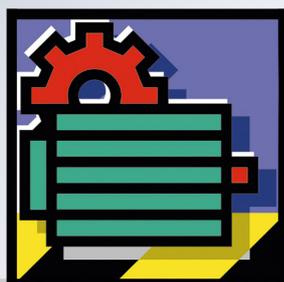


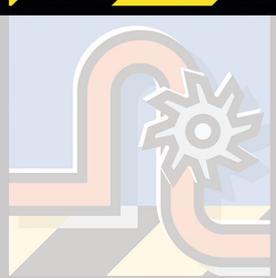
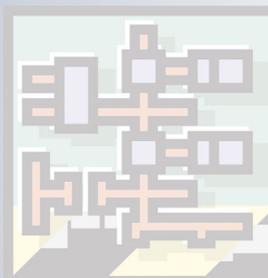
Handbuch | DE

TS5055

TwinCAT 2 | NC Flying Saw



Supplement | Motion



Inhaltsverzeichnis

1	Vorwort	5
1.1	Hinweise zur Dokumentation	5
1.2	Sicherheitshinweise	6
1.3	Hinweise zur Informationssicherheit	7
2	Allgemein	8
3	Synchronisation auf Geschwindigkeit	11
4	Synchronisation auf Position	15
5	Parametrierbare Randbedingungen, Festlegung des Operation-Modes	18
6	Charakteristische Kennwerte	21
7	Berechnung der Synchronisierungsphase	23
8	Bewegungsumkehr der Masterachse / Rücklaufsperr	25
9	Diagonalsäge	29
10	Interfaces	30
11	Bedienung aus dem Systemmanager	32
12	PLC-API	33
12.1	TcMC2_FlyingSaw	33
12.1.1	MC_GearInVelo	33
12.1.2	MC_GearInPos	36
12.1.3	MC_ReadFlyingSawCharacteristics.....	39
12.2	TcMC2.....	40
12.2.1	MC_GearOut.....	40
12.3	Datentypen	41
12.3.1	ST_SyncMode.....	41
12.3.2	MC_FlyingSawCharacValues	42
12.4	Beispielprogramm	44
12.4.1	Beispielprogramm Fliegende Säge	44
13	Fehlersituationen und Fehlercodes:	45

1 Vorwort

1.1 Hinweise zur Dokumentation

Diese Beschreibung wendet sich ausschließlich an ausgebildetes Fachpersonal der Steuerungs- und Automatisierungstechnik, das mit den geltenden nationalen Normen vertraut ist.

Zur Installation und Inbetriebnahme der Komponenten ist die Beachtung der Dokumentation und der nachfolgenden Hinweise und Erklärungen unbedingt notwendig.

Das Fachpersonal ist verpflichtet, für jede Installation und Inbetriebnahme die zu dem betreffenden Zeitpunkt veröffentlichte Dokumentation zu verwenden.

Das Fachpersonal hat sicherzustellen, dass die Anwendung bzw. der Einsatz der beschriebenen Produkte alle Sicherheitsanforderungen, einschließlich sämtlicher anwendbaren Gesetze, Vorschriften, Bestimmungen und Normen erfüllt.

Disclaimer

Diese Dokumentation wurde sorgfältig erstellt. Die beschriebenen Produkte werden jedoch ständig weiter entwickelt.

Wir behalten uns das Recht vor, die Dokumentation jederzeit und ohne Ankündigung zu überarbeiten und zu ändern.

Aus den Angaben, Abbildungen und Beschreibungen in dieser Dokumentation können keine Ansprüche auf Änderung bereits gelieferter Produkte geltend gemacht werden.

Marken

Beckhoff®, TwinCAT®, TwinCAT/BSD®, TC/BSD®, EtherCAT®, EtherCAT G®, EtherCAT G10®, EtherCAT P®, Safety over EtherCAT®, TwinSAFE®, XFC®, XTS® und XPlanar® sind eingetragene und lizenzierte Marken der Beckhoff Automation GmbH.

Die Verwendung anderer in dieser Dokumentation enthaltenen Marken oder Kennzeichen durch Dritte kann zu einer Verletzung von Rechten der Inhaber der entsprechenden Bezeichnungen führen.

Patente

Die EtherCAT-Technologie ist patentrechtlich geschützt, insbesondere durch folgende Anmeldungen und Patente:

EP1590927, EP1789857, EP1456722, EP2137893, DE102015105702

mit den entsprechenden Anmeldungen und Eintragungen in verschiedenen anderen Ländern.

EtherCAT®

EtherCAT® ist eine eingetragene Marke und patentierte Technologie lizenziert durch die Beckhoff Automation GmbH, Deutschland

Copyright

© Beckhoff Automation GmbH & Co. KG, Deutschland.

Weitergabe sowie Vervielfältigung dieses Dokuments, Verwertung und Mitteilung seines Inhalts sind verboten, soweit nicht ausdrücklich gestattet.

Zu widerhandlungen verpflichten zu Schadenersatz. Alle Rechte für den Fall der Patent-, Gebrauchsmuster- oder Geschmacksmustereintragung vorbehalten.

1.2 Sicherheitshinweise

Sicherheitsbestimmungen

Beachten Sie die folgenden Sicherheitshinweise und Erklärungen!
Produktspezifische Sicherheitshinweise finden Sie auf den folgenden Seiten oder in den Bereichen Montage, Verdrahtung, Inbetriebnahme usw.

Haftungsausschluss

Die gesamten Komponenten werden je nach Anwendungsbestimmungen in bestimmten Hard- und Software-Konfigurationen ausgeliefert. Änderungen der Hard- oder Software-Konfiguration, die über die dokumentierten Möglichkeiten hinausgehen, sind unzulässig und bewirken den Haftungsausschluss der Beckhoff Automation GmbH & Co. KG.

Qualifikation des Personals

Diese Beschreibung wendet sich ausschließlich an ausgebildetes Fachpersonal der Steuerungs-, Automatisierungs- und Antriebstechnik, das mit den geltenden Normen vertraut ist.

Erklärung der Symbole

In der vorliegenden Dokumentation werden die folgenden Symbole mit einem nebenstehenden Sicherheitshinweis oder Hinweistext verwendet. Die Sicherheitshinweise sind aufmerksam zu lesen und unbedingt zu befolgen!

GEFAHR

Akute Verletzungsgefahr!

Wenn der Sicherheitshinweis neben diesem Symbol nicht beachtet wird, besteht unmittelbare Gefahr für Leben und Gesundheit von Personen!

WARNUNG

Verletzungsgefahr!

Wenn der Sicherheitshinweis neben diesem Symbol nicht beachtet wird, besteht Gefahr für Leben und Gesundheit von Personen!

VORSICHT

Schädigung von Personen!

Wenn der Sicherheitshinweis neben diesem Symbol nicht beachtet wird, können Personen geschädigt werden!

HINWEIS

Schädigung von Umwelt oder Geräten

Wenn der Hinweis neben diesem Symbol nicht beachtet wird, können Umwelt oder Geräte geschädigt werden.

Tipp oder Fingerzeig

i Dieses Symbol kennzeichnet Informationen, die zum besseren Verständnis beitragen.

1.3 Hinweise zur Informationssicherheit

Die Produkte der Beckhoff Automation GmbH & Co. KG (Beckhoff) sind, sofern sie online zu erreichen sind, mit Security-Funktionen ausgestattet, die den sicheren Betrieb von Anlagen, Systemen, Maschinen und Netzwerken unterstützen. Trotz der Security-Funktionen sind die Erstellung, Implementierung und ständige Aktualisierung eines ganzheitlichen Security-Konzepts für den Betrieb notwendig, um die jeweilige Anlage, das System, die Maschine und die Netzwerke gegen Cyber-Bedrohungen zu schützen. Die von Beckhoff verkauften Produkte bilden dabei nur einen Teil des gesamtheitlichen Security-Konzepts. Der Kunde ist dafür verantwortlich, dass unbefugte Zugriffe durch Dritte auf seine Anlagen, Systeme, Maschinen und Netzwerke verhindert werden. Letztere sollten nur mit dem Unternehmensnetzwerk oder dem Internet verbunden werden, wenn entsprechende Schutzmaßnahmen eingerichtet wurden.

Zusätzlich sollten die Empfehlungen von Beckhoff zu entsprechenden Schutzmaßnahmen beachtet werden. Weiterführende Informationen über Informationssicherheit und Industrial Security finden Sie in unserem <https://www.beckhoff.de/secguide>.

Die Produkte und Lösungen von Beckhoff werden ständig weiterentwickelt. Dies betrifft auch die Security-Funktionen. Aufgrund der stetigen Weiterentwicklung empfiehlt Beckhoff ausdrücklich, die Produkte ständig auf dem aktuellen Stand zu halten und nach Bereitstellung von Updates diese auf die Produkte aufzuspielen. Die Verwendung veralteter oder nicht mehr unterstützter Produktversionen kann das Risiko von Cyber-Bedrohungen erhöhen.

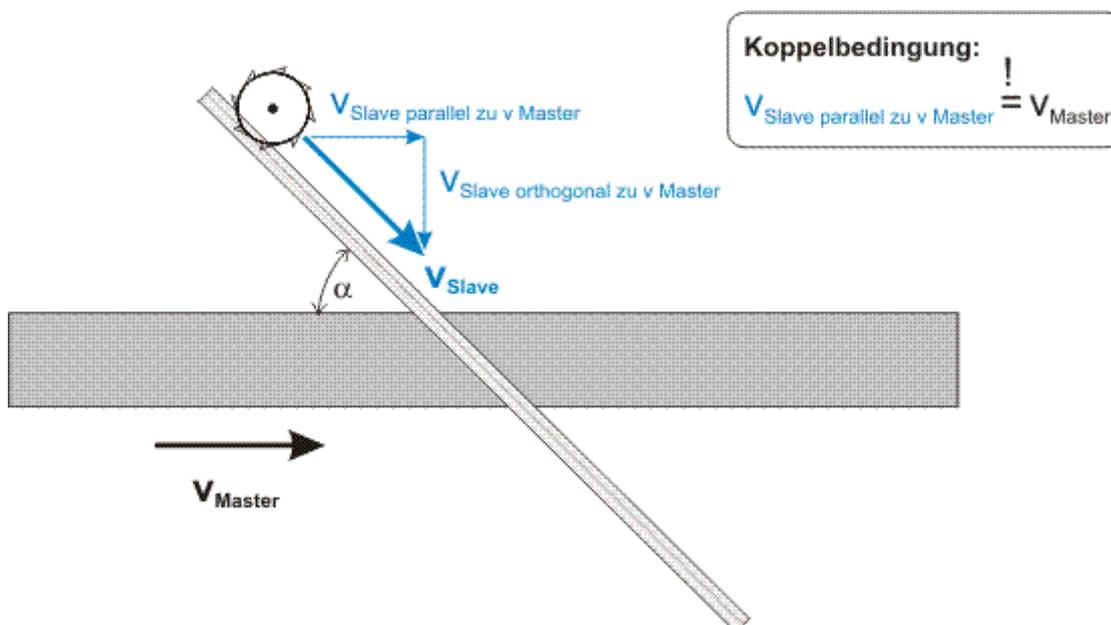
Um stets über Hinweise zur Informationssicherheit zu Produkten von Beckhoff informiert zu sein, abonnieren Sie den RSS Feed unter <https://www.beckhoff.de/secinfo>.

2 Allgemein

Die Fliegende Säge ist eine Slave-Achse mit der Funktionalität, auf eine fahrende Masterachse aufzusynchronisieren. Für einen Bearbeitungsvorgang fährt die Slave-Achse synchron zu der Masterachse. Durch diese synchrone Bewegung zur Masterachse wird die Bearbeitung eines Werkstücks auch während des Transports möglich.

Ein wesentlicher Unterschied zwischen der "Fliegenden Säge" und der "Universellen Fliegenden Säge" besteht in dem, für die Synchronisation geforderten Anfangszustand der Slave-Achse. Bei der "Universellen Fliegenden Säge" ist der Start der Synchronisation im Gegensatz zur "Fliegenden Säge" auch dann möglich, wenn die Slave-Achse bereits gestartet ist und somit nicht im Stillstand ist. Weiterhin werden bei der "Universellen Fliegenden Säge" verbesserte Sollwertprofile berechnet, die vom Benutzer über verschiedenste Randbedingungen beeinflusst werden können.

Das Verhältnis der Mastergeschwindigkeit zur Slave-Geschwindigkeit in der Synchronphase wird mit einem variablen Koppelfaktor parametrisiert. Dieser Koppelfaktor wird beispielsweise bei einer Diagonalsäge ungleich 1 gewählt, damit die Geschwindigkeitskomponente der Slave-Achse in Richtung der Masterachse ($v_{\text{Slave parallel zu } v_{\text{Master}}}$) in der Synchronphase gleich der Mastergeschwindigkeit (v_{Master}) ist. (siehe Bild)



Grundsätzlich stellt die Universelle Fliegende Säge zwei verschiedene Synchronisierungsmöglichkeiten zur Verfügung. Bei der *Synchronisation auf Geschwindigkeit* wird der Slave unter Berücksichtigung des Koppelfaktors schnellstmöglich auf den Master synchronisiert. Die Koppelpositionen der Master- und Slave-Achse ergeben sich hierbei durch die Zielsetzung der schnellstmöglichen Synchronisation. Im Gegensatz hierzu werden die Koppelpositionen der Master- und Slave-Achse bei der *Synchronisation auf Position* von dem Benutzer parametrisiert. Die Master- und Slave-Achse bewegen sich hierbei also spätestens ab den vorgegebenen Positionen synchron.

Bei beiden beschriebenen Synchronisierungsmöglichkeiten besteht die Möglichkeit, verschiedene Randbedingungen für die Synchronisierungsphase vorzugeben. Mit diesen Randbedingungen kann der Synchronisierungsvorgang an die Anforderungen der Maschine angepasst werden.



In dieser Dokumentation ist die Universelle Fliegende Säge *TcMc2_FlyingSaw.lib* beschrieben. Diese steht ab TwinCAT 2.9 Build 248 zur Verfügung. Wenn Sie die Vorgängerversion *TcNcFlyingSaw.lib* nutzen, finden Sie [hier](#) weitere Informationen.

Interfaces

Die Bedienung und Überwachung der Universellen Fliegende Säge erfolgt aus der SPS mit entsprechenden Funktionsbausteinen. Zu Inbetriebnahmezwecken kann die Universelle Fliegende Säge aber auch direkt aus dem [TwinCAT-System Manager](#) [► 32] gestartet werden. Als unterlagerte Schnittstellen wird hierbei das zyklische NC-SPS-Achs-Interface sowie die ADS-Kommunikation verwendet.

Synchronisation auf Geschwindigkeit

Bei der [Synchronisation auf Geschwindigkeit](#) [► 11] wird die Slave-Achse mit den angegebenen Dynamikparametern schnellst möglich auf die Master-Achse synchronisiert. In der Synchronphase ist die Slave-Geschwindigkeit proportional zu der Mastergeschwindigkeit, so dass gilt:

$$v_{Slave} = F_{CouplingFactor} \cdot v_{Master}$$

Ablauf der Synchronisierung

Die Synchronisation der Slave-Achse mit der Master-Achse erfolgt nach dem folgenden Ablauf:

1. Start der Universellen Fliegenden Säge. Dieses entspricht dem logischen Ankoppeln an die Master-Achse. Dieser Zeitpunkt wird als Koppelzeitpunkt bezeichnet.
2. Synchronisierungsphase: Der Slave wird aus seinem Anfangszustand auf die Mastergeschwindigkeit beschleunigt, wobei die vom Benutzer angegebenen Randbedingungen für die Slavebewegung eingehalten werden. Der Zeitpunkt, bei dem von der Synchronisierungsphase in die Synchronphase gewechselt wird, wird als Synchronisierungszeitpunkt bezeichnet.
3. Synchronphase: Der Slave bewegt sich synchron zu dem Master.
4. Abkoppeln der Universellen Fliegenden Säge. Dieses entspricht einer Onlinewandlung und der gekoppelte Slave wird wieder zum eigenständigen Master, der sich mit der aus der Onlinewandlung ergebenden Geschwindigkeit endlos weiter bewegt.
5. Beispielsweise Nachstarten oder Stopp des ehemaligen Slaves. Es steht wieder die volle Funktionalität einer TwinCAT NC Master-Achse zur Verfügung.

Synchronisation auf Position

Bei der [Synchronisation auf Position](#) [► 15] wird die Slave-Achse mit den angegebenen Dynamikparametern an der angegebenen Synchronposition auf den Master synchronisiert. Das heißt, die Slave-Achse erreicht exakt in den Synchronpositionen von Master und Slave die Synchrongeschwindigkeit und bewegt sich ab dieser Position synchron zu dieser. Für die Slave-Geschwindigkeit gilt in der Synchronphase:

$$v_{Slave} = F_{CouplingFactor} \cdot v_{Master}$$

Ablauf der Synchronisierung

Die Synchronisation der Slave-Achse mit der Master-Achse erfolgt nach dem folgenden Ablauf:

1. Start der Universellen Fliegenden Säge, dieses entspricht dem logischen Ankoppeln an die Master-Achse. Dieser Zeitpunkt wird als Koppelzeitpunkt bezeichnet.
2. Synchronisierungsphase: Der Slave wird aus seinem Anfangszustand auf die Mastergeschwindigkeit beschleunigt und erreicht exakt in der vorgegebenen Synchronposition des Masters die Slave-Synchronposition und die Synchrongeschwindigkeit., wobei die vom Benutzer angegebenen Randbedingungen der Slavebewegung eingehalten werden. Der Zeitpunkt, bei dem von der Synchronisierungsphase in die Synchronphase gewechselt wird, wird als Synchronisierungszeitpunkt bezeichnet.
3. Synchronphase: Der Slave bewegt sich synchron zu dem Master.

4. Abkoppeln der Universellen Fliegenden Säge. Dieses entspricht einer Onlinewandlung und der gekoppelte Slave wird wieder zum eigenständigen Master, der sich mit der aus der Onlinewandlung ergebenden Geschwindigkeit endlos weiter bewegt.
5. Beispielsweise Nachstarten oder Stopp des ehemaligen Slaves. Es steht wieder die volle Funktionalität einer TwinCAT NC Master-Achse zur Verfügung.

Parametrierbare Randbedingungen der Synchronisierung

Grundsätzlich dürfen für die Berechnung des Synchronisierungsprofils die Master- und Slave-Achse beliebige Anfangsbedingungen besitzen.

Der Übergang der Slavebewegung aus dem Ausgangszustand in den Synchronzustand wird so berechnet, dass die vom Benutzer parametrierbaren Randbedingungen [► 18] der Slavebewegung eingehalten werden. Mit diesen Randbedingungen kann beispielsweise die maximale Slave-Geschwindigkeit limitiert, oder ein Überschwingen in der Position verhindert werden.

Die Berechnung und Überprüfung der parametrierbaren Randbedingungen erfolgt auf Basis der ermittelten charakteristischen Kennwerte der Synchronisierungsphase. Bei der Ermittlung der *charakteristischen Kennwerte* wird idealisiert vorausgesetzt, dass sich die Master-Achse ab dem Koppelzeitpunkt beschleunigungsfrei mit konstanter Geschwindigkeit weiterbewegt. Somit ist die exakte Berechnung und Überprüfung der parametrisierten Randbedingungen nur unter dieser Voraussetzung möglich. Eine andere sinnvolle Annahme bezüglich der zukünftigen Masterbewegung ist nicht möglich, da zum Koppelzeitpunkt die zukünftige Masterbewegung nicht bekannt ist.

Eine mögliche zukünftige Masterbeschleunigung wirkt sich durch die Kopplung auch auf die Slave-Dynamik aus. Eine solche Masterbeschleunigung führt dazu, dass die berechneten und überprüften Kennwerte teilweise in Abhängigkeit der Masterbeschleunigung über- oder unterschritten werden. Der tabellarischen Beschreibung der charakteristischen Kennwerte kann entnommen werden, welche charakteristischen Kennwerte [► 21] von einer Masterbeschleunigung beeinflusst werden.

Charakteristische Kennwerte der Slavebewegung

Nach dem Start der Universellen Fliegenden Säge stehen dem Benutzer die charakteristischen Kennwerte [► 21] der Slavebewegung, die dieser in der Synchronisierungsphase durchläuft, zur Verfügung. Diese Kennwertstruktur enthält Größen wie die maximale Slave-Beschleunigung, die mini- und maximale Slave-Position usw. Diese Werte werden unter der Annahme eines unbeschleunigten Masters berechnet und sind somit teilweise nur für einen solchen exakt richtig.

HINWEIS

Die Masterbeschleunigung zum Startzeitpunkt der "Universellen Fliegenden Säge" geht maßgeblich in die Profilberechnung und dessen Optimierungen ein. D.h. für eine Encoderachse als Master müssen die Geschwindigkeit und die Beschleunigung sorgfältig gefiltert, oder sogar die Berechnung der Istbeschleunigung abgewählt werden (siehe. „Encoder Mode“).

3 Synchronisation auf Geschwindigkeit

Bei der Synchronisation auf Geschwindigkeit wird eine Slave-Achse schnellst möglich aus einem beliebigen Anfangszustand auf eine fahrende Masterachse synchronisiert, wobei sich die Synchrongeschwindigkeit des Slaves aus der Mastergeschwindigkeit multipliziert mit dem Koppelfaktor ergibt.

Die Synchronisationsphase der Slave-Achse wird so berechnet, dass die vom Benutzer parametrisierten [Randbedingungen](#) [► 18] eingehalten werden. Die Berechnung und Überprüfung dieser Randbedingungen erfolgt unter der Annahme, dass sich die Master-Achse ab dem Koppelzeitpunkt unbeschleunigt bewegt. Bei einer nicht kräftefreien Master-Achse können die parametrisierten Randbedingung teilweise über- oder unterschritten werden.

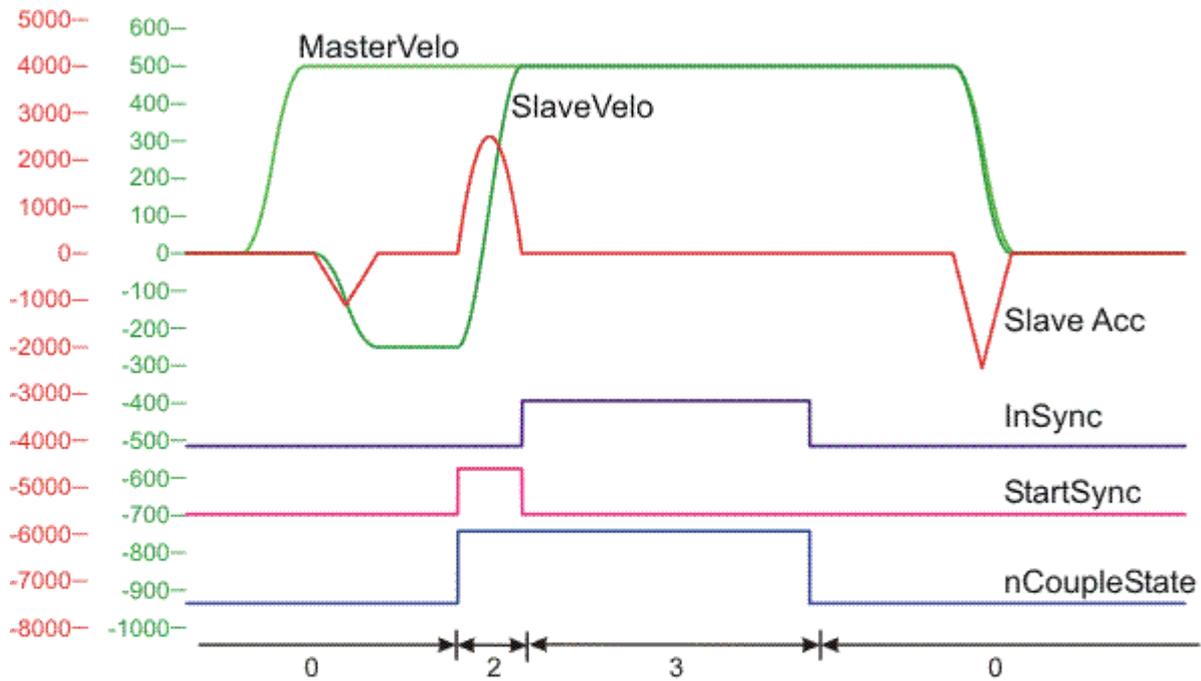
Beispiel 1:

In der im Bild dargestellten Phase 0 (links) bewegen sich die beiden Achsen vollkommen unabhängig voneinander. Die zukünftige Master-Achse wird auf 500 mm/s beschleunigt und die zukünftige Slave-Achse wird auf -250 mm/s beschleunigt. In dem Phasenwechsel von 0 auf 2 wird die Synchronisation auf Geschwindigkeit der Universellen Fliegende Säge gestartet (StartSync = 1) und das Sollwertprofil zur Synchronisation berechnet. Dieses berechnete Slave-Sollwertprofil wird in der Phase 2 der Slave-Achse vorgegeben. Am Ende der Phase 2 erreicht die Slave-Achse exakt die Synchrongeschwindigkeit, welche der Geschwindigkeit der Master-Achse multipliziert mit dem Koppelfaktor entspricht. Ab diesem Zeitpunkt befinden sich die Achsen in der Synchronphase (Phase 3). In dieser Synchronphase bewegen sich beide Achsen unter Berücksichtigung des Koppelfaktors synchron. Die Synchronphase wird durch das Abkoppelkommando beendet. Dieses entspricht in dem Bild dem Übergang von der Phase 3 in die Phase 0. Ab diesem Zeitpunkt handelt es sich wieder um 2 unabhängige Masterachsen. Zum Zeitpunkt der Abkopplung erfolgt eine Onlinewandlung der Slave-Achse zu einer Master-Achse. Bei dieser Onlinewandlung wird eine mögliche Beschleunigung oder Verzögerung des Slaves abgebaut und hierdurch die Geschwindigkeit des ehemaligen Slaves bestimmt, mit der sich dieser endlos weiter bewegt.

Die aktuelle Phase der Slave-Achse kann der Variablen `nAxisState` im zyklischen Achsinterfaces entnommen werden. (Die Bezeichnung der einzelnen Phasen in den Bildern stimmt nicht mit den Werten der Variablen `nAxisState` [► 30] überein.)

```
(* Startparameter *)
fMasterVelo    := 500;
fSlaveVelo     := -250;

(* Koppelparameter *)
fGearRatio     := 1.0;
fSlaveAcc      := 2500;
fSlaveDec      := 2500;
fSlaveJerk     := 5000;
```



Phase 0: unabhängige Masterachsen

Phase 2: Synchronisation

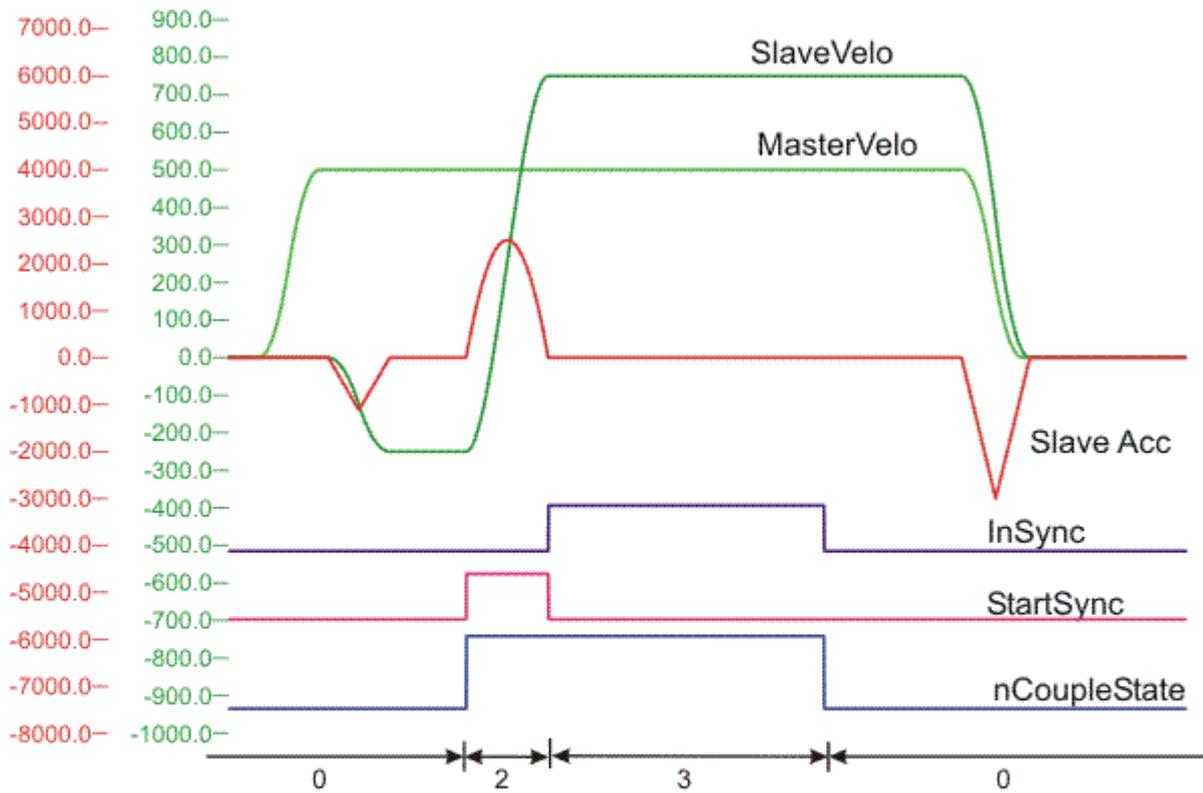
Phase 3: Synchronphase

Beispiel 2:

Wie Beispiel 1, nur mit dem Koppelfaktor 1.5.

```
(* Startparameter *)
fMasterVelo := 500;
fSlaveVelo  := -250;

(* Koppelparameter *)
fGearRatio  := 1.5;
fSlaveAcc   := 2500;
fSlaveDec   := 2500;
fSlaveJerk  := 5000;
```



Phase 0: unabhängige Masterachsen

Phase 2: Synchronisation

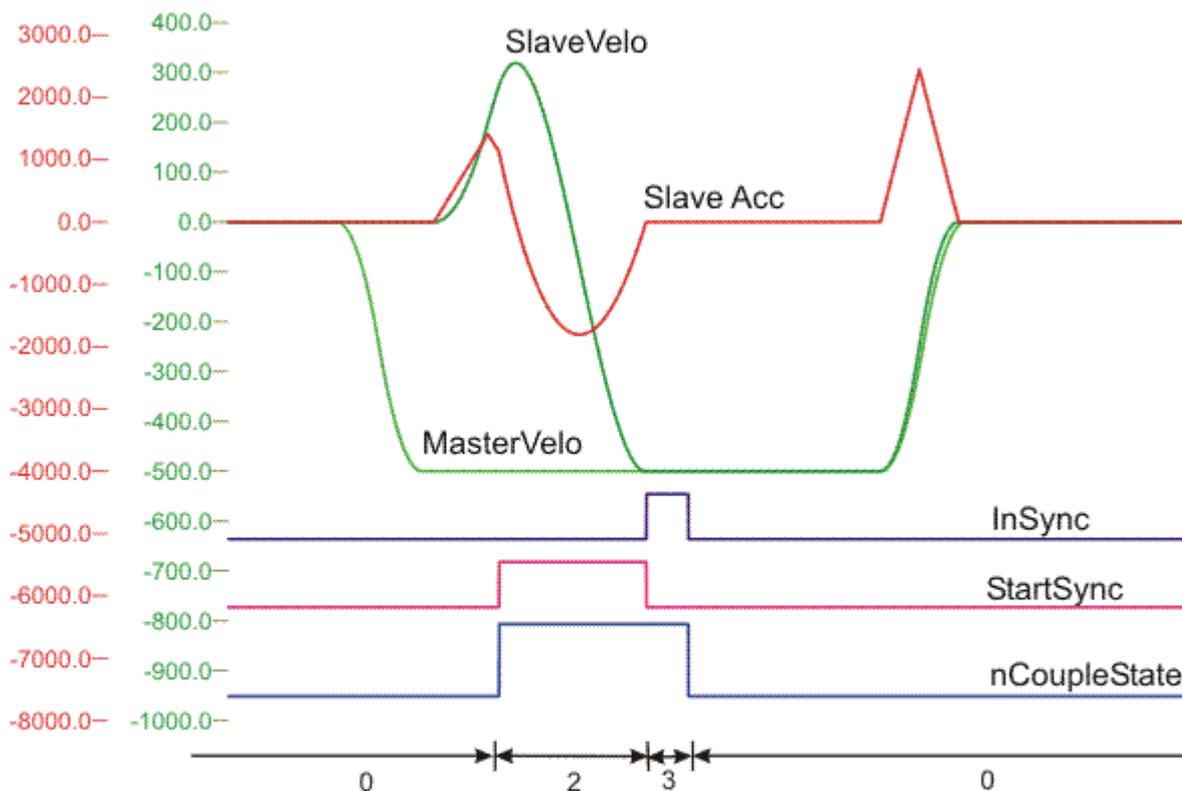
Phase 3: Synchronphase

Beispiel 3:

Start der Kopplung in der Beschleunigungsphase des zukünftigen Slaves.

```
(* Startparameter *)
fMasterVelo := -500;
fSlaveVelo  := 400;

(* Koppelparameter *)
fGearRatio := 1;
fSlaveAcc  := 2500;
fSlaveDec  := 2500;
fSlaveJerk := 5000;
```



Phase 0: unabhängige Masterachsen

Phase 2: Synchronisation

Phase 3: Synchronphase

SPS Funktionsbausteine

Die Kopplung erfolgt mit dem Funktionsbaustein MC_GearInVelo. Zum Beenden der Synchronphase, also zum Entkoppeln (Onlinewandlung), wird der Funktionsbaustein MC_GearOut verwendet.

4 Synchronisation auf Position

Bei der Synchronisation auf Position wird die Slave Achse aus einem Anfangszustand so auf die Masterachse synchronisiert, dass sie die geforderte Synchrongeschwindigkeit ($v_{\text{Master}} = \text{Koppelfaktor}$) und Synchronposition exakt in der Synchronposition des Masters erreicht.

Die Synchronisierungsphase der Slave-Achse wird so berechnet, dass die vom Benutzer parametrisierten [Randbedingungen](#) [► 18] eingehalten werden. Die Berechnung und Überprüfung dieser Randbedingungen erfolgt unter der Annahme, dass sich die Master-Achse ab dem Koppelzeitpunkt unbeschleunigt bewegt. Bei einer nicht kräftefreien Master-Achse können die parametrisierten Randbedingung teilweise über- oder unterschritten werden.

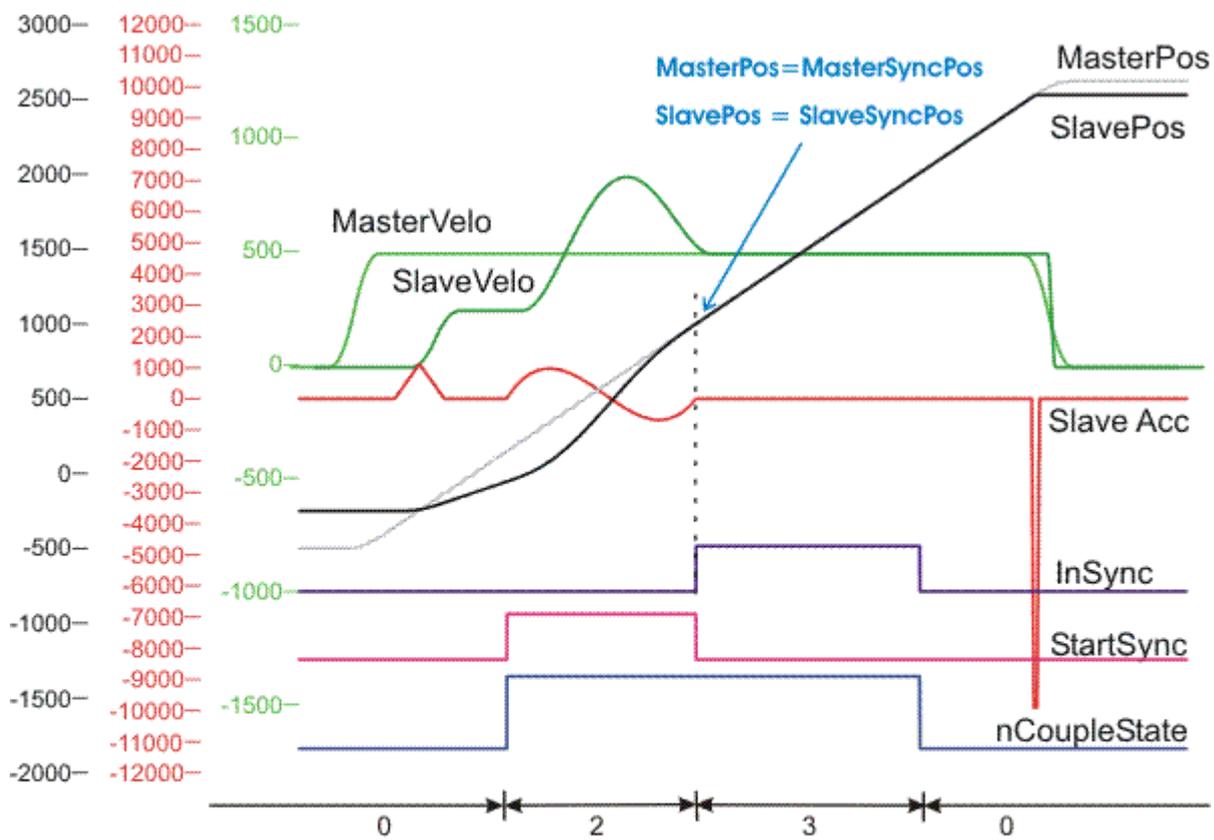
Beispiel 1:

In dem dargestellten Beispiel werden die zukünftige Master- und Slave-Achse unabhängig voneinander in positive Richtung gestartet. Zum Zeitpunkt des dargestellten Phasenwechsels von 0 auf 2 wird die Universelle Fliegende Säge mit einer Synchronisation auf Position gestartet (StartSync = 1). Eine Synchronisation auf Position bedeutet, dass die Slave-Achse exakt an der Master-Synchronposition (Master-Sync-Position) die Synchrongeschwindigkeit erreicht und sich an der Slave-Synchronposition (Slave-Sync-Position) befindet. Die Synchrongeschwindigkeit entspricht der Mastergeschwindigkeit multipliziert mit dem gewählten Koppelfaktor. In dem dargestellten Beispiel ist die Master-Sync-Position identisch mit der Slave-Sync-Position, so dass die Master- und Slave-Position bei der positiven Flanke des *InSync*-Signals identisch sind. Der Koppelfaktor in diesem Beispiel ist zu 1 gewählt worden, so dass die Master- und Slave-Geschwindigkeit in der Synchronphase (3) identisch sind. Zum Zeitpunkt des Phasenwechsels von 3 auf 0 wird die Slave-Achse von der Master-Achse abgekoppelt (Onlinewandlung) und bewegt sich ab diesem Zeitpunkt wieder als unabhängige Masterachse. Zum Zeitpunkt der Abkopplung erfolgt eine Onlinewandlung der Slave-Achse zu einer Master-Achse. Bei dieser Onlinewandlung wird eine mögliche Beschleunigung oder Verzögerung des Slaves abgebaut und somit die Geschwindigkeit des ehemaligen Slaves bestimmt, mit der sich dieser endlos weiter bewegt.

Die aktuelle Phase der Slave-Achse kann der Variablen *nAxisState* im zyklischen Achsinterfaces entnommen werden. (Die Bezeichnung der einzelnen Phasen in den Bildern stimmt nicht mit den Werten der Variablen *nAxisState* [► 30] überein.)

```
(* Startparameter *)
fMasterVelo      := 500;
fSlaveVelo       := 250;
fMasterStartPos  := -500;
fSlaveStartPos   := -250;

(* Koppelparameter *)
fGearRatio       := 1;
fMasterSynchronPos := 1000;
fSlaveSynchronPos := 1000;
fSlaveAcc        := 2500;
fSlaveDec        := 2500;
fSlaveJerk       := 5000;
```



Phase 0: unabhängige Masterachsen

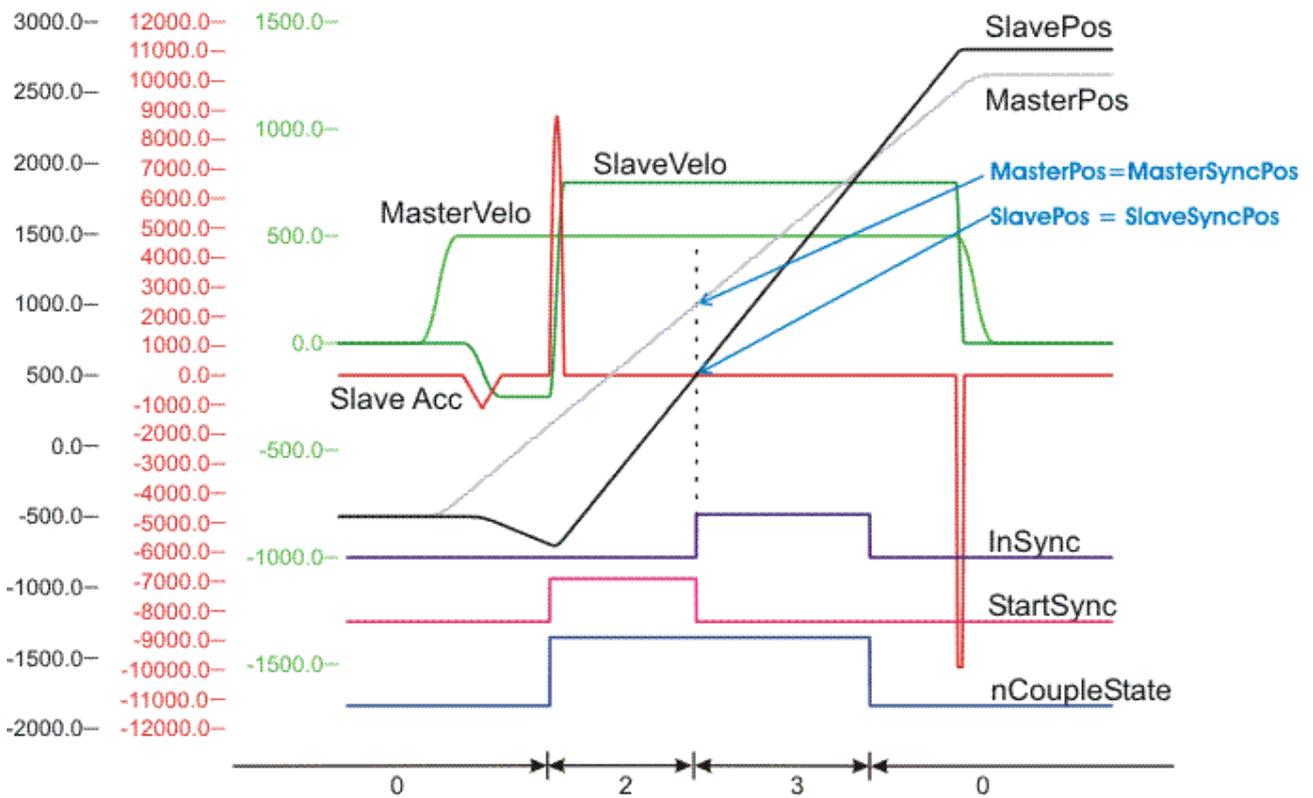
Phase 2: Synchronisation

Phase 3: Synchronphase

Beispiel 2:

```
(* Startparameter *)
fMasterVelo      := 500;
fSlaveVelo      := -250;
fMasterStartPos := -500;
fSlaveStartPos  := -500;

(* Koppelparameter *)
fGearRatio      := 1.5;
fMasterSynchronPos := 1000;
fSlaveSynchronPos  := 500;
fSlaveAcc       := 10000;
fSlaveDec       := 10000;
fSlaveJerk      := 50000;
```



Phase 0: unabhängige Masterachsen

Phase 2: Synchronisation

Phase 3: Synchronphase

SPS Funktionsbausteine

Die Kopplung erfolgt mit dem Funktionsbaustein MC_GearInPos. Zum Beenden der Synchronphase, also zum Abkoppeln (Onlinewandlung des Slaves in einen eigenständigen Master), wird der Funktionsbaustein MC_GearOut verwendet.

5 Parametrierbare Randbedingungen, Festlegung des Operation-Modus

Bei der Universellen Fliegenden Säge ist es möglich, verschiedenste Randbedingungen für die Slavebewegung in der Synchronisierungsphase anzugeben. Mit diesen Randbedingungen ist es möglich, für die in der folgenden Tabelle aufgeführten Slavegrößen Grenzwerte vorzugeben. Eine Überprüfung, ob die einzelnen Grenzwerte eingehalten werden, kann mit der Bitmaske *SyncMode* aktiviert werden. Mit den Randbedingungen, die für die Synchronisierungsphase angegeben werden, wird auch das Sollwertprofil für die Synchronisierung beeinflusst. Ob und wie die einzelnen Bedingungen das Profil beeinflussen kann dem Diagramm [▶ 23](#) des folgenden Kapitels entnommen werden.

Synchronisierungs-Modus			Beschreibung	
Define: GealnSyncMode				
GEARINSYNCMODE_POSITIONBASED			In diesem Mode der Universellen Fliegenden Säge wird zur Synchronisierung der Slaveachse auf die Masterachse ein, von der Masterposition abhängiges Profil generiert.	
GEARINSYNCMODE_TIMEBASED			In diesem Mode der Universellen Fliegenden Säge wird zur Synchronisierung der Slaveachse auf die Masterachse ein zeitabhängiges Bewegungsprofil generiert, welches die Einhaltung aller dynamischen Grenzwerte der Slaveachse sicherstellt. Dieser Mode ist zur Zeit nur bei einer Kopplung auf Geschwindigkeit verfügbar.	
Bitmasken für den "SyncMode"			Beschreibung	Randbedingung
Define: GEARINSYNC_CH ECKMASK_	Wert Dezimal	Wert Hexadezimal		
MINPOS		1 0x0000 0001	Prüft, ob die Software Endlage Min (Maschinendaten) der Slave-Achse unterschritten wird.	$pos_{Slave} \geq pos_{SlaveMin}$
MAXPOS		2 0x0000 0002	Prüft, ob die Software Endlage Max (Maschinendaten) der Slave-Achse überschritten wird.	$pos_{Slave} \leq pos_{SlaveMax}$
MAXVELO		4 0x0000 0004	Prüft, ob die maximal erlaubte Slavegeschwindigkeit (Maschinendaten) überschritten wird.	$ v_{Slave} \leq v_{SlaveMax}$
MAXACC		8 0x0000 0008	Prüft, ob die maximale Slavebeschleunigung (Maschinendaten) überschritten wird	$acc_{Slave} \leq acc_{SlaveMax}$

MAXDEC	16	0x0000 0010	Prüft, ob die maximale Slave-Verzögerung (Maschinendaten) überschritten wird	$dec_{Slave} \leq dec_{SlaveMax}$
MAXJERK	32	0x0000 0020	Prüft, ob der maximale Slave-Ruck (Maschinendaten) überschritten wird.	$j_{Slave} \leq j_{SlaveMax}$
OVERSHOOTPOS	256	0x0000 0100	Prüft ein Überschwingen der Slaveposition.	
UNDERSHOOTPOS	512	0x0000 0200	Prüft ein Unterschwingen der Slaveposition.	
OVERSHOOTVELO	1024	0x0000 0400	Prüft ein Überschwingen der Slavegeschwindigkeit.	
UNDERSHOOTVELO	2048	0x0000 0800	Prüft ein Unterschwingen der Slavegeschwindigkeit.	
OVERSHOOTVELOZERO	4096	0x0000 1000	Prüft ein Überschreiten der Slavegeschwindigkeit über 0.0.	
UNDERSHOOTVELOZERO	8192	0x0000 2000	Prüft ein Unterschreiten der Slavegeschwindigkeit unter 0.0	
Bitmasken für Operationsmodi			Beschreibung	
GEARINSYNC_OPMASK_				
ROLLBACKLOCK	65536	0x0001 0000	Bit = 0: (default)	
			Wenn der Slave die Synchronphase erreicht hat, wird die synchrone Kopplung für alle folgenden Masterbewegungen beibehalten, bis die Kopplung aufgehoben wird. Dieses gilt somit auch für eine Bewegungsumkehr des Masters, wenn der Master rückwärts über die Koppelposition hinaus fährt. (Weitere Erläuterungen) [► 25]	
			Bit = 1:	
			Das Setzen dieses Bits aktiviert die Rücklaufsperrung, die ein Stillsetzen des Slaves bewirkt, wenn der Master nach einer Bewegungsumkehr rückwärts über die Koppelposition hinaus fährt. (Weitere Erläuterungen) [► 25]	

INSTANTSTOPON ROLLBACK	131072	0x0002 0000	<p>Bit = 0: (default) Bei Erreichen der Koppelposition wird die Slavegeschwindigkeit mit einem Polynom 5. Ordnung stetig abgebaut, wobei dieses Polynom auf ein möglichst schnelles Stillsetzen optimiert ist. (Weitere Erläuterungen) [▶ 25]</p> <p>Bit = 1: Bei Erreichen der Koppelposition wird der Slave innerhalb eines NC-Ticks sollwertseitig stillgesetzt. Die Slavegeschwindigkeit wird auf 0.0 gesetzt und die Position beibehalten. Dieses abrupte Stillsetzen kann zum Ansprechen der Schleppabstandsüberwachung führen! (Weitere Erläuterungen) [▶ 25]</p>
PREFERCONSTV ELO	1048576	0x0010 0000	<p>Bit = 0: (default) Standardeinstellung</p> <p>Bit = 1: Es wird versucht anstatt nur eines Polynoms 5ter Ordnung eine Phase mit konstanter Geschwindigkeit zu verwenden. Dadurch kann sich eine Kombination aus einem Polynom 5ter Ordnung, einer Synchronphase und einem weiteren Polynom 5ter Ordnung ergeben (P5-P1-P5), (siehe weitere Erläuterungen) [▶ 25]. Dabei wird die maximal gegebene Beschleunigung und Verzögerung verwendet. Zur Kontrolle und Beschränkung des Rucks wird empfohlen in der Bitmaske MAXJERK zu setzen.</p>
IGNOREMASTER ACC	2097152	0x0020 0000	<p>Bit = 0: (default) Standardeinstellung</p> <p>Bit = 1: Bei der Berechnung der Kopplung wird die Beschleunigung des Masters ignoriert, dass heißt auf Null gesetzt. Dadurch werden die internen Optimierungen verwendet. Bei moderaten Beschleunigung führt diese Vorgabe zu tolerierbaren Schleppabständen. Nach dem Abbau des Schleppabstandes ist die relative Positionsgenauigkeit von dieser Einstellung unabhängig.</p>

HINWEIS

Die zuvor aufgeführten Checks beziehen sich nur auf die Synchronisierungsphase (GEARINSYNC-STATE_SYNCHRONIZING) und nicht auf die Synchronfahrphase. Ferner sind diese Berechnungen und Checks nur unter der Annahme möglich, dass sich der Master ab dem Koppelzeitpunkt weiter mit konstanter Geschwindigkeit bewegt, also beschleunigungsfrei ist. Andere Annahmen für den Master zu treffen ergibt keinen Sinn, da zum Koppelzeitpunkt allgemein nicht bekannt ist, wie sich der Master in der Zukunft bewegen wird.

6 Charakteristische Kennwerte

Nach dem erfolgreichen Start der Universellen Fliegenden Säge können mit dem SPS-Funktionsbaustein MC_ReadFlyingSawCharacteristics die charakteristischen Kennwerte des für die **Synchronisierungsphase** berechneten Slave-Sollwertprofils ausgelesen werden. Diese Kennwerte beziehen sich also nicht auf Synchron- oder Vorphase und gelten nur für die Synchronisierungsphase!

Die einzelnen charakteristischen Kennwerte sind in der unten dargestellten Tabelle beschrieben.

HINWEIS

Bei der Berechnung der dynamischen charakteristischen Kennwerte wird davon ausgegangen, dass sich der Master ab dem Start der Universellen Fliegenden Säge (ab dem Koppelzeitpunkt) unbeschleunigt weiterbewegt. Folglich sind die berechneten Kennwerte nur für einen unbeschleunigten Master exakt richtig und werden gegebenenfalls teilweise über- oder unterschritten, wenn der Master beschleunigt oder verzögert wird.

Struktur der charakteristischen Kennwerte

Die charakteristischen Kennwerte werden in der umfangreichen Struktur MC_FlyingSawCharacValues abgelegt, in der die folgenden Größen enthalten sind:

- i** Der in der Tabelle beschriebene Startzeitpunkt bezieht sich immer auf den Start der Synchronisierungsphase und der Endzeitpunkt auf das Ende der Synchronisierungsphase (nicht auf das Ende der Synchronphase)! Es steht hier also immer die Synchronisierungsphase im Mittelpunkt.
- In der 3. Spalte der Tabelle wird angegeben, ob eine mögliche Masterbeschleunigung den charakteristischen Kennwert beeinflusst oder nicht.

Bezeichnung	Beschreibung	Unabhängig von einer Masterbeschleunigung
fMasterVeloNom	Master-Geschwindigkeit im Startzeitpunkt der Universellen Fliegenden Säge	nein
fMasterPosStart	Masterposition im Startzeitpunkt der Universellen Fliegenden Säge	ja
fSlavePosStart	Slaveposition im Startzeitpunkt der Universellen Fliegenden Säge	ja
fSlaveVeloStart	Slave-Geschwindigkeit im Startzeitpunkt der Universellen Fliegenden Säge	nein
fSlaveAccStart	Slave-Beschleunigung im Startzeitpunkt der Universellen Fliegenden Säge	nein
fSlaveJerkStart	Ruck des Slaves im Startzeitpunkt der Universellen Fliegenden Säge	nein
fMasterPosEnd	Masterposition am Ende der Synchronisierungsphase	ja
fSlavePosEnd	Slaveposition am Ende der Synchronisierungsphase	ja
fSlaveVeloEnd	Slave-Geschwindigkeit am Ende der Synchronisierungsphase	nein
fSlaveAccEnd	Slave-Beschleunigung am Ende der Synchronisierungsphase	nein
fSlaveJerkEnd	Ruck des Slaves am Ende der Synchronisierungsphase	nein
fMPosAtSPosMin	Masterposition zum Zeitpunkt der minimalen Slaveposition	nein

Bezeichnung	Beschreibung	Unabhängig von einer Masterbeschleunigung
fSlavePosMin	Minimale Slaveposition	ja
fMPosAtSVeloMin	Masterposition zum Zeitpunkt der minimalen Slave-Geschwindigkeit	nein
fSlaveVeloMin	Minimale Slave-Geschwindigkeit	nein
fMPosAtSAccMin	Masterposition zum Zeitpunkt der minimalen Slave-Beschleunigung	nein
fSlaveAccMin	Minimale Slave-Beschleunigung	nein
fSVeloAtSAccMin	Slave-Geschwindigkeit zum Zeitpunkt der minimalen Slave-Beschleunigung	nein
fSlaveJerkMin	Minimaler Slave Ruck	nein
fSlaveDynMomMin	Minimales dynamisches Moment des Slaves (NOT SUPPORTED YET !)	nein
fMPosAtSPosMax	Masterposition zum Zeitpunkt der maximalen Slaveposition	nein
fSlavePosMax	Maximale Slave Position	ja
fMPosAtSVeloMax	Masterposition zum Zeitpunkt der maximalen Slavegeschwindigkeit	nein
fSlaveVeloMax	Maximale Slave-Geschwindigkeit	nein
fMPosAtSAccMax	Masterposition zum Zeitpunkt der maximalen Slave-Beschleunigung	nein
fSlaveAccMax	Maximale Slave-Beschleunigung	nein
fSVeloAtSAccMax	Slave-Geschwindigkeit zum Zeitpunkt der maximalen Slave-Beschleunigung	nein
fSlaveJerkMax	Maximaler Ruck des Slaves	nein
fSlaveDynMomMax	Maximales dynamisches Moment des Slaves (NOT SUPPORTED YET !)	nein
fSlaveVeloMean	Mittlere absolute Slave-Geschwindigkeit (NOT SUPPORTED YET !)	nein
fSlaveAccEff	Effektive Slave-Beschleunigung (NOT SUPPORTED YET !)	nein

7 Berechnung der Synchronisierungsphase

Bei der Berechnung der Synchronisierungsphase wird versucht, unter Einhaltung der vom Benutzer angegebenen Randbedingungen eine optimale Lösung zu finden. Wenn die vorgegebenen Randbedingungen nicht eingehalten werden können, wird die Kopplung mit einem entsprechenden Fehler abgelehnt.

Optimierungen

Wie in dem unten dargestellten Ablaufdiagramm ersichtlich ist, beeinflussen die einzelnen Bitmasken zum Teil die internen Optimierungsschritte der Profilberechnung, da je nach vorgegebenen Regeln (s. [Parametrierbare Randbedingungen \[► 18\]](#)) ein Optimum gesucht wird. Im Wesentlichen wird ein Polynom 5. Ordnung oder eine Kombination aus einem Polynom 5. Ordnung mit einem Polynom 1. Ordnung verwendet. Ein Polynom 5. Ordnung ist im Allgemeinen nicht überschwingungsfrei, die Beschleunigungen sind aber moderater als bei der Kombination aus einem Polynom 5. Ordnung mit einem Polynom 1. Ordnung. Die Kombination der Polynome aus 1. und 5. Ordnung wird so berechnet, dass diese immer überschwingungsfrei ist. Bei ihr treten aber höhere Beschleunigungen und Verzögerungen auf. Falls zum Beispiel die Istgeschwindigkeit mit der Synchrongeschwindigkeit übereinstimmt, aber eine bestimmte Positionsdifferenz ausgeglichen werden muss. Dann wird intern die optimale Geschwindigkeit in Abhängigkeit von der maximalen Beschleunigung errechnet. Ergebnis ist ein Polynom 5ter Potenz ein Polynom 1. Ordnung mit der errechneten Geschwindigkeit und ein Polynom 5ter Potenz. Mindestens eines der beiden Polynome 5ter Potenz nutzt die maximale Beschleunigung aus. Damit es dabei nicht zu extremen Ruckwerten kommt, sollte die Überprüfung des Ruckes eingeschaltet sein.



Die dargestellten Optimierungen können nur dann durchgeführt werden, wenn zum Koppelzeitpunkt sowohl die Master- als auch die Slave-Achse beschleunigungsfrei sind. Bei beschleunigten Achsen wird zur Synchronisation ein Polynom 5. Ordnung verwendet, welches auf die Einhaltung der vorgegebenen Randbedingungen geprüft wird, aber nicht optimiert werden kann.

HINWEIS

Handelt es sich bei der Masterachse um eine Encoderachse („fremdes Encodersystem“), die üblicherweise mathematisch niemals beschleunigungsfrei ist, muss besondere Sorgfalt auf die Filterung der Istbeschleunigung gelegt werden (oder beim Encoder die Ermittlung der Istbeschleunigung abgewählt werden, d. h. zu Null gesetzt werden). Ferner gibt es für diese Kombination (Masterencoderachse mit der Universellen Fließenden Säge als Slave) intern in der NC einen Algorithmus, der Masterbeschleunigungen, die betragsmäßig kleiner als $(2.0 \cdot \text{Skalierungsfaktor} / \text{Zykluszeit}^2)$ sind, im Ankoppelmoment zu Null setzt.

Optimierungsschritt 1:

Zielsetzung: „Unter- bzw. überschwingungsfreies Geschwindigkeitsprofil“

Es wird zunächst versucht ein Profil zu berechnen, dass in der Geschwindigkeit über- bzw. unterschwingungsfrei aufsynchronisiert (Kombination aus Polynom 1ter Ordnung und Polynom 5ter Ordnung bzw. umgekehrt, kurz Polynom1+Polynom5 bzw. Polynom5+Polynom1). Wenn hierbei der Check der Beschleunigungen aktiv ist und mindestens einer dieser Grenzwerte (Acc, Dec) überschritten wird, dann wird ein weiteres allgemein nicht-überschwingungsfreies Profil (Polynom5) berechnet. Wenn auch dieses Profil einen der aktiven Grenzwerte (Acc, Dec) überschreitet, wird letztlich das Aufsynchronisierungskommando mit einem Fehlercode abgelehnt.

Optimierungsschritt 2:

Zielsetzung: „Limitierung auf maximal erlaubte Geschwindigkeit“

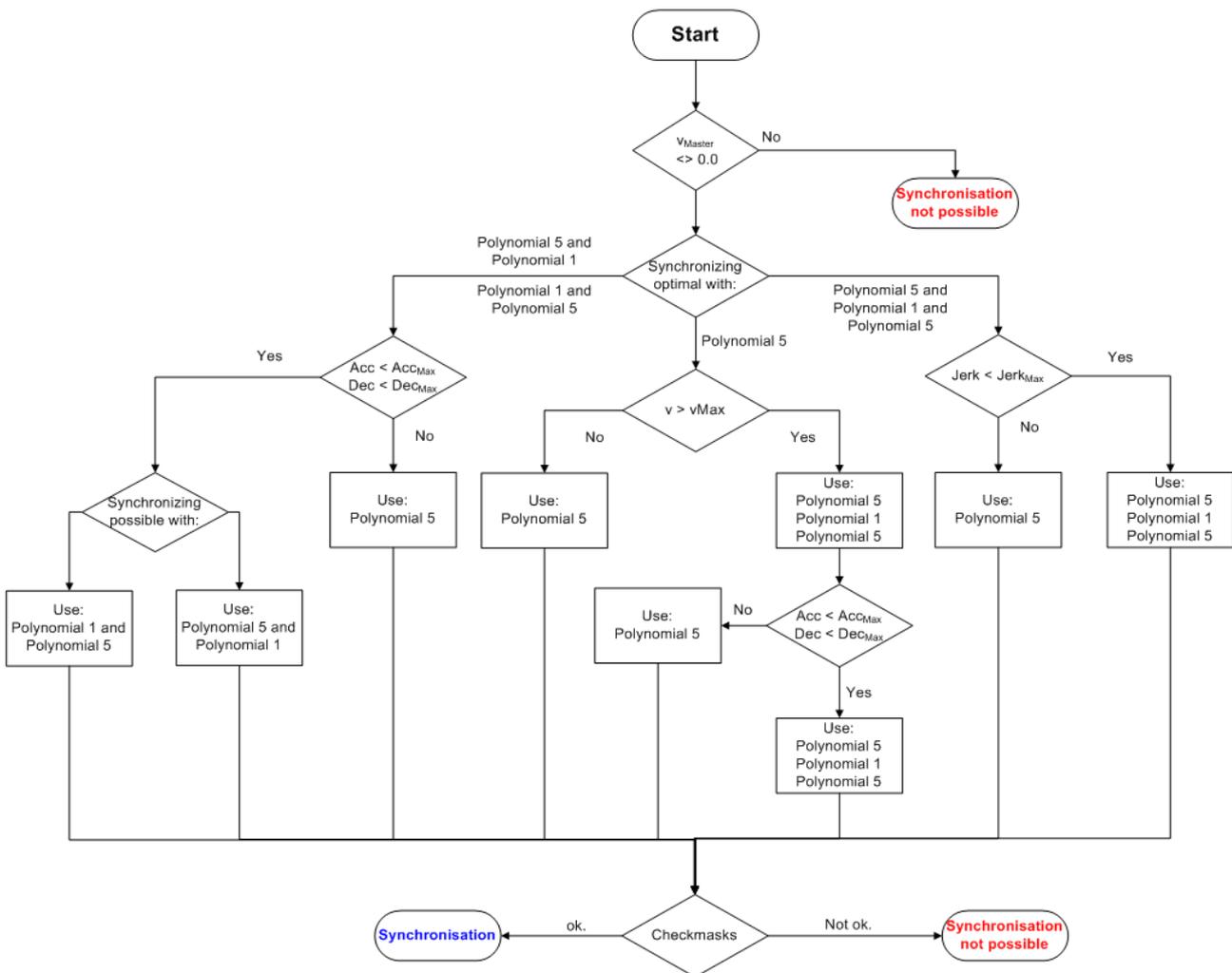
Wenn der Optimierungsschritt 1 nicht möglich gewesen ist, wird im Optimierungsschritt 2 für ein allgemeines Standardprofil (Polynom5) geprüft, ob die maximal erlaubte Geschwindigkeit der Slave-Achse überschritten wird. Wenn dies der Fall ist, wird versucht ein Profil zu erzeugen, welches als maximale Profilgeschwindigkeit genau die maximal erlaubte Geschwindigkeit der Slave-Achse (Maschinendaten) besitzt (Polynom5+Polynom1+Polynom5). Angemerkt sei, dass dieser Optimierungsversuch in der Regel zu größeren Beschleunigungen bzw. Verzögerungen führt.

Wenn nun der Check der Beschleunigungen aktiv ist und mindestens einer der Grenzwerte überschritten

wird, wird dieser Optimierungsschritt 2 verworfen und letztlich ein allgemein nicht-überschwingungsfreies Profil (Polynom5) errechnet. Wenn auch dieses Profil einen der aktiven Grenzwerte (Acc, Dec) überschreitet, wird letztlich das Aufsynchronisierungskommando mit einem Fehlercode abgelehnt.

Ablaufdiagramm der durchgeführten Optimierungen:

Die intern durchgeführten Optimierungen sind in dem folgenden Ablaufdiagramm graphisch dargestellt. Im Wesentlichen wird das Slave-Sollwertprofil als Polynom 5. Ordnung berechnet. Dieses Polynom 5. Ordnung kann mit einem Polynom 1. Ordnung kombiniert werden, um die parametrisierten Randbedingungen einzuhalten. Dem unten dargestellten Ablaufdiagramm kann entnommen werden, wie die einzelnen Randbedingungen die Auswahl der Polynome und somit den Verlauf des Sollwertprofils beeinflussen. Mit der Beschriftung "Polynom n und Polynom m" wird zum Ausdruck gebracht, dass in der Synchronisierungsphase erst das Polynom n und danach das Polynom m zum Einsatz kommt.



8 Bewegungsumkehr der Masterachse / Rücklaufsperr

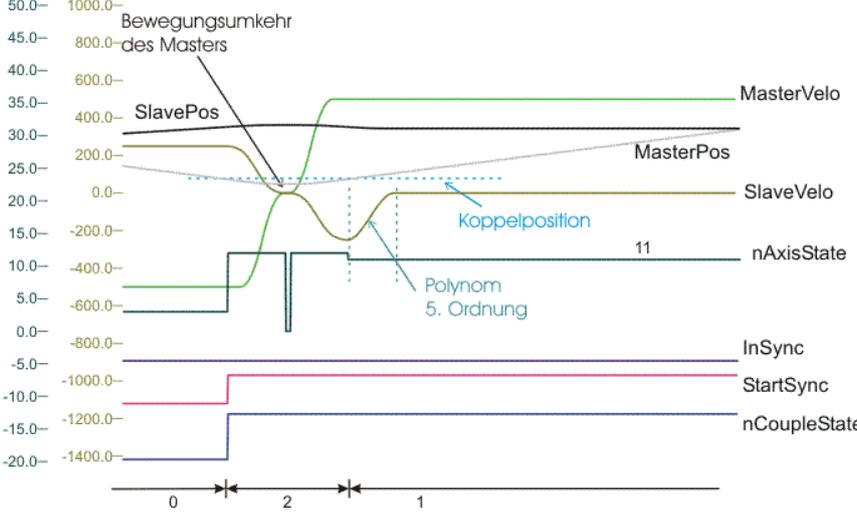
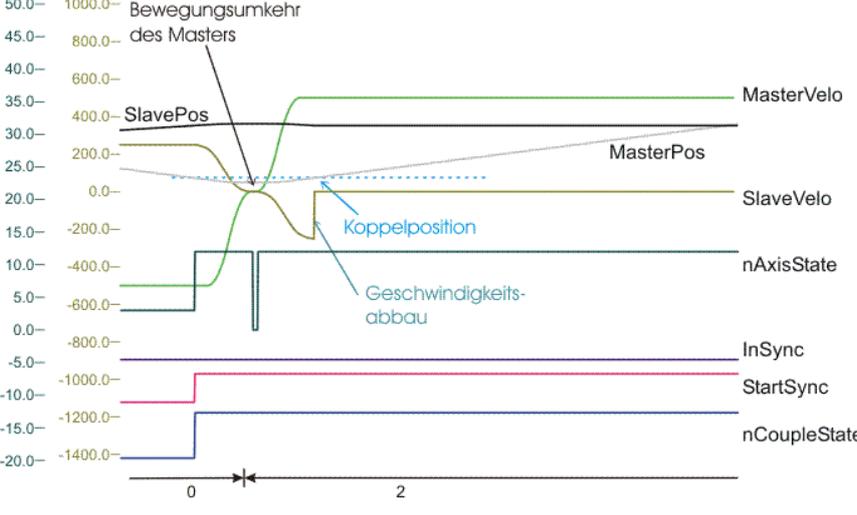
Das Verhalten der Universellen Fliegenden Säge bei einer Bewegungsumkehr des Masters kann über 2 Bits des SyncModes festgelegt werden. Das Bit `GEARINSYNC_OPMASK_ROLLBACKLOCK` aktiviert die Rücklaufsperr, die den Slave stillsetzt, wenn der Master nach einer Bewegungsumkehr rückwärts über die Koppelposition (die Position, an der die Universelle Fliegende Säge gestartet worden ist) hinaus fährt. Mit dem zweiten Bit `GEARINSYNC_OPMASK_INSTANTSTOPONROLLBACK` wird festgelegt, wie der Dynamikabbau der Slave-Achse erfolgt.

HINWEIS

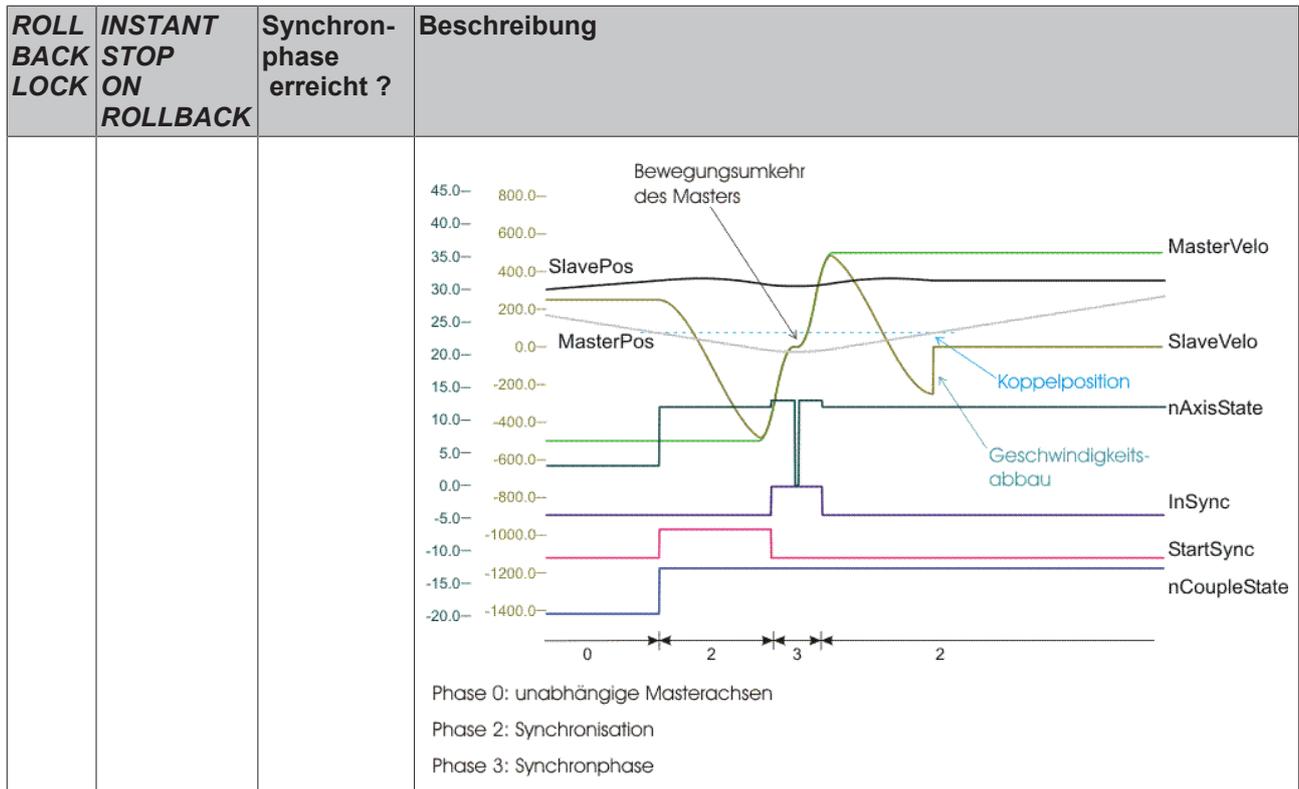
Die Wirkung dieser beiden Bits muss danach unterschieden werden, ob die Synchronphase vor der Bewegungsumkehr erreicht wird, oder ob die Bewegungsumkehr schon in der Synchronisierungsphase erfolgt.

In der folgenden Übersicht wird die Wirkung der Bits `GEARINSYNC_OPMASK_ROLLBACKLOCK` und `GEARINSYNC_OPMASK_INSTANTSTOPONROLLBACK` ausführlich erläutert.

ROLL BACK LOCK	INSTANT STOP ON ROLLBACK	Synchronphase erreicht ?	Beschreibung
0	0	ja	<p>Fall 1:</p> <p>Bei der links angegebenen Bitkombination wird die synchrone Kopplung für alle Masterbewegungen beibehalten, wenn sich die Universelle Fliegende Säge einmal in der Synchronphase befindet. In dem unten dargestellten Bild erfolgt in der Synchronphase eine Bewegungsumkehr der Master-Achse, so dass diese rückwärts über die Koppelposition hinaus fährt. Die synchrone Kopplung wird hierbei beibehalten, so dass sich auch der Slave rückwärts über die Koppelposition hinaus bewegt.</p> <p style="font-size: small;"> Phase 0: unabhängige Masterachsen Phase 2: Synchronisation Phase 3: Synchronphase </p>
0	0	nein	<p>Fall 2:</p> <p>Vor dem Erreichen der Synchronphase erfolgt in der Synchronisierungsphase eine Bewegungsumkehr der Master-Achse, so dass diese rückwärts über die Koppelposition hinaus fährt. Da die Synchronphase nicht erreicht wurde, wird in diesem Fall die Geschwindigkeit der Slave-Achse beim Erreichen der Koppelposition mit einem Polynom 5. Ordnung abgebaut und die Achse stillgesetzt.</p>

ROLL BACK LOCK	INSTANT STOP ON ROLLBACK	Synchron- phase erreicht ?	Beschreibung
			 <p>Phase 0: unabhängige Masterachsen Phase 1: Vorphase Phase 2: Synchronisation</p>
0	1	nein	<p>Fall 3: Wie Fall 2, nur dass die Geschwindigkeit der Slave-Achse beim Erreichen der Koppelposition innerhalb eines Ticks abgebaut wird.</p> <p>Durch den Geschwindigkeitsabbau in einem Tick kann die Schleppabstandsüberwachung der Achse ausgelöst werden.</p>  <p>Phase 0: unabhängige Masterachsen Phase 2: Synchronisation</p>
0	1	ja	<p>Fall 4: Das Verhalten ist identisch mit dem Fall 1!</p>
1	0	nein	<p>Fall 5: Das Verhalten ist identisch mit dem Fall 2!</p>
1	0	ja	<p>Fall 6:</p>

ROLL BACK LOCK	INSTANT STOP ON ROLLBACK	Synchron- phase erreicht ?	Beschreibung
			<p>Mit der links angegebenen Bitkombination wird die Rücklaufsperr der Universellen Fliegende Säge aktiviert. In dem unten dargestellten Bild erfolgt in der Synchronphase eine Bewegungsumkehr der Master-Achse, so dass diese rückwärts über die Koppelposition hinaus fährt. Bei dieser rückwärtigen Bewegung der Masterachse wird ab dem Erreichen der Koppelposition die Geschwindigkeit der Slave-Achse mit einem Polynom 5. Ordnung abgebaut. Es wird also ein Rücklaufen der Slave-Achse insofern verhindert, dass ab dem Erreichen der Koppelpositon die Slave-Geschwindigkeit stetig abgebaut wird.</p> <p>Phase 0: unabhängige Masterachsen Phase 1: Vorphase Phase 2: Synchronisation Phase 3: Synchronphase</p>
1	1	nein	<p>Fall 7: Das Verhalten ist identisch mit dem Fall 3.</p>
1	1	ja	<p>Fall 8: Mit der links angegebenen Bitkombination wird die Rücklaufsperr der Universellen Fliegende Säge aktiviert. In dem unten dargestellten Bild erfolgt in der Synchronphase eine Bewegungsumkehr der Master-Achse, so dass diese rückwärts über die Koppelposition hinaus fährt. Bei dieser rückwärtigen Bewegung der Masterachse wird ab dem Erreichen der Koppelposition die Geschwindigkeit der Slave-Achse in einem Tick abgebaut. Es wird also ein Rücklaufen der Slave-Achse insofern verhindert, dass ab dem Erreichen der Koppelpositon die Slave-Geschwindigkeit abgebaut wird.</p> <p>Durch den Geschwindigkeitsabbau in einem Tick kann die Schleppabstandsüberwachung der Achse ausgelöst werden.</p>



9 Diagonalsäge

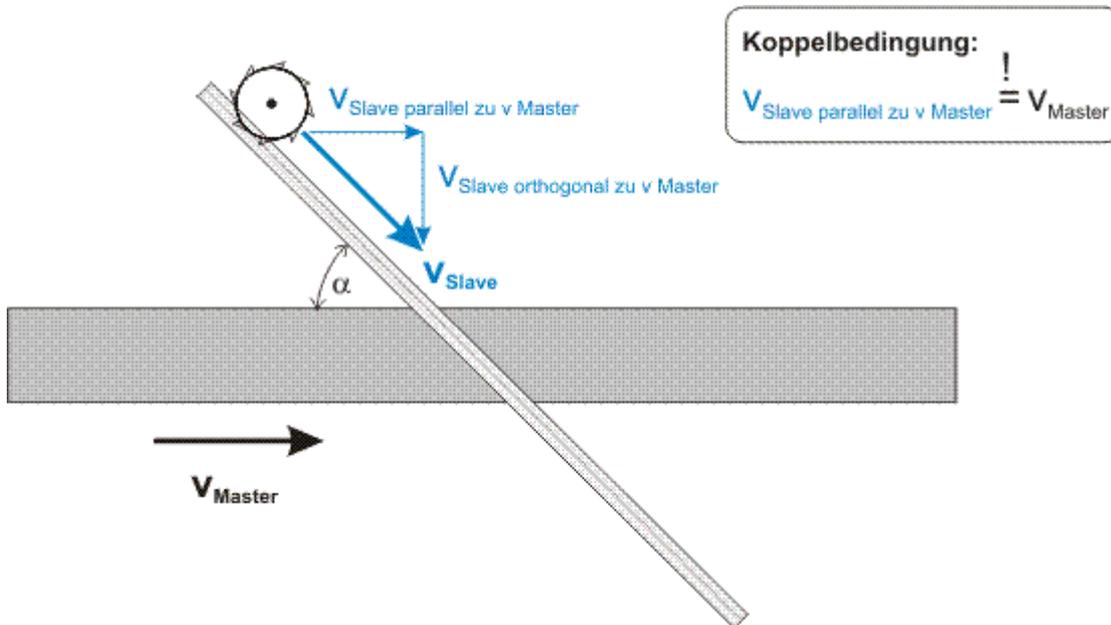
Bei der Diagonalsäge handelt es sich im einfachsten Fall um eine Synchronisation auf Geschwindigkeit.

Im Gegensatz zur Fliegenden Säge wird bei der Universellen Fliegenden Säge nicht der Winkel zwischen der Master- und Slavebewegungsrichtung angegeben, sondern der Koppelfaktor. Dieser Koppelfaktor muss so berechnet werden, dass die Geschwindigkeitskomponente der Slave-Achse in Richtung der Masterachse in der Synchronphase gleich der Mastergeschwindigkeit ist. Für die Berechnung des Koppelfaktors ergibt sich somit die folgende Gleichung:

$$F_{CouplingFactor} = \frac{1}{\cos(\alpha)} \cdot F_{Gear\ Ratio}$$

Ein Winkel α von 90° kann natürlich nicht realisiert werden, da die Slave-Achse dann keine Geschwindigkeitskomponente in Richtung der Master-Achse besitzt.

Schematische Darstellung einer "Fliegenden Diagonalsäge"



10 Interfaces

Zyklisches Achsinterface

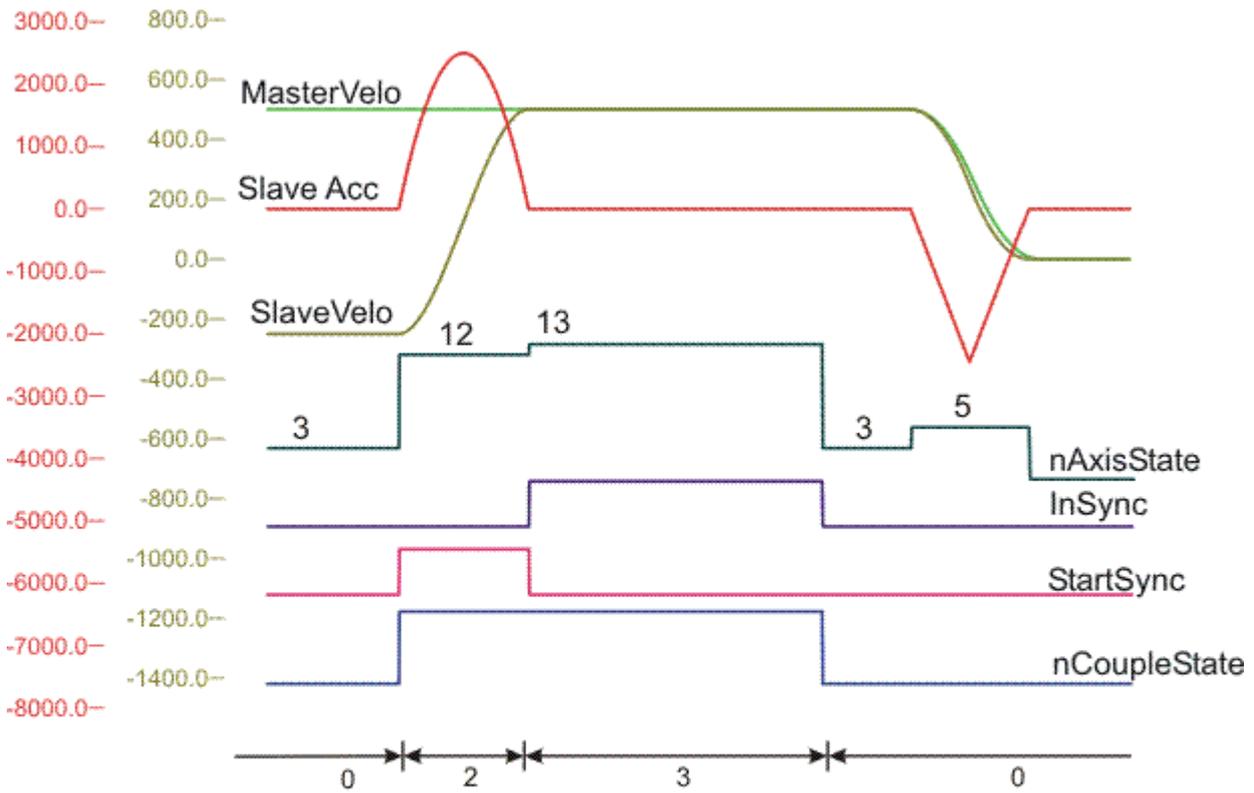
Die Variable `nAxisState` In dem zyklischen Achsinterface gibt den Bewegungszustand der Slave- oder Master-Achse an. In der Tabelle sind die Bewegungszustände der Slave-Achse beschrieben. Bei allen nicht aufgeführten States handelt es sich um Master-States, die separat erläutert sind.



Die Slave-States gelten vorerst nur für Slaves vom Typ "Universelle Fliegende Säge"!

Bewegungszustand / Fahrphasen der kontinuierlichen Slave-Achse (Servo)			
nAxisState	Typ	Beschreibung	
0	Allgemein	Generator nicht aktiv	(INACTIVE)
sonst	Master-State	Erläuterung, siehe TwinCAT NC General-Dokumentation	
11	Slave-State	Slave befindet sich in einer Bewegungs-Vorphase	(PREPHASE)
12	Slave-State	Slave befindet sich in der Synchronisierungsphase	(SYNCHRONIZING)
13	Slave-State	Slave ist aufsynchronisiert und fährt synchron	(SYNCHRON)

Das Bild zeigt eine typische Kopplung mit der Universellen Fliegenden Säge. Vor dem Start der Universellen Fliegenden Säge handelt es sich um 2 unabhängige Masterachsen (`nAxisState < 10`). Nach dem Start der Kopplung in der Synchronisierungsphase ist der `nAxisState = 12`. Der Wechsel in die Synchronphase wird durch den `nAxisState = 13` signalisiert. Nachdem die Slave-Achse von der Master-Achse abgekoppelt worden ist, ist der `nAxisState < 10`, es handelt sich also wieder um einen Master-Achsen Bewegungszustand.



Phase 0: unabhängige Masterachsen

Phase 2: Synchronisation

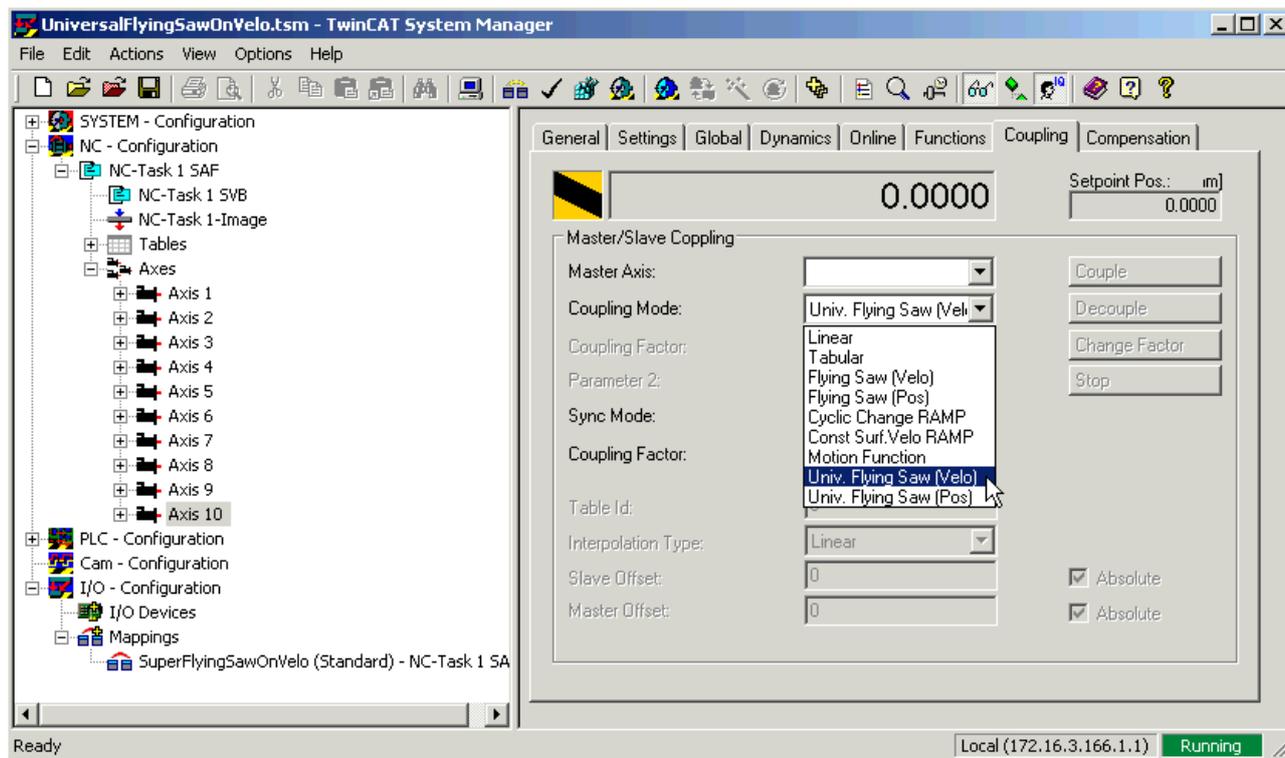
Phase 3: Synchronphase

ADS-Schnittstelle

Dokumentation der [ADS-Schnittstelle](#).

11 Bedienung aus dem Systemmanager

Die Universelle Fliegende Säge kann zu Inbetriebnahmezwecken auch direkt aus dem System Manager gestartet werden.



12 PLC-API

In vielen Anlagen werden Werkstücke während des Transportes bearbeitet. Dazu ist es notwendig, Werkzeuge und Werkstücke bezüglich Position und Geschwindigkeit zu synchronisieren, so dass das Werkzeug wie an einem stillstehenden Werkstück eingesetzt werden kann. Ein Beispiel für eine solche Anwendung ist eine Säge, die während der Fahrt einen Schnitt quer durch das transportierte Material führt (Fliegende Säge). Um diese Art der Anwendungen zu realisieren, stellt TwinCAT die Fliegende Säge zur Verfügung.

Zur einfachen Handhabung der Fliegenden Säge dient die TwinCAT PLC Library **TcMC2_FlyingSaw.lib**, die als *zusätzliches Produkt* erhältlich ist. Ein [Beispielprogramm zum Thema Fliegende Säge \[► 44\]](#) verwendet diese Bibliothek.

Zur Bedienung der Universellen Fliegenden Säge stehen die folgenden SPS-Funktionsbausteine zur Verfügung.

Funktion	Baustein
Synchronisation auf Geschwindigkeit	MC_GearInVelo [► 33]
Synchronisation auf Position	MC_GearInPos [► 36]
Lesen der charakteristischen Kennwerte	MC_ReadFlyingSawCharacteristics [► 39]
Abkoppeln des Slaves	MC_GearOut [► 40]

12.1 TcMC2_FlyingSaw

12.1.1 MC_GearInVelo



Mit dem Funktionsbaustein **MC_GearInVelo** wird eine lineare Master-Slave-Kopplung (Getriebekopplung) aktiviert. Falls die Master-Achse bereits in Bewegung ist, synchronisiert sich die Slave-Achse auf die Master-Geschwindigkeit. Der Baustein akzeptiert einen festen Getriebefaktor im Zähler-Nenner-Format.

Die Slave-Achse kann mit dem Funktionsbaustein [MC_GearOut](#) abgekoppelt werden. Wird der Slave während der Fahrt abgekoppelt, so behält er seine Geschwindigkeit bei und kann mit [MC_Stop](#) oder [MC_Halt](#) angehalten werden.

Eingänge

```

VAR_INPUT
  Execute          : BOOL;
  RatioNumerator   : LREAL;
  RatioDenominator : UINT;
  SyncMode         : ST_SyncMode;
  Velocity         : LREAL;
  Acceleration     : LREAL;

```

```

Deceleration      : LREAL;
Jerk              : LREAL;
BufferMode        : MC_BufferMode;
Options           : ST_GearInVeloOptions;
END_VAR
    
```

Execute	Mit einer steigenden Flanke am Eingang <i>Execute</i> wird das Kommando ausgeführt.
RatioNumerator	Getriebefaktor Zähler. Alternativ kann der Getriebefaktor als Fließkommawert angegeben werden, wenn der Nenner 1 ist.
RatioDenominator	Getriebefaktor Nenner
SyncMode	In der Datenstruktur <i>SyncMode</i> [▶ 41] werden über einzelne Flags die Randbedingungen für den Synchronisierungsvorgang festgelegt.
Velocity	Maximale Slave-Geschwindigkeit in der Synchronisierungsphase. Wenn keine Geschwindigkeit angegeben wird, wird die Maximalgeschwindigkeit der Achse aus den Systemmanager-Daten verwendet. Die hier angegebene Geschwindigkeit wird nur überprüft, wenn die Überprüfung mit der Variablen <i>SyncMode</i> [▶ 41] aktiviert wird.
Acceleration	Maximale Slave-Beschleunigung in der Synchronisierungsphase. Wenn keine Beschleunigung angegeben wird, wird die Maximalbeschleunigung der Achse aus den Systemmanager-Daten verwendet. Die hier angegebene Beschleunigung wird nur überprüft, wenn die Überprüfung mit der Variablen <i>SyncMode</i> [▶ 41] aktiviert wird.
Deceleration	Maximale Slave-Verzögerung in der Synchronisierungsphase. Wenn keine Verzögerung angegeben wird, wird die maximale Verzögerung der Achse aus den Systemmanager-Daten verwendet. Die hier angegebene Verzögerung wird nur überprüft, wenn die Überprüfung mit der Variablen <i>SyncMode</i> [▶ 41] aktiviert wird.
Jerk	Maximaler Slave-Ruck in der Synchronisierungsphase. Wenn kein Ruck angegeben wird, wird der maximale Ruck der Achse aus den Systemmanager-Daten verwendet. Der hier angegebene Ruck wird nur überprüft, wenn die Überprüfung mit der Variablen <i>SyncMode</i> [▶ 41] aktiviert wird.
BufferMode	Zur Zeit nicht implementiert
Options	Zur Zeit nicht implementiert

Bei einem Verhältnis 1:4 muss der *RatioNumerator* 1 sein und der *RatioDenominator* 4. Alternativ kann der *RatioDenominator* 1 sein und der Getriebefaktor wird im *RatioNumerator* als Fließkommazahl 0.25 angegeben. Der *RatioNumerator* darf negativ sein.

Ausgänge

```

VAR_OUTPUT
  StartSync      : BOOL;
  InSync         : BOOL;
  Busy           : BOOL;
  Active         : BOOL;
  CommandAborted : BOOL;
  Error          : BOOL;
  ErrorID        : UDINT;
END_VAR
    
```

StartSync	Wird TRUE, wenn die Synchronisierung mit der Master-Achse gestartet wurde.
InSync	Wird TRUE, wenn die Kopplung erfolgreich durchgeführt wurde und die Slave-Achse mit der Master-Achse synchronisiert ist.

Busy	Der <i>Busy</i> -Ausgang wird TRUE, sobald das Kommando mit <i>Execute</i> gestartet wird und bleibt TRUE, solange der Befehl abgearbeitet wird. Wenn <i>Busy</i> wieder FALSE wird, so ist der Funktionsbaustein bereit für einen neuen Auftrag. Gleichzeitig ist einer der Ausgänge <i>InSync</i> , <i>CommandAborted</i> oder <i>Error</i> gesetzt.
Active	Active zeigt an, dass das Kommando ausgeführt wird. (zur Zeit ist Active=Busy, siehe BufferMode)
CommandAborted	Wird TRUE, wenn das Kommando nicht vollständig ausgeführt werden konnte. Die Achse kann während des Koppelvorgangs entkoppelt worden sein (gleichzeitige Kommandoausführung).
Error	Wird TRUE, sobald ein Fehler eintritt.
ErrorID	Liefert bei einem gesetzten Error-Ausgang die Fehlernummer, siehe Übersicht NC-Fehler .

Ein/Ausgänge

VAR_IN_OUT

 Master : AXIS_REF;

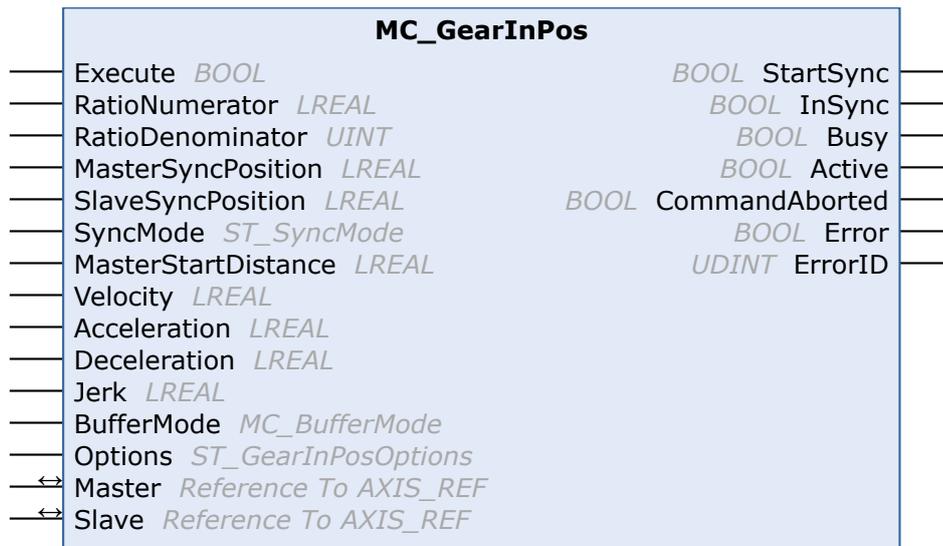
 Slave : AXIS_REF;

END_VAR

Master	Achsdatenstruktur des Masters.
Slave	Achsdatenstruktur des Slaves.

Die Achsdatenstruktur vom Typ AXIS_REF adressiert eine Achse eindeutig im System. Sie enthält unter anderem den aktuellen Status der Achse, wie Position, Geschwindigkeit oder Fehlerzustand.

12.1.2 MC_GearInPos



Der Funktionsbaustein *MC_GearInPos* synchronisiert eine Slave-Achse positionsgenau mit einer Master-Achse (Fliegende Säge). Die Synchronongeschwindigkeit wird genau am Synchronpunkt von Master und Slave erreicht.

Die Master-Achse muss sich bereits in Bewegung befinden, sonst ist die Synchronisierung nicht möglich.

Die Slave-Achse kann mit dem Funktionsbaustein *MC_GearOut* abgekoppelt werden. Wird der Slave während der Fahrt abgekoppelt, so behält er seine Geschwindigkeit bei und kann mit *MC_Stop* oder *MC_Halt* angehalten werden.

Eingänge

```

VAR_INPUT
    Execute           : BOOL;
    RatioNumerator    : LREAL;
    RatioDenominator  : UINT;
    MasterSyncPosition : LREAL;
    SlaveSyncPosition : LREAL;
    SyncMode          : ST_SyncMode;
    MasterStartDistance : LREAL;
    Velocity          : LREAL;
    Acceleration      : LREAL;
    Deceleration      : LREAL;
    Jerk              : LREAL;
    BufferMode         : MC_BufferMode;
    Options           : ST_GearInPosOptions;
END_VAR
    
```

Execute	Mit einer steigenden Flanke am Eingang <i>Execute</i> wird das Kommando ausgeführt.
RatioNumerator	Getriebefaktor Zähler. Alternativ kann der Getriebefaktor als Fließkommawert angegeben werden, wenn der Nenner 1 ist.
RatioDenominator	Getriebefaktor Nenner
MasterSyncPosition	Synchronposition des Masters
SlaveSyncPosition	Synchronposition des Slaves
SyncMode	In der Datenstruktur <i>SyncMode</i> [▶ 41] werden über einzelne Flags die Randbedingungen für den Synchronisierungsvorgang festgelegt.
MasterStartDistance	Zur Zeit nicht implementiert

Velocity	<p>Maximale Slave-Geschwindigkeit in der Synchronisierungsphase. Wenn keine Geschwindigkeit angegeben wird, wird die Maximalgeschwindigkeit der Achse aus den Systemmanager-Daten verwendet.</p> <p>Die hier angegebene Geschwindigkeit wird nur überprüft, wenn die Überprüfung mit der Variablen <code>SyncMode</code> [▶ 41] aktiviert wird.</p>
Acceleration	<p>Maximale Slave-Beschleunigung in der Synchronisierungsphase. Wenn keine Beschleunigung angegeben wird, wird die Maximalbeschleunigung der Achse aus den Systemmanager-Daten verwendet.</p> <p>Die hier angegebene Beschleunigung wird nur überprüft, wenn die Überprüfung mit der Variablen <code>SyncMode</code> [▶ 41] aktiviert wird.</p>
Deceleration	<p>Maximale Slave-Verzögerung in der Synchronisierungsphase. Wenn keine Verzögerung angegeben wird, wird die maximale Verzögerung der Achse aus den Systemmanager-Daten verwendet.</p> <p>Die hier angegebene Verzögerung wird nur überprüft, wenn die Überprüfung mit der Variablen <code>SyncMode</code> [▶ 41] aktiviert wird.</p>
Jerk	<p>Maximaler Slave-Ruck in der Synchronisierungsphase. Wenn kein Ruck angegeben wird, wird der maximale Ruck der Achse aus den Systemmanager-Daten verwendet.</p> <p>Der hier angegebene Ruck wird nur überprüft, wenn die Überprüfung mit der Variablen <code>SyncMode</code> [▶ 41] aktiviert wird.</p>
BufferMode	Zur Zeit nicht implementiert
Options	Zur Zeit nicht implementiert

Bei einem Verhältnis 1:4 muss der *RatioNumerator* 1 sein und der *RatioDenominator* 4. Alternativ kann der *RatioDenominator* 1 sein und der Getriebefaktor wird im *RatioNumerator* als Fließkommazahl 0.25 angegeben. Der *RatioNumerator* darf negativ sein.

Ausgänge

```

VAR_OUTPUT
  StartSync      : BOOL;
  InSync         : BOOL;
  Busy           : BOOL;
  Active         : BOOL;
  CommandAborted : BOOL;
  Error         : BOOL;
  ErrorID       : UDINT;
END_VAR
    
```

StartSync	Wird TRUE, wenn die Synchronisierung mit der Master-Achse gestartet wurde.
InSync	Wird TRUE, wenn die Kopplung erfolgreich durchgeführt wurde und die Slave-Achse mit der Master-Achse synchronisiert ist.
Busy	Der <i>Busy</i> -Ausgang wird TRUE, sobald das Kommando mit <i>Execute</i> gestartet wird und bleibt TRUE, solange der Befehl abgearbeitet wird. Wenn <i>Busy</i> wieder FALSE wird, so ist der Funktionsbaustein bereit für einen neuen Auftrag. Gleichzeitig ist einer der Ausgänge <i>InSync</i> , <i>CommandAborted</i> oder <i>Error</i> gesetzt.
Active	Active zeigt an, dass das Kommando ausgeführt wird. (zur Zeit ist Active=Busy, siehe BufferMode)
CommandAborted	Wird TRUE, wenn das Kommando nicht vollständig ausgeführt werden konnte. Die Achse kann während des Koppelvorgangs entkoppelt worden sein (gleichzeitige Kommandoausführung).
Error	Wird TRUE, sobald ein Fehler eintritt.

ErrorID	Liefert bei einem gesetzten Error-Ausgang die Fehlernummer, siehe Übersicht NC-Fehler .
----------------	---

Ein/Ausgänge

```
VAR_IN_OUT
  Master : AXIS_REF;
  Slave  : AXIS_REF;
END_VAR
```

Master	Achsdatenstruktur des Masters.
Slave	Achsdatenstruktur des Slaves.

Die Achsdatenstruktur vom Typ `AXIS_REF` adressiert eine Achse eindeutig im System. Sie enthält unter anderem den aktuellen Status der Achse, wie Position, Geschwindigkeit oder Fehlerzustand.

12.1.3 MC_ReadFlyingSawCharacteristics



Mit dem Funktionsbaustein *MC_ReadFlyingSawCharacteristics* können die charakteristischen Kennwerte der Synchronisierungsphase der Universellen Fliegenden Säge ausgelesen werden.

Eingänge

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

Execute	Mit einer steigenden Flanke wird das Auslesen der Kennwerte aus der TwinCAT-NC gestartet.
----------------	---



Die berechneten Daten stehen erst nach dem Start der Universellen Fliegenden Säge zur Verfügung.

Ausgänge

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

Done	Wird TRUE, wenn der Datensatz erfolgreich gelesen worden ist.
Busy	Der <i>Busy</i> -Ausgang wird TRUE, sobald das Kommando mit <i>Execute</i> gestartet wird und bleibt TRUE, solange der Befehl abgearbeitet wird. Wenn <i>Busy</i> wieder FALSE wird, so ist der Funktionsbaustein bereit für einen neuen Auftrag. Gleichzeitig ist einer der Ausgänge <i>Done</i> oder <i>Error</i> gesetzt.
Error	Wird TRUE, sobald ein Fehler eintritt.
ErrorID	Liefert bei einem gesetzten ERR-Ausgang die Fehlernummer, siehe <u>Übersicht NC-Fehler</u> .

Ein/Ausgänge

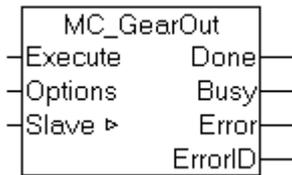
```
VAR_IN_OUT
    Slave : AXIS_REF;
    CamTableCharac : MC_FlyingSawCharacValues;
END_VAR
```

Slave	Achsstruktur des Slaves.
CamTableCharac	Struktur mit den <u>charakteristischen Kennwerten</u> . ▶ 42

Die Achsdatenstruktur vom Typ *AXIS_REF* adressiert eine Achse eindeutig im System. Sie enthält unter anderem den aktuellen Status der Achse, wie Position, Geschwindigkeit oder Fehlerzustand.

12.2 TcMC2

12.2.1 MC_GearOut



Mit dem Funktionsbaustein *MC_GearOut* wird ein Master-Slave-Kopplung deaktiviert.

i Wenn eine Slave-Achse in der Bewegung abgekoppelt wird, so wird sie nicht automatisch gestoppt, sondern sie erreicht eine konstante Geschwindigkeit, mit der sie endlos weiterfährt. Die Achse kann mit *MC_Halt* oder *MC_Stop* angehalten werden.

HINWEIS

Wenn der Sollwertgeneratortype der Achse auf "7 Phasen (optimiert)" eingestellt ist, wird die Slaveachse nach dem Abkoppeln beschleunigungsfrei gefahren und mit der sich einstellenden konstanten Geschwindigkeit weitergefahren. Es erfolgt keine Positionierung um den mit dem Koppelfaktor umgerechneten Masterverfahrweg, sondern es stellt sich ein Verhalten wie nach einem *MC_MoveVelocity* ein. In TwinCAT 2.10 ist der Sollwertgeneratortyp wählbar. Ab TwinCAT 2.11 ist der Sollwertgeneratortype fest auf "7 Phasen (optimiert)" eingestellt. Bei der Umstellung eines Projektes von TwinCAT 2.10 auf TwinCAT 2.11 ergibt sich damit das hier beschriebene Verhalten. Ein Update bestehender Applikationen auf Version 2.11 kann daher, je nach Anwendung, eine Anpassung des SPS-Programms erforderlich machen.

Eingänge

```

VAR_INPUT
    Execute : BOOL;
    Options : ST_GearOutOptions;
END_VAR
  
```

Execute	Mit einer steigenden Flanke am Eingang <i>Execute</i> wird das Kommando ausgeführt.
Options	Zurzeit nicht implementiert

Ausgänge

```

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

Done	Wird TRUE, wenn die Achse erfolgreich abgekoppelt wurde.
Busy	Der <i>Busy</i> -Ausgang wird TRUE, sobald das Kommando mit <i>Execute</i> gestartet wird und bleibt TRUE, solange der Befehl abgearbeitet wird. Wenn <i>Busy</i> wieder FALSE wird, so ist der Funktionsbaustein bereit für einen neuen Auftrag. Gleichzeitig ist einer der Ausgänge <i>Done</i> oder <i>Error</i> gesetzt.
Error	Wird im Fehlerfall TRUE.
ErrorID	Liefert bei einem gesetzten Error-Ausgang die Fehlernummer.

Ein/Ausgänge

```

VAR_IN_OUT
    Slave : AXIS_REF;
END_VAR
  
```

Slave	Achsdatenstruktur des Slaves.
--------------	-------------------------------

Die Achsdatenstruktur vom Typ `AXIS_REF` adressiert eine Achse eindeutig im System. Sie enthält unter anderem den aktuellen Status der Achse, wie Position, Geschwindigkeit oder Fehlerzustand.

12.3 Datentypen

12.3.1 ST_SyncMode

```

TYPE ST_SyncMode :
STRUCT
  (* mode *)
  GearInSyncMode                : E_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 E_GearInSyncMode :
(
  (* synchronization based on the master position, slave dynamics depend on master dynamics *)
  GEARINSYNCMODE_POSITIONBASED,

  (* synchronization based on a standalone slave PTP profile, master independent slave dynamics *)
  GEARINSYNCMODE_TIMEBASED
);
END_TYPE

```



Das zeitbasierte Bewegungsprofil (`GEARINSYNCMODE_TIMEBASED`) ist zurzeit nur für den Funktionsbaustein `MC_GearInVelo` implementiert.

Siehe auch:

[Wirkungsweise der einzelnen Bits \[► 18\]](#)

[Fehlercodes](#)

12.3.2 MC_FlyingSawCharacValues

```

TYPE MC_FlyingSawCharacValues :
STRUCT
  (* Master Velocity*)
  fMasterVeloNom      : LREAL; (* 1. master nominal velocity (normed=> 1.0) *)

  (* characteristic slave data *)
  (*=====*)

  (* Start of cam table *)
  fMasterPosStart    : LREAL; (* 2. master start position*)
  fSlavePosStart     : LREAL; (* 3. slave start position *)
  fSlaveVeloStart    : LREAL; (* 4. slave start velocity *)
  fSlaveAccStart     : LREAL; (* 5. slave start acceleration *)
  fSlaveJerkStart    : LREAL; (* 6. slave start jerk *)

  (* End of cam table*)
  fMasterPosEnd      : LREAL; (* 7. master end position *)
  fSlavePosEnd       : LREAL; (* 8. slave end position *)
  fSlaveVeloEnd      : LREAL; (* 9. slave end velocity *)
  fSlaveAccEnd       : LREAL; (* 10. slave end acceleration *)
  fSlaveJerkEnd      : LREAL; (* 11. slave end jerk *)

  (* minimum slave position *)
  fMPosAtSPosMin    : LREAL; (* 12. master position AT slave minimum position *)
  fSlavePosMin      : LREAL; (* 13. slave minimum position *)

  (* minimum Slave velocity *)
  fMPosAtSVeloMin   : LREAL; (* 14. master position AT slave minimum velocity *)
  fSlaveVeloMin     : LREAL; (* 15. slave minimum velocity *)

  (* minimum slave acceleration *)
  fMPosAtSAccMin    : LREAL; (* 16. master position AT slave minimum acceleration *)
  fSlaveAccMin      : LREAL; (* 17. slave minimum acceleration *)
  fSVeloAtSAccMin   : LREAL; (* 18. slave velocity AT slave minimum acceleration *)

  (* minimum slave jerk and dynamic momentum *)
  fSlaveJerkMin     : LREAL; (* 19. slave minimum jerk *)
  fSlaveDynMomMin   : LREAL; (* 20. slave minimum dynamic momentum (NOT SUPPORTED YET !) *)

  (* maximum slave position *)
  fMPosAtSPosMax    : LREAL; (* 21. master position AT slave maximum position *)
  fSlavePosMax      : LREAL; (* 22. slave maximum position *)

  (* maximum Slave velocity *)
  fMPosAtSVeloMax   : LREAL; (* 23. master position AT slave maximum velocity *)
  fSlaveVeloMax     : LREAL; (* 24. slave maximum velocity *)

  (* maximum slave acceleration *)
  fMPosAtSAccMax    : LREAL; (* 25. master position AT slave maximum acceleration *)
  fSlaveAccMax      : LREAL; (* 26. slave maximum acceleration *)
  fSVeloAtSAccMax   : LREAL; (* 27. slave velocity AT slave maximum acceleration *)

  (* maximum Slave slave jerk and dynamic momentum *)
  fSlaveJerkMax     : LREAL; (* 28. slave maximum jerk *)
  fSlaveDynMomMax   : LREAL; (* 29. slave maximum dynamic momentum (NOT SUPPORTED YET !) *)

  (* mean and effective values *)
  fSlaveVeloMean    : LREAL; (* 30. slave mean absolute velocity (NOT SUPPORTED YET !) *)
  fSlaveAccEff      : LREAL; (* 31. slave effective acceleration (NOT SUPPORTED YET !) *)

  (* reserved space for future extension *)
  reserved          : ARRAY[32..47] OF LREAL;

  (* organization structure of the cam table *)
  CamTableID       : UDINT;
  NumberOfRows     : UDINT; (* number of cam table entries, e.g. number of points *)
  NumberOfColumns  : UDINT; (* number of table columns, typically 1 or 2 *)
  TableType        : UINT; (* MC_TableType *)
  Periodic         : BOOL;

  reserved2        : ARRAY[1..121] OF BYTE;
END_STRUCT
END_TYPE

```

Typdefinition für die charakteristischen Kenngrößen einer Fliegende Säge Synchronisierung.

12.4 Beispielprogramm

12.4.1 Beispielprogramm Fliegende Säge

Das Beispielprogramm zeigt einen typischen zyklischen Ablauf einer Fliegenden Säge. Das Programm verwaltet drei Achsen: Master, Slave und Tool. Die Slave- und Tool-Achsen stehen zu Beginn in ihrer Grundposition bei 0. Zunächst wird die Master-Achse gestartet, die man sich als Transportsystem vorstellen kann, das endloses Material in eine Schneidevorrichtung transportiert. Das Material soll in konstanten Abständen geschnitten werden. Dazu wird die Slave-Achse, also die Fliegende Säge auf die Master-Achse aufsynchronisiert und fährt synchron mit dem Transport mit. In dieser Phase wird die Tool-Achse aktiviert, die die eigentliche Bearbeitung durchführt. Anschließend wird die Slave-Achse abgekoppelt und an ihre Ausgangsposition zurückgefahren. Noch während der Rückfahrt wird die nächste Synchronisierung gestartet und ein Punkt im konstanten Abstand zur letzten Bearbeitungsposition anvisiert.

Das Beispielprogramm benötigt die Fliegende Säge Bibliothek und läuft vollständig im Simulationsmodus. Der Ablauf kann im TwinCAT-Scope-View mit beiliegender Konfiguration verfolgt werden.

Zum Speichern des Beispielprogramms hier klicken:

https://infosys.beckhoff.com/content/1031/TS5055_TCNC_FlyingSaw/Resources/437571723.zip

13 Fehlersituationen und Fehlercodes:

Folgende Situationen bzw. Parameter sind nicht erlaubt und führen zum Fehler:

- Eine Kopplung mit der Universellen Fliegenden Säge ist zur Zeit nicht möglich, wenn die Mastergeschwindigkeit zum Koppelzeitpunkt Null ist und die Slave Geschwindigkeit ungleich Null ist. Die Kombination aus Master-Geschwindigkeit und Slave-Geschwindigkeit gleich Null ist hingegen erlaubt.
- Getriebefaktor ist Null (ansonsten darf der Getriebefaktor jeden Wert annehmen)
- Maximalgeschwindigkeit, Beschleunigung, Verzögerung oder Ruck sind negativ (hierbei wird Bezug auf die Achsdaten (Maschinendaten) des Systemmanagers genommen, die vorzeichenlos als Betrag interpretiert werden)
- Die Synchronpositionen müssen in Bezug auf die Startpositionen zum Koppelzeitpunkt für den Master immer geometrisch in der Zukunft liegen und
 - a.) für den Slave ebenfalls geometrisch in der Zukunft liegen, wenn der Koppelfaktor größer Null ist,
 - b) für den Slave geometrisch in der Vergangenheit liegen, wenn der Koppelfaktor kleiner Null ist.

HINWEIS

Verwendung von Tabellen

Für die Realisierung der Funktionalität der Universellen Fliegenden Säge wird intern eine **Motion Function Tabelle** verwendet. Die intern verwendete Tabelle darf nicht durch anderweitige Tabellenfunktionen (z.B. durch eine Kurvenscheibe) belegt sein. Anderenfalls wird die Kopplung mit einem Fehlercode abgelehnt. Die Tabellen-Id der intern verwendeten Tabelle kann mit der folgenden Gleichung berechnet werden:

$$TableId_{FlyingSaw} = 256 - SlaveAxisId$$

Beschreibung der Fehlercodes

Beschreibung der Fehlercodes

Mehr Informationen:
www.beckhoff.de/ts5055

Beckhoff Automation GmbH & Co. KG
Hülshorstweg 20
33415 Verl
Deutschland
Telefon: +49 5246 9630
info@beckhoff.de
www.beckhoff.de

