of the axis coupling.

**bCoupleOff:** Edge-triggered signal for uncoupling the axes.

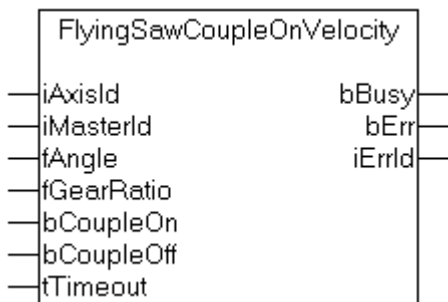
**tTimeout:** ADS timeout (about 1 second).

**bBusy:** Becomes TRUE with a rising edge at bCoupleOn or bCoupleOff, and remains TRUE until the block has executed the command.

**bErr:** Becomes TRUE if an error occurs while executing the command.

**bErrId:** Error number (ADS or NC error number).

## 11.1.2 FlyingSawCoupleOnVelocity



**FlyingSawCoupleOnVelocity** couples a slave axis to a master axis as a flying saw. The synchronisation speed is reached as quickly as possible.

### Interface

```

VAR_INPUT
  iAxisId      : UINT; (* Flying saw axis ID *)
  iMasterId    : UINT;
  fAngle       : LREAL; (* parallel = 90 degrees *)
  fGearRatio   : LREAL; (* 0.0 is invalid, then default is 1.0 *)
  bCoupleOn    : BOOL; (* couple *)
  bCoupleOff   : BOOL; (* decouple *)
  tTimeout     : TIME;
END_VAR
VAR_OUTPUT
  bBusy        : BOOL;
  bErr         : BOOL;
  iErrId       : UDINT;
END_VAR

```

**iAxisId:** Axis ID of the slave axis that is to be coupled as a flying saw

**iMasterId:** Axis ID of the master axis that is to be coupled to.

**fAngle :** Angle of the flying saw (this is 90° for a parallel axis)

**fGearRatio :** Gearing factor. A gearing factor of 0 is invalid, and will be interpreted as 1

**bCoupleOn:** Edge-triggered signal for execution of the axis coupling.

**bCoupleOff:** Edge-triggered signal for uncoupling the axes.

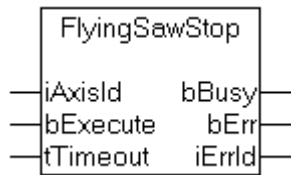
**tTimeout:** ADS timeout (about 1 second).

**bBusy:** Becomes TRUE with a rising edge at bCoupleOn or bCoupleOff, and remains TRUE until the block has executed the command.

**bErr:** Becomes TRUE if an error occurs while executing the command.

**bErrId:** Error number (ADS or NC error number).

### 11.1.3 FlyingSawStop



**FlyingSawStop** stops a slave that has been coupled as a flying saw. The stop is carried out in the coupled state, so it cannot therefore be initiated with the AXACT function block.

#### Interface

```

VAR_INPUT
    iAxisId    : UINT; (* Flying saw axis ID *)
    bExecute   : BOOL; (* Stop Flying Saw *)
    tTimeout   : TIME;
END_VAR
VAR_OUTPUT
    bBusy      : BOOL;
    bErr       : BOOL;
    iErrId     : UDINT;
END_VAR
    
```

**iAxisId:** Axis ID of the slave axis that is coupled as a flying saw

**iMasterId:** Axis ID of the master axis

**bExecute:** Edge-triggered signal for execution of the command.

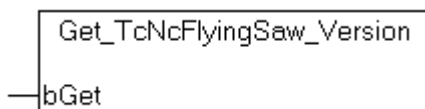
**tTimeout:** ADS timeout (about 1 second).

**bBusy:** Becomes TRUE with a rising edge at bExecute and remains TRUE until the block has executed the command.

**bErr:** Becomes TRUE if an error occurs while executing the command.

**bErrId:** Error number (ADS or NC error number).

### 11.1.4 Get\_TcNcFlyingSaw\_Version



**Get\_TcNcFlyingSaw\_Version** determines the version number of the PLC library TcNcFlyingSaw.lib. The function returns the version number in a string.

#### Interface

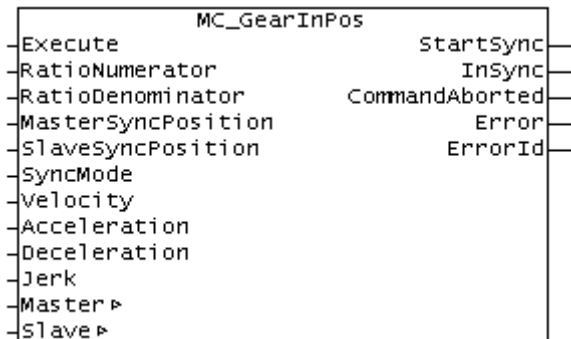
```

FUNCTION Get_TcNcFlyingSaw_Version: STRING(20)
VAR_INPUT
    bGet : BOOL;
END_VAR
    
```

**bGet:** Signal for execution of the command.

## 11.2 PLCopen Function blocks

### 11.2.1 MC\_GearInPos



**MC\_GearInPos** couples a slave axis to a master axis as a flying saw. The synchronisation velocity is achieved exactly at the synchronous position of the master and slave.

The interface to and function of this block is specified by the PLCopen. The block can therefore be seen as a supplement to the PLCopen Motion Control Library.

#### VAR\_INPUT

```
VAR_INPUT
  Execute           : BOOL;
  RatioNumerator    : LREAL := 1;
  RatioDenominator  : UINT := 1;
  MasterSyncPosition : LREAL; (* absolute master synchronous position *)
  SlaveSyncPosition : LREAL; (* absolute slave synchronous position *)
  SyncMode          : ST_SyncMode;
  Velocity          : LREAL := 0;
  Acceleration      : LREAL; (* 0 : use default value *)
  Deceleration      : LREAL; (* 0 : use default value *)
  Jerk              : LREAL; (* 0 : use default value *)
END_VAR
```

<b>Execute :</b>	Synchronisation begins with a rising edge.
<b>RatioNumerator :</b>	Numerator for the gearing factor (see <a href="#">note [► 24]</a> )
<b>RatioDenominator :</b>	Denominator for the gearing factor (see <a href="#">note [► 24]</a> )
<b>MasterSyncPosition :</b>	The master's synchronous position.
<b>SlaveSyncPosition :</b>	The slave's synchronous position.
<b>SyncMode :</b>	The boundary conditions for the synchronisation are specified in the <a href="#">SyncMode [► 28]</a> variables.
<b>Velocity :</b>	Maximum slave velocity in the synchronisation phase. If a velocity is not specified, the maximum velocity of the axis from the system manager data is used. <b>Warning:</b> The velocity given here is only checked if this checking is activated through the SyncMode variable (GearInSync_CheckMask_MaxVelo=TRUE).
<b>Acceleration :</b>	Maximum slave acceleration in the synchronisation phase. If an acceleration is not specified, the maximum acceleration of the axis from the system manager data is used. <b>Warning:</b> The acceleration given here is only checked if this checking is activated through the SyncMode variable (GearInSync_CheckMask_MaxAcc=TRUE).



**Deceleration :**

Maximum slave deceleration in the synchronisation phase. If a deceleration is not specified, the maximum deceleration of the axis from the system manager data is used.

**Warning:** The deceleration given here is only checked if this checking is activated through the SyncMode variable (GearInSync\_CheckMask\_MaxDec=TRUE).

**Jerk :**

Maximum slave jerk in the synchronisation phase. If a jerk is not specified, the maximum jerk of the axis from the system manager data is used.

**Warning:** The jerk given here is only checked if this checking is activated through the SyncMode variable (GearInSync\_CheckMask\_MaxJerk=TRUE).



**Negative values**

If a gearing factor of 1:3 is wanted, the numerator is set to 1 and the denominator to 3. The numerator can also be assigned negative values.

The slave axis can be decoupled using the MC\_GearOut function block from the TcMC Motion Control Library. If the slave is decoupled while it is moving, then it retains its velocity and can be halted using MC\_Stop.

**VAR\_OUTPUT**

```
VAR_OUTPUT
  StartSync      : BOOL;
  InSync         : BOOL;
  CommandAborted : BOOL;
  Error          : BOOL;
  ErrorId        : UDINT;
END_VAR
```

**StartSync :**

Becomes TRUE as soon as synchronization starts.

**InSync :**

Becomes TRUE as soon as the slave has synchronized with the master.

**CommandAborted :**

Becomes TRUE, if the coupling could not be carried out.

**Error :**

Becomes TRUE, as soon as an error occurs.

**ErrorID :**

Supplies the error number

**VAR\_IN\_OUT**

```
VAR_IN_OUT
  Master : NCTOPLC_AXLESTRUCT;
  Slave  : NCTOPLC_AXLESTRUCT;
END_VAR
```

**Master :**

Axis structure of the master.

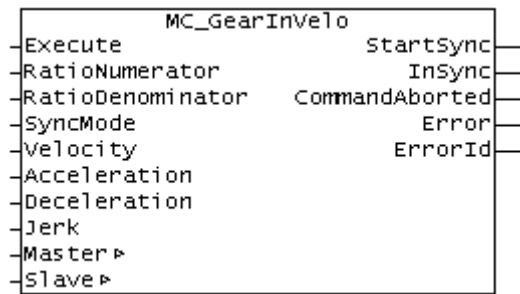
**Slave :**

Axis structure of the slave.

**Requirements**

Development environment	Target system type	PLC libraries to be linked
TwinCAT v2.9	PC (i386)	TcNcFlyingSaw.Lib

## 11.2.2 MC\_GearInVelo



**MC\_GearInVelo** couples a slave axis to a master axis as a flying saw.

The interface to and function of this block is specified by the PLCopen. The block can therefore be seen as a supplement to the [PLCopen Motion Control Library](#).

### VAR\_INPUT

```
VAR_INPUT
  Execute      : BOOL;
  RatioNumerator : LREAL := 1;
  RatioDenominator : UINT := 1;
  SyncMode     : ST_SyncMode;
  Velocity     : LREAL := 0;
  Acceleration : LREAL; (* 0 : use default value *)
  Deceleration : LREAL; (* 0 : use default value *)
  Jerk        : LREAL; (* 0 : use default value *)
END_VAR
```

<b>Execute:</b>	The rising edge starts the master-slave coupling as a flying saw.
<b>RatioNumerator :</b>	Numerator for the gearing factor. (See <a href="#">note [▶ 26]</a> )
<b>RatioDenominator :</b>	Denominator for the gearing factor. (See <a href="#">note [▶ 26]</a> )
<b>SyncMode :</b>	The boundary conditions for the synchronisation are specified in the <a href="#">SyncMode [▶ 28]</a> variables.
<b>Velocity :</b>	Maximum slave velocity in the synchronisation phase. If a velocity is not specified, the maximum velocity of the axis from the system manager data is used. <b>Warning:</b> The velocity given here is only checked if this checking is activated through the SyncMode variable.
<b>Acceleration :</b>	Maximum slave acceleration in the synchronisation phase. If an acceleration is not specified, the maximum acceleration of the axis from the system manager data is used. <b>Warning:</b> The acceleration given here is only checked if this checking is activated through the SyncMode variable.
<b>Deceleration :</b>	Maximum slave deceleration in the synchronisation phase. If a deceleration is not specified, the maximum deceleration of the axis from the system manager data is used. <b>Warning:</b> The deceleration given here is only checked if this checking is activated through the SyncMode variable.
<b>Jerk :</b>	Maximum slave jerk in the synchronisation phase. If a jerk is not specified, the maximum jerk of the axis from the system manager data is used. <b>Warning:</b> The jerk given here is only checked if this checking is activated through the SyncMode variable.

**i Negative values**

If a gearing factor of 1:3 is wanted, the numerator is set to 1 and the denominator to 3. The numerator can also be assigned negative values.

The slave axis can be decoupled using the MC\_GearOut function block from the TcMC Motion Control Library. If the slave is decoupled while it is moving, then it retains its velocity and can be halted using MC\_Stop.

**VAR\_OUTPUT**

```
VAR_OUTPUT
  StartSync      : BOOL;
  InSync         : BOOL;
  CommandAborted : BOOL;
  Error          : BOOL;
  ErrorId        : UDINT;
END_VAR
```

- StartSync :** Becomes TRUE as soon as synchronisation starts.
- InSync :** Becomes TRUE as soon as the slave has synchronised with the master.
- CommandAborted :** Becomes TRUE, if the coupling could not be carried out.
- Error :** Becomes TRUE, as soon as an error occurs.
- ErrorID :** Supplies the error number

**VAR\_IN\_OUT**

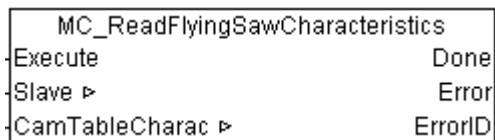
```
VAR_IN_OUT
  Master : NCTOPLC_AXLESTRUCT;
  Slave  : NCTOPLC_AXLESTRUCT;
END_VAR
```

- Master :** Axis structure of the master.
- Slave :** Axis structure of the slave.

**Requirements**

Development environment	Target system type	PLC libraries to be linked
TwinCAT v2.9	PC (i386)	TcNcFlyingSaw.Lib

**11.2.3 MC\_ReadFlyingSawCharacteristics**



The **MC\_ReadFlyingSawCharacteristics** function allows the characteristic figures for the synchronisation phase of the Universal Flying Saw to be read.

**VAR\_INPUT**

```
VAR_INPUT
  Execute : BOOL;
END_VAR
```

- Execute:** A rising edge initiates reading the characteristic values from the TwinCAT NC.



### Information availability

The data calculated is not available until the Universal Flying Saw starts.

### VAR\_OUTPUT

```
VAR_OUTPUT
  Done      : BOOL;
  Error     : BOOL;
  ErrorID   : UDINT;
END_VAR
```

**Done :** Is set to TRUE when the data record has been successfully read.

**Error :** Becomes TRUE, as soon as an error occurs.

**ErrorID :** Supplies the error number

### VAR\_IN\_OUT

```
VAR_IN_OUT
  Slave           : NCTOPLC_AXLESTRUCT;
  CamTableCharac : MC_FlyingSawCharacValues;
END_VAR
```

**Slave :** Axis structure of the slave.

**CamTableCharac: :** Structure containing the characteristic values

### Requirements

Development environment	Target system type	PLC libraries to be linked
TwinCAT v2.9	PC (i386)	TcNcFlyingSaw.Lib

## 11.2.4 The ST\_SyncMode data structure

```
TYPE ST_SyncMode :
STRUCT
  (* mode *)
  GearInSyncMode           : ST_GearInSyncMode;

  (* 32 bit check mask ... *)
  GearInSync_CheckMask_MinPos      : BOOL;
  GearInSync_CheckMask_MaxPos      : BOOL;
  GearInSync_CheckMask_MaxVelo     : BOOL;
  GearInSync_CheckMask_MaxAcc      : BOOL;
  GearInSync_CheckMask_MaxDec      : BOOL;
  GearInSync_CheckMask_MaxJerk     : BOOL;
  GearInSync_CheckMask_OvershootPos : BOOL;
  GearInSync_CheckMask_UndershootPos : BOOL;
  GearInSync_CheckMask_OvershootVelo : BOOL;
  GearInSync_CheckMask_UndershootVelo : BOOL;
  GearInSync_CheckMask_OvershootVeloZero : BOOL;
  GearInSync_CheckMask_UndershootVeloZero : BOOL;

  (* operation masks ... *)
  GearInSync_OpMask_RollbackLock    : BOOL;
  GearInSync_OpMask_InstantStopOnRollback : BOOL;
  GearInSync_OpMask_PreferConstVelo  : BOOL;
  GearInSync_OpMask_IgnoreMasterAcc  : BOOL;
  GearInSync_OpMask_IgnoreSlaveAcc   : BOOL;
  GearInSync_OpMask_DetailedErrorCodes : BOOL;
END_STRUCT
END_TYPE
```

```
TYPE ST_GearInSyncMode :
(
  GEARINSYNCMODE_POSITIONBASED, (* synchronization based on the master position, slave dynamics
depend on master dynamics *)

  GEARINSYNCMODE_TIMEBASED (* synchronization based on a standalone slave PTP profile, master
```

```
independet slave dynamics *)  
);  
END_TYPE
```



the time based synchronization profile (GEARINSMODE\_TIMEBASED) is actually only implemented in the MC\_GearInVelo function block.

Operation of the individual bits.



More Information:  
**[www.beckhoff.com](http://www.beckhoff.com)**

Beckhoff Automation GmbH & Co. KG  
Hülshorstweg 20  
33415 Verl  
Germany  
Phone: +49 5246 9630  
[info@beckhoff.com](mailto:info@beckhoff.com)  
[www.beckhoff.com](http://www.beckhoff.com)

