

Technologiemodul



Winder Dancer-controlled _____

Referenzhandbuch

DE

1	Über diese Dokumentation	4
1.1	Dokumenthistorie	6
1.2	Verwendete Konventionen	7
1.3	Gestaltung der Sicherheitshinweise	8
2	Sicherheitshinweise	9
3	Übersicht der Funktionen des Technologiemoduls	11
3.1	Funktionsbeschreibung	11
3.2	TM-Funktionsübersicht	13
3.3	Wichtige Hinweise zum Betrieb des Technologiemoduls	14
4	Anwendungsbereiche	16
5	Funktionsbaustein L_TW2P_WinderDancerCtrlState	17
5.1	Eingänge und Ausgänge	19
5.2	Eingänge	19
5.3	Ausgänge	23
5.4	Parameter	26
6	State Machine	32
7	Signalflussplan	33
7.1	Struktur des Signalflusses	34
7.2	Struktur der Angriffspunkte	36
8	Beschreibung der Basisfunktionen	37
8.1	Handfahren (Jogging)	38
8.2	Referenzfahrt (Homing)	40
8.3	QSP Application	43
8.4	Halt	44
8.5	Absolute Positionierung	45
8.6	Relative Positionierung	47
8.7	Endlosfahrt mit definierter Geschwindigkeit	50
8.8	Abbruch und Beendigung einer Positionierung	53
8.9	Override für Geschwindigkeit	56
9	Beschreibung der Technologiefunktionen	58
9.1	Festlegung der Wickelrichtung (Aufwickeln/Abwickeln)	58
9.2	Festlegung der Materialzuführung an den Wickler	60
9.3	Leitwertquelle für die Durchmesserberechnung	62
9.4	Drehzahlvorsteuerung	63
9.5	Durchmesserberechnung	64
9.6	Durchmesser halten	66
9.7	Durchmesser über einen Sensor vorgeben	67
9.8	Durchmesserberechnung mit Berücksichtigung der Tänzerbewegung	69
9.9	Identifikation des Durchmessers durch Anheben des Tänzers	71
9.10	Materiallängenzähler	75
9.10.1	Materiallängenzähler über Eingang 'IrSetLineVel'	76
9.10.2	Materiallängenzähler über Eingang 'IrSetLineVelDiamCalc'	77
9.10.3	Materiallängenzähler über Referenzachse 'MaterialCounterAxis'	78
9.11	Tippbetrieb mit 1/d-Bewertung	79
9.12	Synchronisierung auf die Liniengeschwindigkeit	81
9.13	Normierung der Tänzerlage	82
9.14	Teaching-Funktion für Tänzerendlagen	84

9.15	Überwachung der Tänzerposition	86
9.16	PI-Regler für die Tänzerlageregelung	87
9.17	Reduzierte Reglerdynamik bei geringer Regelabweichung	89
9.18	Zugkraftsteuerung über Kennlinienfunktion	90
9.18.1	Kennlinie für einen linearen Zugkraftverlauf (Parameter dwSelectTensCurve = 0)	91
9.18.2	Kennlinie für einen linearen Drehmomentverlauf (Parameter dwSelectTensCurve = 1)	92
9.18.3	Freidefinierbare Kennlinie mit 65 Stützstellen (Parameter dwSelectTensCurve = 2)	93
9.19	Bahnrißüberwachung	94
9.20	Persistente Variablen	96
9.21	Beschleunigungskompensation	98
9.22	Identifikation der Massenträgheitsmomente	100
9.23	Adaption der Drehzahlreglerverstärkung	104
9.24	Begrenzung der Master-Liniengeschwindigkeit	106
9.25	Beendigung des Wickelprozesses	108
	Index	110
	Ihre Meinung ist uns wichtig	112

1 Über diese Dokumentation

Diese Dokumentation ...

- enthält ausführliche Informationen zu den Funktionalitäten des Technologiemoduls "Winder Dancer-controlled (V2)";
- ordnet sich in die Handbuchsammlung "Controller-based Automation" ein. Diese besteht aus folgenden Dokumentationen:

Dokumentationstyp	Thema
Produktkatalog	Controller-based Automation (Systemübersicht, Beispieltopologien) Lenze-Controller (Produktinformationen, Technische Daten)
Systemhandbücher	Visualisierung (Systemübersicht/Beispieltopologien)
Kommunikationshandbücher Online-Hilfen	Bussysteme • Controller-based Automation EtherCAT® • Controller-based Automation CANopen® • Controller-based Automation PROFIBUS® • Controller-based Automation PROFINET®
Referenzhandbücher Online-Hilfen	Lenze-Controller: • Controller 3200 C • Controller c300 • Controller p300 • Controller p500
Software-Handbücher Online-Hilfen	Lenze Engineering Tools: • »PLC Designer« (Programmierung) • »Engineer« (Parametrierung, Konfigurierung, Diagnose) • »VisiWinNET® Smart« (Visualisierung) • »Backup & Restore« (Datensicherung, Wiederherstellung, Aktualisierung)

Weitere Technische Dokumentationen zu Lenze-Produkten

Weitere Informationen zu Lenze-Produkten, die in Verbindung mit der Controller-based Automation verwendbar sind, finden Sie in folgenden Dokumentationen:

Planung / Projektierung / Technische Daten	
☐	Produktkataloge • Controller-based Automation • Controller • Inverter Drives/Servo Drives
Montage und Verdrahtung	
☒	Montageanleitungen • Controller • Kommunikationskarten (MC-xxx) • I/O-System 1000 (EPM-Sxxx) • Inverter Drives/Servo Drives • Kommunikationsmodule
☐	Gerätehandbücher • Inverter Drives/Servo Drives
Parametrierung / Konfigurierung / Inbetriebnahme	
☐	Online-Hilfe / Referenzhandbücher • Controller • Inverter Drives/Servo Drives • I/O-System 1000 (EPM-Sxxx)
☐	Online-Hilfe / Kommunikationshandbücher • Bussysteme • Kommunikationsmodule
Beispielapplikationen und Vorlagen	
☐	Online-Hilfe / Software- und Referenzhandbücher • Application Sample i700 • Application Samples 8400/9400 • FAST Application Template Lenze/PackML • FAST Technologiemodule

Symbole:

-  Gedruckte Dokumentation
- ☐ PDF-Datei / Online-Hilfe im Lenze Engineering Tool



Tipp!

Aktuelle Dokumentationen und Software-Updates zu Lenze-Produkten finden Sie im Download-Bereich unter:

www.lenze.com

Zielgruppe

Diese Dokumentation richtet sich an alle Personen, die ein Lenze-Automatisierungssystem auf Basis der Lenze FAST Application Software programmieren und in Betrieb nehmen.

1.1

Dokumenthistorie

Version			Beschreibung
4.0	04/2021	TD06	Fehlerkorrekturen & Ergänzungen
3.0	04/2020	TD17	Fehlerkorrekturen & Ergänzungen
2.0	11/2019	TD06	Fehlerkorrekturen & Ergänzungen
1.0	04/2019	TD06	Erstausgabe

1 Über diese Dokumentation

1.2 Verwendete Konventionen

Diese Dokumentation verwendet folgende Konventionen zur Unterscheidung verschiedener Arten von Information:

Informationsart	Auszeichnung	Beispiele/Hinweise
Zahlenschreibweise		
Dezimaltrennzeichen	Punkt	Es wird generell der Dezimalpunkt verwendet. Zum Beispiel: 1234.56
Hexadezimalzahl	0x	Für Hexadezimalzahlen wird der Präfix "0x" verwendet. Beispiel: 0x60F4
Textauszeichnung		
Programmname	» «	»PLC Designer« ...
Variablenbezeichner	<i>kursiv</i>	Durch Setzen von <i>bEnable</i> auf TRUE ...
Funktionsbausteine	fett	Der Funktionsbaustein L_MC1P_AxisBasicControl ...
Funktionsbibliotheken		Die Funktionsbibliothek L_TT1P_TechnologyModules ...
Quellcode	Schriftart "Courier new"	... dwNumerator := 1; dwDenominator := 1; ...
Hyperlink	<u>unterstrichen</u>	Optisch hervorgehobener Verweis auf ein anderes Thema. Wird in dieser Online-Dokumentation per Mausklick aktiviert.
Symbole		
Seitenverweis		Verweis auf weiterführende Informationen: Seitenzahl in PDF-Datei.
Schrittweise Anleitung		Schrittweise Anleitungen sind durch ein Piktogramm gekennzeichnet.

Variablenbezeichner

Die von Lenze verwendeten Konventionen, die für die Variablenbezeichner von Lenze Systembausteinen, Funktionsbausteinen sowie Funktionen verwendet werden, basieren auf der sogenannten "Ungarischen Notation", wodurch anhand des Bezeichners sofort auf die wichtigsten Eigenschaften (z. B. den Datentyp) der entsprechenden Variable geschlossen werden kann, z. B. *xAxisEnabled*.

1.3

Gestaltung der Sicherheitshinweise

⚠ GEFAHR!

Kennzeichnet eine außergewöhnlich große Gefahrensituation. Wird dieser Hinweis nicht beachtet, kommt es zu schweren irreversiblen Verletzungen oder zum Tod.

⚠ WARNUNG!

Kennzeichnet eine außergewöhnlich große Gefahrensituation. Wird dieser Hinweis nicht beachtet, kann es zu schweren irreversiblen oder tödlichen Verletzungen kommen.

⚠ VORSICHT!

Kennzeichnet eine Gefahrensituation. Wird dieser Hinweis nicht beachtet, kann es zu leichten oder mittleren Verletzungen kommen.

HINWEIS

Kennzeichnet Sachgefahren. Wird dieser Hinweis nicht beachtet, kann es zu Sachschäden kommen.

Anwendungshinweise

Piktogramm	Signalwort	Bedeutung
	Tipp!	Nützlicher Tipp zur einfachen Bedienung
		Verweis auf andere Dokumentation

2 Sicherheitshinweise

Beachten Sie die Sicherheitshinweise in dieser Dokumentation, wenn Sie ein Automatisierungssystem oder eine Anlage mit einem Lenze-Controller in Betrieb nehmen möchten.



Die Gerätedokumentation enthält Sicherheitshinweise, die Sie beachten müssen!

Lesen Sie die mitgelieferten und zugehörigen Dokumentationen der jeweiligen Komponenten des Automatisierungssystems sorgfältig durch, bevor Sie mit der Inbetriebnahme des Controllers und der angeschlossenen Geräte beginnen.

⚠ GEFAHR!

Hohe elektrische Spannung

Personenschäden durch gefährliche elektrische Spannung

Mögliche Folgen

Tod oder schwere Verletzungen

Schutzmaßnahmen

Die Spannungsversorgung ausschalten, bevor Arbeiten an den Komponenten des Automatisierungssystems durchgeführt werden.

Nach dem Ausschalten der Spannungsversorgung spannungsführende Geräteteile und Leistungsanschlüsse nicht sofort berühren, weil Kondensatoren aufgeladen sein können.

Die entsprechenden Hinweisschilder auf dem Gerät beachten.

⚠ GEFAHR!

Personenschäden

Verletzungsgefahr besteht durch ...

- nicht vorhersehbare Motorbewegungen (z. B. ungewollte Drehrichtung, zu hohe Geschwindigkeit oder ruckhafter Lauf);
- unzulässige Betriebszustände bei der Parametrierung, während eine Online-Verbindung zum Gerät besteht.

Mögliche Folgen

Tod oder schwere Verletzungen

Schutzmaßnahmen

- Anlagen mit eingebauten Umrichtern ggf. mit zusätzlichen Überwachungs- und Schutzeinrichtungen nach den jeweils gültigen Sicherheitsbestimmungen ausrüsten (z. B. Gesetz über technische Arbeitsmittel, Unfallverhütungsvorschriften).
- Während der Inbetriebnahme einen ausreichenden Sicherheitsabstand zum Motor oder den vom Motor angetriebenen Maschinenteilen einhalten.

HINWEIS

Beschädigung oder Zerstörung von Maschinenteilen

Beschädigung oder Zerstörung von Maschinenteilen besteht durch ...

- Kurzschluss oder statische Entladungen (ESD);
- nicht vorhersehbare Motorbewegungen (z. B. ungewollte Drehrichtung, zu hohe Geschwindigkeit oder ruckhafter Lauf);
- unzulässige Betriebszustände bei der Parametrierung, während eine Online-Verbindung zum Gerät besteht.

Schutzmaßnahmen

- Vor allen Arbeiten an den Komponenten des Automatisierungssystems immer die Spannungsversorgung ausschalten.
 - Elektronische Bauelemente und Kontakte nur berühren, wenn zuvor ESD-Maßnahmen getroffen wurden.
 - Anlagen mit eingebauten Umrichtern ggf. mit zusätzlichen Überwachungs- und Schutzeinrichtungen nach den jeweils gültigen Sicherheitsbestimmungen ausrüsten (z. B. Gesetz über technische Arbeitsmittel, Unfallverhütungsvorschriften).
-

3 Übersicht der Funktionen des Technologiemoduls

3.1 Funktionsbeschreibung

Varianten von Wickelantrieben

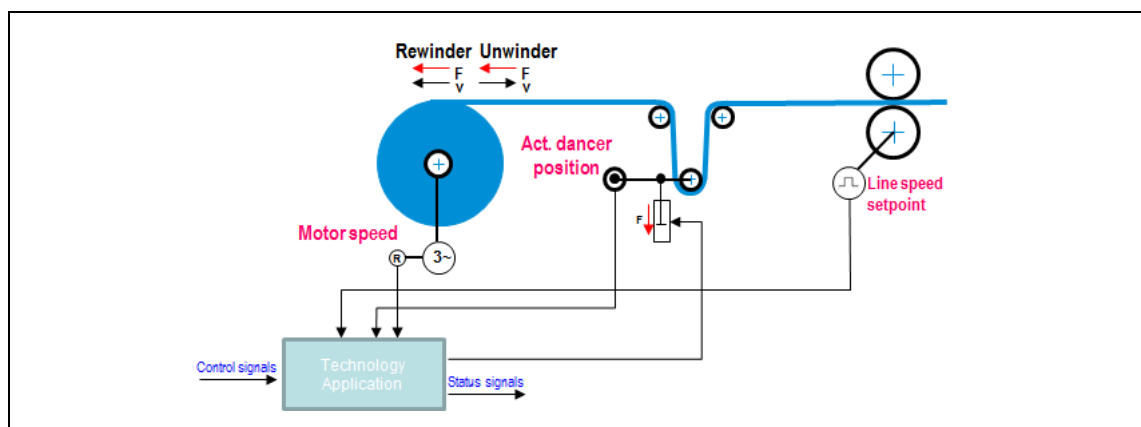
Wickelantriebe sind in vielen technologischen Prozessen ein wesentlicher Bestandteil einer Gesamtanlage. In Abhängigkeit von Material und Wickelprozess kommen unterschiedliche Steuer- und Regelverfahren zum Einsatz:

- Tänzerlageregelung
- Zugkraftsteuerung
- Zugkraftregelung

Kurzbeschreibung

Mit dieser standardisierten Applikation kann ein tänzerlagegeregelter Wickelantrieb projiziert werden.

Bei der Tänzerlageregelung wird der Antrieb in Drehzahlregelung betrieben. Zur Vorsteuerung dient das Liniengeschwindigkeitssignal multipliziert mit dem Kehrwert des Durchmessers. Die Tänzerposition wird erfasst und mit der Sollposition verglichen. Bei einer Abweichung sorgt der Tänzerlageregler für eine Korrektur des Drehzahlsollwertes.



[3-1] Aufbau eines tänzergeregelten Wicklers

- Für die Tänzerlageregelung ist es möglich, einen PI-Regler zu verwenden. Die Tänzerendlagen können über die Parameterstruktur [L_TW2P_scPar_WinderDancerCtrlState](#) (26) eingestellt werden oder mit der Teaching-Funktion identifiziert werden. Die Zugkraftsteuerung kann über eine lineare Kennlinienfunktion eingestellt werden. Die Sollwerte der Zugkraft können direkt am Ausgang des Technologiemoduls ausgelesen werden.
 - Es stehen insgesamt drei Kennlinien für die Zugkraftsteuerung zur Verfügung:
 - Kennlinie für einen linearen Zugkraftverlauf
 - Kennlinie für einen linearen Drehmomentverlauf
 - Frei definierbare Kennlinie mit 64 Stützpunkten
- Zudem kann während der Tänzerlageregelung das Drehmoment mit der Beschleunigungskompensation vorgesteuert werden.

Ergänzend bietet das Technologiemodul die Möglichkeit, das Massenträgheitsmoment der Wicklerachse zu identifizieren und für die Parametrierung des Technologiemoduls zu verwenden. Zudem ist eine Adaption der Drehzahlreglerverstärkung, in Abhängigkeit des aktuellen Massenträgheitsmoments, im laufenden Betrieb ausführbar.

3.2 TM-Funktionsübersicht

Neben den Funktionen zur Bedienung des Funktionsbausteins **L_MC1P_AxisBasicControlV2** bietet das Technologiemodul folgende Funktionalitäten:

Funktionalität
Basic Motion (V2):
Handfahren (Jogging) (📖 38)
Referenzfahrt (Homing) (📖 40)
OSP Application (📖 43)
Halt
Absolute Positionierung (📖 45)
Relative Positionierung (📖 47)
Endlosfahrt mit definierter Geschwindigkeit (📖 50)
Abbruch und Beendigung einer Positionierung (📖 53)
Override für Geschwindigkeit (📖 56)
Winder Dancer-controlled (V2):
Festlegung der Wickelrichtung (Aufwickeln/Abwickeln) (📖 58)
Automatische Erkennung der Wickelrichtung (📖 59)
Leitwertquelle für die Durchmesserberechnung (📖 62)
Drehzahlvorsteuerung (📖 63)
Durchmesserberechnung (📖 64)
Durchmesser halten (📖 66)
Durchmesser über einen Sensor vorgeben (📖 67)
Durchmesserberechnung mit Berücksichtigung der Tänzerbewegung (📖 69)
Materiallängenzähler (📖 75)
Synchronisierung auf die Liniengeschwindigkeit (📖 81)
Normierung der Tänzerlage (📖 82)
Teaching-Funktion für Tänzerendlagen (📖 84)
Überwachung der Tänzerposition (📖 86)
PI-Regler für die Tänzerlageregelung (📖 87)
Zugkraftsteuerung über Kennlinienfunktion (📖 90)
Bahnrißüberwachung (📖 94)
Persistente Variablen (📖 96)
Begrenzung der Master-Liniengeschwindigkeit (📖 106)
Beschleunigungskompensation (📖 98)
Identifikation der Massenträgheitsmomente (📖 100)
Adaption der Drehzahlreglerverstärkung (📖 104)
Identifikation des Durchmessers durch Anheben des Tänzers (📖 71)
Reduzierte Reglerdynamik bei geringer Regelabweichung (📖 89)
Beendigung des Wickelprozesses (📖 108)



»PLC Designer« Online-Hilfe

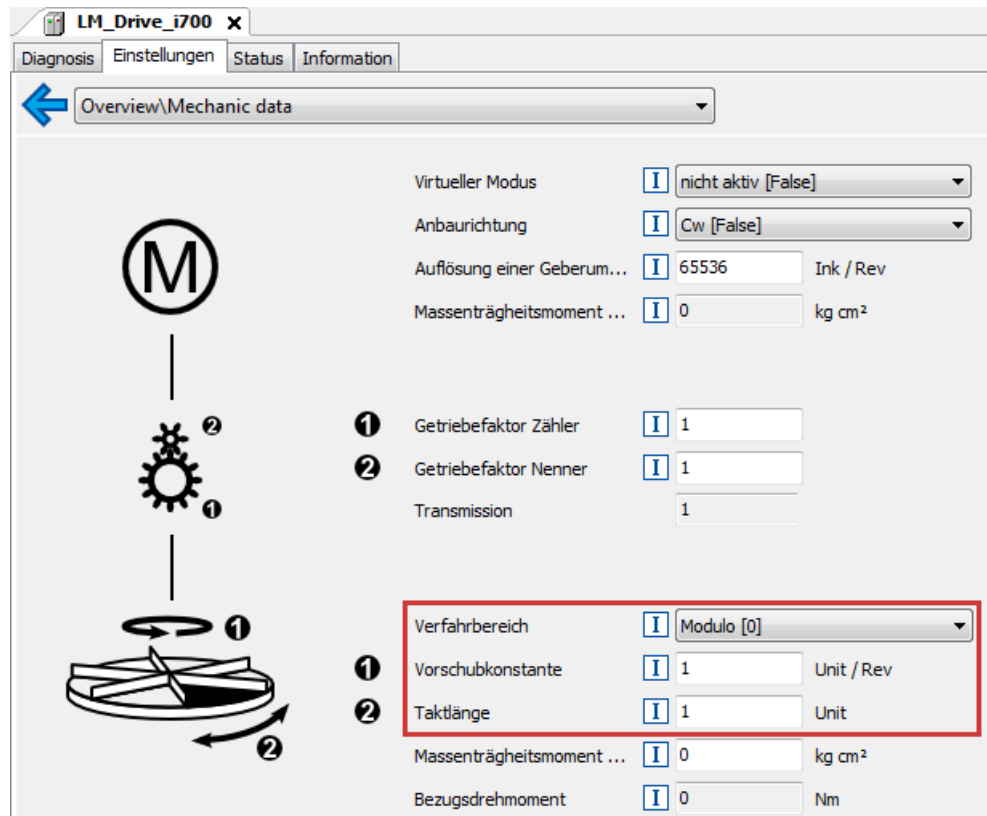
Hier finden Sie ausführliche Informationen zum Funktionsbaustein **L_MC1P_AxisBasicControlV2**.

3.3 Wichtige Hinweise zum Betrieb des Technologiemoduls

Das Technologiemodul ...

- unterstützt nicht den Simulationsmodus im »PLC Designer«;
- unterstützt keine virtuellen Achsen;
- unterstützt nur rotatorische Wicklerachsen.

Stellen Sie im »PLC Designer« für jede Achse unter der Registerkarte **Einstellungen** folgende Parameter ein:



Beachten Sie, dass die Geschwindigkeit der Linie in der Einheit [mm/s] vorgegeben wird.

Einstellung des Betriebsmodus

Der Betriebsmodus (Mode of Operation) für die Wicklerachse wird über das Technologiemodul vorgegeben. Der Wechsel zwischen "zyklisch synchroner Position (csp)", "zyklisch synchroner Geschwindigkeit (csv)" und "zyklisch synchronem Drehmoment (cst)" erfolgt dabei intern abhängig vom aktiven Status des Moduls.

Anlauf der Achsen

Der Anlauf der Achsen nach gesperrtem Regler hängt von der beauftragten Bewegungsart ab.

Unterschieden wird zwischen der Bewegungsart mit pegelaktivierter Funktion und der Bewegungsart mit flankenaktivierter Funktion.

- Die Bewegungsart mit pegelaktivierter Funktion wird bei Reglerfreigabe automatisch aktiv, solange keine höherpriorie Funktion aktiv ist. Zu dieser Bewegungsart gehören
 - [Handfahren \(Jogging\)](#) (📖 38)
 - [QSP Application](#) (📖 43)
- Die Bewegungsart mit flankenaktivierter Funktion muss nach Reglerfreigabe mit einer FALSE↗TRUE-Flanke aktiviert werden. Zu dieser Bewegungsart gehört
 - [Referenzfahrt \(Homing\)](#) (📖 40)



Beispiel zu einer Bewegungsart mit pegelaktivierter Funktion: Handfahren (Jogging)

1. Im gesperrten Achszustand wird der Eingang $xLogPos = TRUE$ gesetzt.
 - Eingang $xEnableOperation = FALSE$ (Achse ist gesperrt.)
==> Zustand "STANDBY" (Ausgang $xOperationEnabled = FALSE$)
 - Eingang $xLogPos = TRUE$ (Handfahren soll ausgeführt werden.)
2. Achse freigeben.
 - Eingang $xEnableOperation = TRUE$
==> Zustand "JOGGING" ($xOperationEnabled = TRUE$)



Beispiel zu einer Bewegungsart mit flankenaktivierter Funktion: Referenzfahrt (Homing)

1. Im gesperrten Achszustand wird der Eingang $xHomeExecute = TRUE$ gesetzt.
 - Eingang $xEnableOperation = FALSE$ (Achse ist gesperrt.)
==> Zustand "STANDBY" (Ausgang $xOperationEnabled = FALSE$)
 - Eingang $xHomeExecute = TRUE$ (Referenzierung soll ausgeführt werden.)
2. Achse freigeben.
 - Eingang $xEnableOperation = TRUE$
==> Zustand "STANDBY" (Ausgang $xOperationEnabled = TRUE$)
3. Referenzierung ausführen.
 - Eingang $xHomeExecute = FALSE↗TRUE$
==> Zustand "HOMING"

4 Anwendungsbereiche

Bei der Tänzerlageregelung wird die Bahnspannung ausschließlich durch eine dem Wickler vorgeschaltete Tänzereinrichtung erzeugt. Der Vorteil dieses Verfahrens liegt im Ausgleichsverhalten bei dynamischen Störgrößen, wie sie z. B. beim Beschleunigen oder bei Unwuchten im Wickelballen auftreten können. Dabei variiert dann zwar die Tänzerlage aber die Zugkraft bleibt konstant. Die Tänzerlageregelung wird vorwiegend dort eingesetzt, wo eine hohe absolute Zugkraftgenauigkeit und Zugkraftkonstanz gefordert ist.

Die Grenzen werden bei der Tänzerlagenregelung nicht durch den Drehmomentstellbereich, sondern in erster Linie durch die mechanische Ausführung des Tänzers, die Dynamik (Masse, Reibung usw.) und den Drehzahlstellbereich des Antriebs bestimmt.

Typische Anwendungen mit Bahngeschwindigkeiten bis zu 1000 m/min:

- Wickeln von Rundmaterial, wie z. B. Kabel oder Draht
- Textilien, Folien, Papier

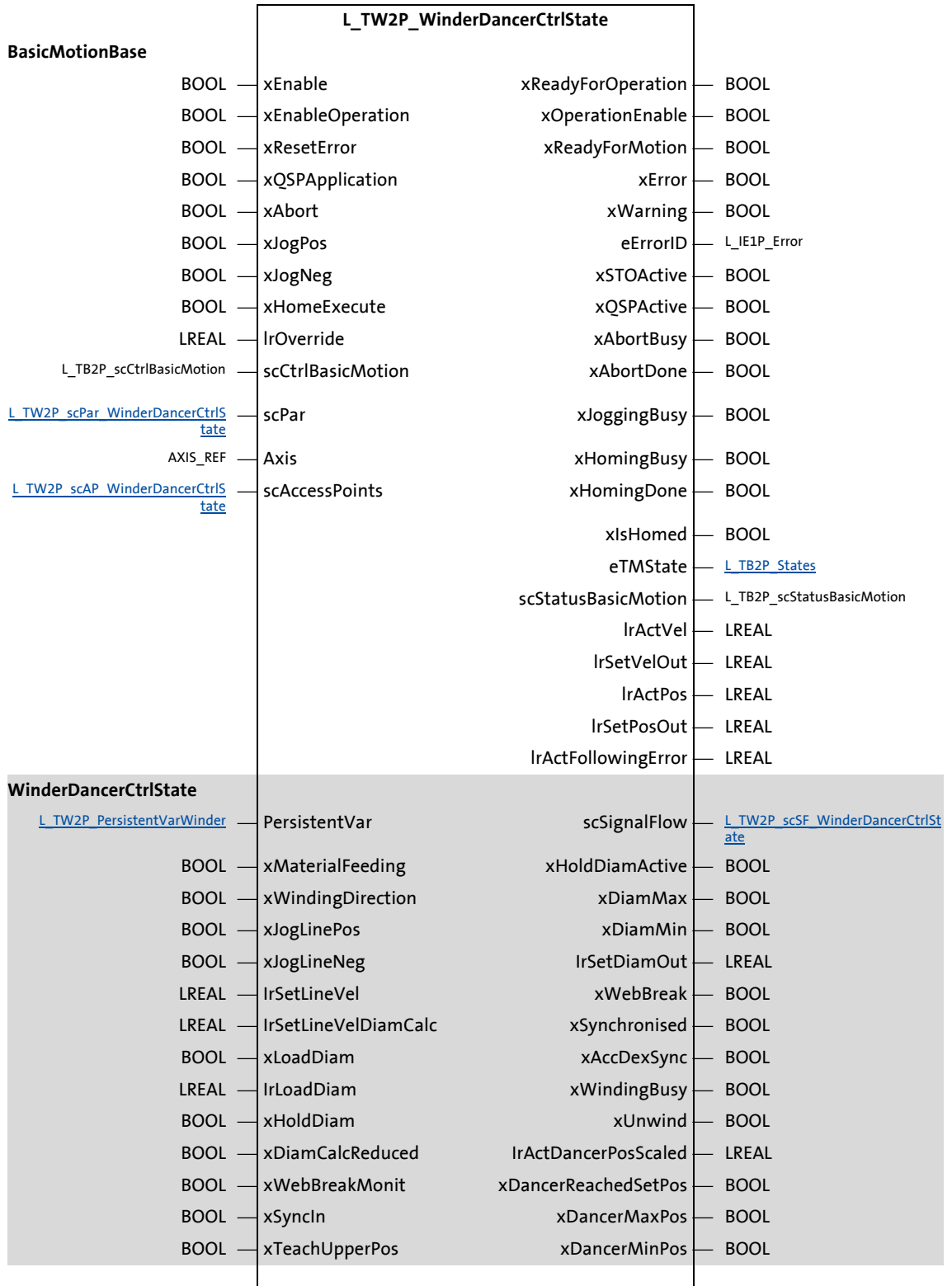
5

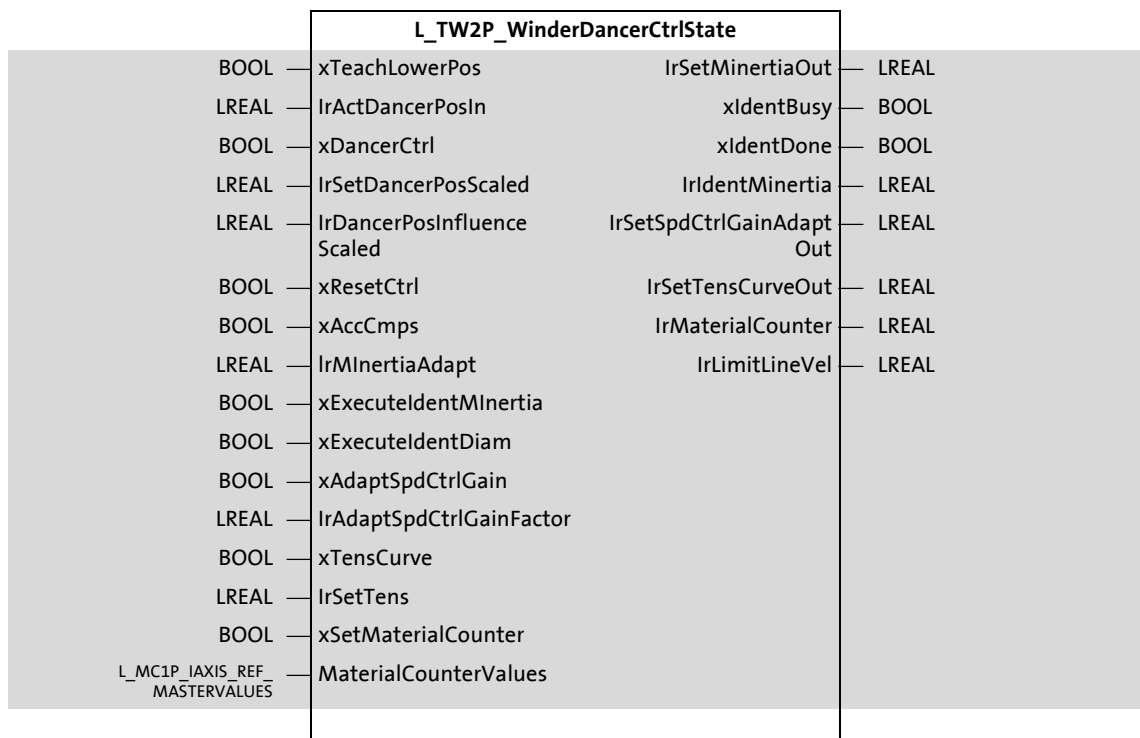
Funktionsbaustein L_TW2P_WinderDancerCtrlState

Die Abbildung zeigt die Zugehörigkeit der Ein- und Ausgänge des Funktionsbausteins.

Die zusätzlichen Ein- und Ausgänge der Variante "State" sind schattiert dargestellt.

► [L_TW2P_scSF_WinderDancerCtrlState](#) (26)





5.1

Eingänge und Ausgänge

Bezeichner	Datentyp	Beschreibung
PersistentVar L_TW2P_PersistentVarWinder		Referenz auf persistente Variablen (97)

5.2

Eingänge

Bezeichner	Datentyp	Beschreibung
xEnable	BOOL	Ausführung des Funktionsbausteins
		TRUE Der Funktionsbaustein wird ausgeführt.
		FALSE Der Funktionsbaustein wird nicht ausgeführt.
xEnableOperation	BOOL	TRUE Die Achse wird betriebsbereit geschaltet. Die Reglerfreigabe wird erteilt.
xResetError	BOOL	Der Eingang ist flankengesteuert und wertet die steigende Flanke aus.
		FALSE Fehler der Achse oder der Software zurücksetzen.
		TRUE
xQSPApplication	BOOL	Der Eingang ist pegelgesteuert und wertet den High-Pegel aus.
		TRUE Aktive Bewegung wird abgebrochen und Achse wird über die QSP Application (43) Stopprampe in den Stillstand geführt. • Ein Wechsel in den Zustand "STOP" erfolgt. • Das Technologiemodul bleibt im Zustand "STOP", solange <i>xQSPApplication</i> = TRUE gesetzt ist.
xAbort	BOOL	TRUE Aktive Bewegung wird abgebrochen und Achse wird über die aktive Profilverzögerung in den Stillstand geführt. • Ein Wechsel in den Zustand "STOP" erfolgt. Die Verzögerungsrampe kann durch eine Beauftragung einer neuen Bewegung (z. B. <i>xMoveVelExecute</i>) abgebrochen werden.
xJogPos	BOOL	Der Eingang ist pegelgesteuert und wertet den High-Pegel aus.
		TRUE Achse in positive Richtung fahren (Handfahren). Ist <i>xJogNeg</i> auch TRUE, wird über den Ausgang <i>eErrorID</i> die Fehlermeldung "InvalidInputJogPosJogNeg" ausgegeben und die Achse stoppt.
xJogNeg	BOOL	Der Eingang ist pegelgesteuert und wertet den High-Pegel aus.
		TRUE Achse in negative Richtung fahren (Handfahren). Ist <i>xJogPos</i> auch TRUE, wird über den Ausgang <i>eErrorID</i> die Fehlermeldung "InvalidInputJogPosJogNeg" ausgegeben und die Achse stoppt.
xHomeExecute	BOOL	Der Eingang ist flankengesteuert und wertet die steigende Flanke aus.
		FALSE Referenzierung starten.
		TRUE Der Abbruch der Funktion erfolgt über den Eingang <i>xHalt</i> .
IrOverride	LREAL	Mit dem Faktor <i>IrOverride</i> wird die vorgegebene Sollgeschwindigkeit multipliziert (z. B. ein Override-Wert von 0.5 erzeugt die Hälfte der Sollgeschwindigkeit) • Wertebereich: 0 ... 1.0 • Initialwert: 1.0 Hinweis: Die Übernahme des Wertes ist unabhängig vom Status in <i>xContinuousUpdate</i> .
scCtrlBasicMotion L_TB2P_scCtrlBasicMotion		Struktureingang zur Ansteuerung von Basic-Motion-Funktionen Siehe Dokumentation zum Technologiemodul "Basic Motion (V2)".
scPar L_TW2P_scPar_WinderDancerCtrlState		Die Parameterstruktur enthält die Parameter des Technologiemoduls.

Bezeichner	Datentyp	Beschreibung
Axis	AXIS_REF	Referenz auf die Achse
scAccessPoints L_TW2P_scAP_WinderDancerCtrlState		Struktur der Angriffspunkte Über die Angriffspunkte können der Signalfluss oder Funktionen des Technologiemodules beeinflusst werden. Der Datentyp ist abhängig von der verwendeten Variante (Base/State).
PersistentVar L_TW2P_PersistentVarWinder		Inhalt der persistenten Variablen innerhalb der Struktur L_TW2P_PersistentVarWinder
xMaterialFeeding	BOOL	Materialführung von oben oder unten an den Wickelballen • Initialwert: FALSE ⚠ WARNUNG! Personenschäden Verletzungsgefahr besteht durch Änderung der Einstellung am Eingang <i>xMaterialFeeding</i> während einer Bewegung. Dies kann zu einem Drehzahlsprung der Wicklerachse führen. Mögliche Folgen Tod oder schwere Verletzungen Schutzmaßnahmen Die Einstellung des Eingangs <i>xMaterialFeeding</i> nur ändern, wenn das Technologiemodul nicht ausgeführt wird und der Umrichter gesperrt ist: • Eingang <i>xEnable</i> = FALSE • Eingang <i>xEnableOperation</i> = FALSE
		TRUE Materialführung von oben
		FALSE Materialführung von unten
xWindingDirection	BOOL	Funktion des Wicklers bei positiver Liniengeschwindigkeit (<i>lrSetLineVel</i> > 0) • Initialwert: FALSE ⚠ WARNUNG! Personenschäden Verletzungsgefahr besteht durch Änderung der Einstellung am Eingang <i>xWindingDirection</i> während einer Bewegung. Dies kann zu einem Drehzahlsprung der Wicklerachse führen. Mögliche Folgen Tod oder schwere Verletzungen Schutzmaßnahmen Die Einstellung des Eingangs <i>xWindingDirection</i> nur ändern, wenn das Technologiemodul nicht ausgeführt wird und der Umrichter gesperrt ist: • Eingang <i>xEnable</i> = FALSE • Eingang <i>xEnableOperation</i> = FALSE
		TRUE Abwickler
		FALSE Aufwickler
xJogLinePos	BOOL	TRUE Achse in positive Materialflussrichtung fahren (Handfahren). Die Bahnparameter (Geschwindigkeit, Beschleunigung und Verzögerung) beziehen sich auf die Umfangsgeschwindigkeit des Wickelballens.
xJogLineNeg	BOOL	TRUE Achse in negative Materialflussrichtung fahren (Handfahren). Die Bahnparameter (Geschwindigkeit, Beschleunigung und Verzögerung) beziehen sich auf die Umfangsgeschwindigkeit des Wickelballens.
lrSetLineVel	LREAL	Aktuelle Liniengeschwindigkeit als Sollwert für den Wickler • Einheit: mm/s
xLoadDiam	BOOL	TRUE Den (Start-)Durchmesser aus dem Eingang <i>lrSetDiam</i> laden. • Initialwert: FALSE

Bezeichner	Datentyp	Beschreibung
IrLoadDiam	LREAL	Vorgabe eines (Start-)Durchmessers Der Durchmesser wird zyklisch geladen, wenn der Eingang <i>xLoadDiam</i> = TRUE gesetzt ist. • Einheit: mm • Initialwert: 0
xHoldDiam	BOOL	Aktuellen Durchmesser halten Initialwert: FALSE TRUE Der aktuelle Durchmesser wird gehalten. FALSE Der aktuelle Durchmesser wird nicht gehalten.
xDiamCalcReduced	BOOL	Modus-Umschaltung der Durchmesserberechnung zwischen langer/kurzer Distanz • Initialwert: FALSE TRUE Durchmesser wird nach kurzer Distanz aktualisiert. FALSE Durchmesser wird nach langer Distanz aktualisiert.
xWebBreakMonit	BOOL	Bahnrißüberwachung aktivieren/deaktivieren • Initialwert: FALSE TRUE Bahnrißüberwachung aktivieren. FALSE Bahnrißüberwachung deaktivieren.
xSyncln	BOOL	Synchronisierung der Wicklerachse auf die Liniengeschwindigkeit <i>IrSetLineVel</i> . Der Eingang ist pegelgesteuert und wertet einen High-Pegel aus. Freigabe der Achse über <i>xEnableOperation</i> erforderlich. Inbetriebnahmefunktion, um die Wickelwelle ohne Material der Liniengeschwindigkeit folgen zu lassen. TRUE Aufsynchronisieren über die Beschleunigung <i>IrSynclnAcc</i> . Folgen der Geschwindigkeit an <i>IrSetLineVel</i> . FALSE Keine synchrone Bewegung zu <i>IrSetLineVel</i> . Stoppen über die Verzögerung <i>IrSynclnDec</i> .
xTeachUpperPos	BOOL	TRUE Die aktuelle Tänzerposition wird als oberer Grenzwert gespeichert.
xTeachLowerPos	BOOL	TRUE Die aktuelle Tänzerposition wird als unterer Grenzwert gespeichert.
IrActDancerPosIn	LREAL	Aktuelle Tänzerposition Die Istposition des Tänzers wird in Form eines analogen Signals (0 ... 10 V) an den Controller zurückgeführt.
xDancerCtrl	BOOL	TRUE Tänzerlageregelung aktivieren. TRUE↘ FALSE Tänzerlageregelung wird deaktiviert. Mit dem definierten Parameter <i>eDancerCtrlStopMode</i> wird das Wickeln beendet.
IrSetDancerPosScaled	LREAL	Skalierter Sollwert für die Tänzerposition • Einheit: x 100 % • Initialwert: 0 • Wertebereich: -1 ... 1 (-100 ... 100 %)
IrDancerPosInfluenceScaled	LREAL	Einflussbereich des Tänzerlagereglers • Einheit: x 100 % • Initialwert: 0 • Wertebereich: 0 ... 1 (0 ... 100 %)
xResetICtrl	BOOL	TRUE Der I-Anteil des PI-Reglers kann ausgeschaltet werden und die Stellgröße (Ausgang des Reglers) aus dem I-Anteil kann über die Rampenfunktion auf den Wert 0 geführt werden. Die Stellgröße aus dem P-Anteil wird nicht beeinflusst.
xAccCmps	BOOL	Beschleunigungskompensation während der Tänzerlageregelung aktivieren/deaktivieren • Initialwert: FALSE

Bezeichner	Datentyp	Beschreibung
IrMinertiaAdapt	LREAL	Multiplikator zum aktuellen Massenträgheitsmoment • Initialwert: 0
xExecutIdentMinertia	BOOL	Der Eingang ist flankengesteuert und wertet die steigende Flanke aus.
	FALSE	Das Massenträgheitsmoment an der Wickelwelle wird ermittelt.
	TRUE	Am Ausgang IrIdentMinertia wird das ermittelte Massenträgheitsmoment in kgcm ² angezeigt.
xExecutIdentDiam	BOOL	Der Eingang ist flankengesteuert und wertet die steigende Flanke aus.
	FALSE	Die Identifikation des Durchmessers wird gestartet.
	TRUE	Der Tänzer wird bei stehender Liniengeschwindigkeit angehoben. Durch den Tänzerhub wird der Wickeldurchmesser aus den ermittelten Daten von zurückgelegter Materiallänge und Drehwinkel errechnet.
xAdaptSpdCtrlGain	BOOL	Adaption der Drehzahlreglerverstärkung ein-/ausschalten. • Initialwert: FALSE
	TRUE	Adaption der Drehzahlreglerverstärkung freigegeben.
	FALSE	Adaption der Drehzahlreglerverstärkung inaktiv.
IrAdaptSpdCtrlGainFactor	LREAL	Der resultierende Wert der Adaption der Drehzahlreglerverstärkung kann über diesen Eingang multiplikativ beeinflusst werden. • Wertebereich: 0 ... 1 • Initialwert: 1
xTensCurve	BOOL	Zugkraftkennlinie freigeben Initialwert: FALSE
	TRUE	Zugkraftkennlinie freigeben.
	FALSE	Zugkraftkennlinie sperren.
IrSetTens	LREAL	Zugkraftsollwert • Einheit: N • Initialwert: 0
xSetMaterialCounter	BOOL	Der Eingang ist flankengesteuert und wertet die FALSE→TRUE-Flanke aus.
	TRUE	Setzt den Materiallängenzähler (Ausgang IrMaterialCounter) auf den Wert, der unter dem Parameter IrSetMaterialPos eingestellt ist.
MaterialCounterValues L_MC1P_IAXIS_REF_MASTERVERVALUES		Hier kann eine Modulo-Achse eines Messrades auf dem Material angeschlossen werden. Wenn eine Achse am Eingang angeschlossen ist, so erfolgt die Ermittlung der Materiallänge anhand der Daten aus der Referenzachse. Dieses Verfahren ist auch für verrauschte Signale geeignet. Falls hier keine Achse angeschlossen ist, erfolgt die Ermittlung der Materiallänge aus der Integration der Materialgeschwindigkeit (Eingang IrSetLineVel oder IrSetLineVelDiamCalc). ▶ Materiallängenzähler (75) ▶ Quellen für die Materiallängenzählung (75) Unterstützte Achsen: • Encoder • Servoumrichter 9400 HighLine • Servoumrichter i700 • Servoumrichter i950

5.3

Ausgänge

Bezeichner	Datentyp	Beschreibung	
xReadyForOperation	BOOL	Die Achse ist fehlerfrei. Die Zwischenkreisspannung ist vorhanden. STO ist nicht aktiv.	
xOperationEnabled	BOOL	Die Achse ist freigegeben. Die Bremse ist gelüftet.	
xReadyForMotion	BOOL	Die Achse ist freigegeben und fehlerfrei. Die Bremse ist gelüftet.	
xError	BOOL	TRUE	Im Technologiemodul liegt ein Fehler vor.
xWarning	BOOL	TRUE	Im Technologiemodul liegt eine Warnung vor.
eErrorID	L_IE1P_Error	ID der Fehler- oder Warnungsmeldung, wenn <i>xError</i> = TRUE oder <i>xWarning</i> = TRUE ist. Referenzhandbuch "FAST & Technologiemodule": Hier finden Sie Informationen zu Fehler- oder Warnungsmeldungen.	
xSTOActive	BOOL	TRUE	Ein Safe Torque Off (STO) wurde ausgelöst und ist aktiv. Die Achse ist drehmomentfrei, der Regler ist gesperrt.
xQSPActive	BOOL	TRUE	Der Eingang <i>xQSPApplication</i> wurde gesetzt. Das Technologiemodul befindet sich im Zustand "STOP", die Achse wird in den Stillstand geführt. Der Ausgang wird zurückgesetzt, sobald sich die Achse im Stillstand befindet.
xAbortBusy	BOOL	TRUE	Die aktivierte Funktion wird durch den Eingang <i>xAbsort</i> abgebrochen.
xAbsortDone	BOOL	TRUE	Die aktivierte Funktion ist durch den Eingang <i>xAbsort</i> abgebrochen worden.
xJoggingBusy	BOOL	TRUE	Die Funktion Handfahren ist aktiv. Das Technologiemodul befindet sich im Zustand "JOGGING".
xHomingBusy	BOOL	TRUE	Die Referenzfahrt wird ausgeführt. Das Technologiemodul befindet sich im Zustand "HOMING".
xHomingDone	BOOL	TRUE	Die Referenzfahrt ist erfolgreich abgeschlossen.
xIsHomed	BOOL	TRUE	Die Achse ist referenziert ("Referenz bekannt").
eTMState		Aktueller Zustand des Technologiemoduls ► State Machine (32)	
scStatusBasicMotion	L_TB2P_scStatusBasicMotion	Struktur der Zustandsdaten Siehe Dokumentation zum Technologiemodul "Basic Motion (V2)".	
IrActVel	LREAL	Aktuelle Geschwindigkeit • Einheit: units/s	
IrSetVelOut	LREAL	Sollgeschwindigkeit • Einheit: units/s	
IrActPos	LREAL	Aktuelle Position • Einheit: units	
IrSetPosOut	LREAL	Sollposition • Einheit: units	
IrActFollowingError	LREAL	Aktueller Schleppabstand • Einheit: units	

Bezeichner	Datentyp	Beschreibung	
xHoldDiamActive	BOOL	Status über Berechnung des Durchmessers. Hinweis: Beim Toggeln dieses Ausgangs wird empfohlen, die Hysteresefunktion zu nutzen (Parameter <i>IrHystlineStandstill</i>). ► Signalflussplan (33)	
		TRUE	Der aktuelle Durchmesser wird gehalten.
		FALSE	Der aktuelle Durchmesser wird aktualisiert.
xDiamMax	BOOL	TRUE	Der maximale Durchmesser wurde erreicht.
xDiamMin	BOOL	TRUE	Der minimale Durchmesser wurde erreicht.
IrSetDiamOut	LREAL	Aktueller berechneter Durchmesser • Einheit: mm	
xWebBreak	BOOL	TRUE	Ein Bahnriß liegt vor.
xSynchronised	BOOL	TRUE	Der Wickler ist auf die Liniengeschwindigkeit synchronisiert.
xAccDecSync	BOOL	TRUE	Die Synchronisierungsfunktion ist aktiv. Der Wickler wird auf- oder asynchronisiert.
xWindingBusy	BOOL	TRUE	Der Wickelbetrieb ist aktiviert.
xUnwind	BOOL	Statusbit für Auf- und Abwickler Hinweis: Beim Toggeln dieses Ausgangs wird empfohlen, die Hysteresefunktion zu nutzen (Parameter <i>IrHystlineStandstill</i>). ► Signalflussplan (33)	
		TRUE	Abwickler
		FALSE	Aufwickler
IrActDancerPosScaled	LREAL	Aktuelle skalierte Tänzerposition • Einheit: x 100 % • Wertebereich: -1 ... 1 (-100 ... 100 %)	
xDancerReachedSetPos	BOOL	TRUE	Der Tänzer hat die Sollposition erreicht und befindet sich innerhalb des Toleranzfensters in Parameter <i>IrDancerInPosWindowScaled</i> .
xDancerMaxPos	BOOL	TRUE	Der Tänzer hat die obere Grenzposition erreicht.
xDancerMinPos	BOOL	TRUE	Der Tänzer hat die untere Grenzposition erreicht.
IrSetMIInertiaOut	LREAL	Aktuelles Massenträgheitsmoment an der Wickelwelle • Einheit: kgcm ²	
xIdentBusy	BOOL	TRUE	Die Identifikation der Massenträgheit oder der Reibung wird ermittelt.
xIdentDone	BOOL	TRUE	Die Identifikation der Massenträgheit oder der Reibung ist abgeschlossen.
IrIdentMIInertia	LREAL	Identifiziertes Massenträgheitsmoment an der Wickelwelle • Einheit: kgcm ²	
IrSetSpdCtrlGainAdaptOut	LREAL	Adaption der Drehzahlreglerverstärkung • Einheit: x 100 % (1 = 100 %)	
IrSetTensCurveOut	LREAL	Aktueller Zugkraftsollwert aus der Kurvenfunktion	

Bezeichner	Datentyp	Beschreibung
IrMaterialCounter	LREAL	Anzeige des Materiallängenzählerstandes auf dem Wickler Je nach Festlegung der Wickelrichtung wird der Materiallängenzähler hoch- oder runtergezählt • Einheit: mm ► Festlegung der Wickelrichtung (Aufwickeln/Abwickeln) (📖 58).
IrLimitLineVel	LREAL	Die maximal erlaubte Liniengeschwindigkeit wird aus dem berechnetem Durchmesser und dem Parameter <i>IrMaxWndSpd</i> bestimmt. • Einheit [mm/s]

5.4

Parameter

L_TW2P_scPar_WinderDancerCtrlState

Bezeichner	Datentyp	Beschreibung
IrVel	LREAL	Geschwindigkeit für Bewegungen, die über die Struktur scCtrlBasicMotion beauftragt werden. • Einheit: units/s • Initialwert: 3600
IrAcc	LREAL	Beschleunigung für Bewegungen, die über die Struktur scCtrlBasicMotion beauftragt werden. Vorgabe, mit welcher Geschwindigkeitsänderung maximal beschleunigt werden soll. • Einheit: units/s ² • Initialwert: 36000
IrDec	LREAL	Verzögerung für Bewegungen, die über die Struktur scCtrlBasicMotion beauftragt werden. Vorgabe, mit welcher Geschwindigkeitsänderung maximal bis zum Stillstand verzögert werden soll. • Einheit: units/s ² • Initialwert: 100
IrJerk	LREAL	Ruck für Bewegungen, die über die Struktur scCtrlBasicMotion beauftragt werden. • Einheit: units/s ³ • Initialwert: 0
eDirection	MC_DIRECTION	Verfahrrichtung: • <i>mcCurrentDirection</i> : Aktuelle Richtung beibehalten (nur bei Modulo-Achse) • <i>mcPositiveDirection</i> : Positive Richtung • <i>mcNegativeDirection</i> : Negative Richtung • <i>mcShortestWay</i> : Kürzester Weg (nur bei Modulo-Achse) • Initialwert: <i>mcPositiveDirection</i>
scHomeExtTP	MC_TRIGGER_REF	Übergabe eines externen Touch-Probe-Ereignisses: • Nur relevant bei der Touch-Probe-Konfiguration "Externer Geber". • Zur Beschreibung der Struktur MC_TRIGGER_REF siehe Funktionsbaustein MC_TouchProbe .
IrLineJerk	LREAL	Ruck für den Tippbetrieb mit der 1/d-Bewertung und zum Ausgleich bei der Kupplungsfunktion • Einheit: rev/s ³ • Initialwert: 100000
IrJogLineAcc	LREAL	Beschleunigung für den Tippbetrieb mit der 1/d-Bewertung. Vorgabe, mit welcher Geschwindigkeitsänderung maximal beschleunigt werden soll. • Einheit: mm/s ² • Initialwert: 100
IrJogLineDec	LREAL	Verzögerung für den Tippbetrieb mit der 1/d-Bewertung. Vorgabe, mit welcher Geschwindigkeitsänderung maximal verzögert werden soll. • Einheit: mm/s ² • Initialwert: 10
IrJogLineVel	LREAL	Geschwindigkeit für den Tippbetrieb mit der 1/d-Bewertung. Vorgabe, mit welcher Geschwindigkeit maximal verfahren werden soll. • Einheit: mm/s • Initialwert: 10
IrSyncLineAcc	LREAL	Beschleunigung zur Synchronisierung auf die Liniengeschwindigkeit • Einheit: mm/s ² • Initialwert: 100
IrSyncLineDec	LREAL	Verzögerung zur Synchronisierung auf die Liniengeschwindigkeit • Einheit: mm/s ² • Initialwert: 100

Bezeichner	Datentyp	Beschreibung
IrWebBreakWindow	LREAL	Bahnrissofenster Der aktuelle Durchmesser wird mit dem Durchmesser aus dem letzten Berechnungszyklus über das Bahnrissofenster verglichen. • Einheit: x 100 % (1.0 = 100 %) • Initialwert: 0.1 (10 %)
IrMaxDiam	LREAL	Maximaler Durchmesser • Einheit: mm • Initialwert: 180
IrMinDiam	LREAL	Minimaler Durchmesser • Einheit: mm • Initialwert: 50
rFiltTimeDiam	REAL	PT1-Filterzeit für den aktuellen Durchmesser (<i>IrSetDiamOut</i>) • Einheit: s • Initialwert: 0.05
IrDiamCalcRegularDist	LREAL	Reguläre Berechnungsdistanz für Durchmesser • Einheit: rev • Initialwert: 1
IrDiamCalcReducedDist	LREAL	Verkürzte Berechnungsdistanz für Durchmesser • Einheit: rev • Initialwert: 0.1
alrAdaptDiamX ARRAY [1...9] OF LREAL		Stützpunkte der Kurvenfunktion für das Laden des Durchmessers • Werte, die am analogen Eingang <i>IrSetDiam</i> anliegen können. • Einheit: mm • Initialwerte: 0, 100, 200, 300, 400, 500, 600, 700, 800
alrAdaptDiamY ARRAY [1...9] OF LREAL		Stützpunkte der Kurvenfunktion für das Laden des Durchmessers • Funktionswerte für den Durchmesser • Einheit: mm • Initialwerte: 0, 100, 200, 300, 400, 500, 600, 700, 800
IrTensCurveCtrlScaled	LREAL	Steigung der Kennlinie für die Zugkraftsteuerung • Einheit: x 100 % (1 = 100 %) • Initialwert: 0 • Mit dem Wert '1' ergibt sich ein konstanter Zugkraftverlauf und damit ein für den Durchmesser proportional ansteigender Sollwert.
IrTensCurveStartDiamScaled	LREAL	Anfangspunkt der Kennlinie für die Zugkraftsteuerung • Einheit: x 100 % • 1 = 100 % = Parameter <i>IrMaxDiam</i> • Initialwert: 0 (0 %)
IrLineVelRef	LREAL	Maximale Liniengeschwindigkeit • Einheit: mm/s • Initialwert: 1000
IrMinLineVel	LREAL	Minimale Liniengeschwindigkeit Bis zu dieser Geschwindigkeit wird der Durchmesser gehalten. • Einheit: mm/s • Initialwert: 1
rFiltTimeMaterialCounter	REAL	Filterzeitkonstante für den Materiallängenzähler (Ausgang <i>IrMaterialCounter</i>) • Initialwert: 0 (Filter ist deaktiviert.)
IrSetMaterialPos	LREAL	Position des Materiallängenzählers Mit einer FALSE/TRUE-Flanke am Eingang <i>xSetMaterialCounter</i> wird der Materiallängenzähler (Ausgang <i>IrMaterialCounter</i>) auf den Wert in <i>IrSetMaterialPos</i> gesetzt. • Einheit: mm • Initialwert: 0
IrMaxWndSpd	LREAL	Festlegung der maximalen Wickelwellen-Drehzahl (abtriebsseitig). • Einheit: rev/s • Initialwert: 100

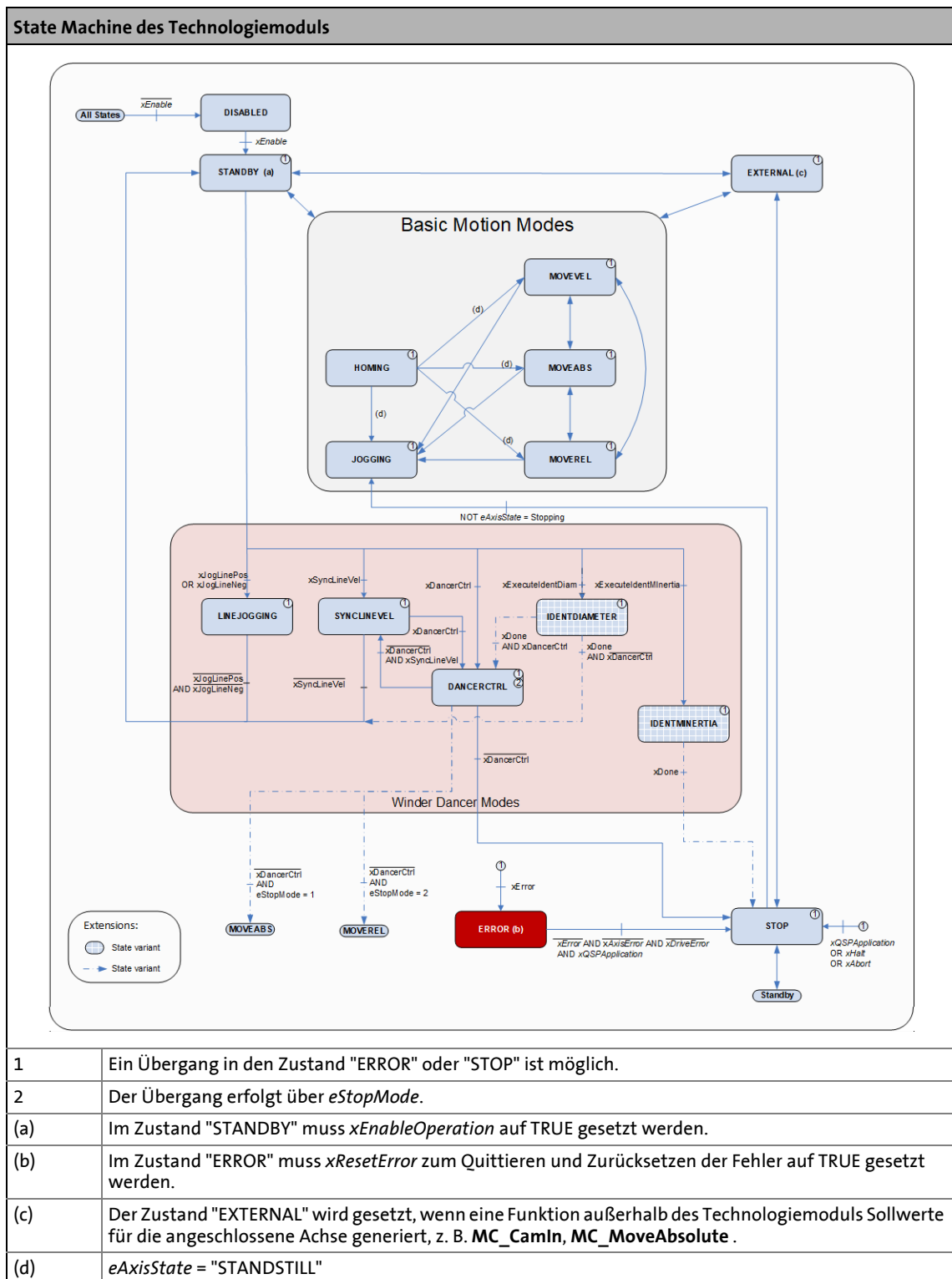
Bezeichner	Datentyp	Beschreibung
xLineVelDiamCalc	BOOL	Berechnung des Durchmessers Initialwert: FALSE
		TRUE Für die Berechnung des Durchmessers wird die Geschwindigkeit aus dem Eingang <i>IrSetLineVelDiamCalc</i> verwendet.
		FALSE Für die Berechnung des Durchmessers wird die Geschwindigkeit aus dem Eingang <i>IrSetLineVel</i> verwendet.
IrDancerPosRamp	LREAL	Beschleunigungsrampe für die Tänzerlagesollwerte - Einheit: 1/s - Initialwert: 1
IrDancerPosCtrlGain	LREAL	Reglerverstärkung Initialwert: 1
IrDancerPosCtrlResetTime	LREAL	Reglernachstellzeit - Einheit: s - Initialwert: 0 (Die Regler-Nachstellzeit ist deaktiviert.)
eResetCtrlMode L_TW2P_ResetCtrlMode		Modus zur Einstellung der Funktion für das Rücksetzen der Stellgröße aus dem I-Anteil über den Eingang <i>xResetCtrl</i> = TRUE Initialwert: 1
		0 Instant Mit steigender Flanke <i>xResetCtrl</i> = TRUE wird sofort (sprungartig) die Stellgröße aus dem I-Anteil = 0 gesetzt.
		1 Ramp Mit steigender Flanke <i>xResetCtrl</i> = TRUE wird die Stellgröße aus dem I-Anteil bis zu Null über den Parameter <i>IrDancerPosCtrlResetTime</i> geführt.
IrDancerPosCtrlLimPos	LREAL	Begrenzung der Tänzerlageregler-Stellgröße (Ausgang des Reglers) in positive Richtung Initialwert: 1
IrDancerPosCtrlLimNeg	LREAL	Begrenzung der Tänzerlageregler-Stellgröße (Ausgang des Reglers) in negative Richtung Initialwert: -1
IrDancerMaxPosScaled	LREAL	Maximale Tänzerposition für das Statusbit <i>xDancerMaxPos</i> - Einheit: x 100 % (1 = 100 %) - Initialwert: 0.95 (95 %)
IrDancerMinPosScaled	LREAL	Minimale Tänzerposition für das Statusbit <i>xDancerMinPos</i> - Einheit: x 100 % (1 = 100 %) - Initialwert: -0.95 (-95 %)
xTeachDancerLimits	BOOL	Quelle der Tänzerlagebegrenzungen Initialwert: FALSE
		TRUE Die Endlagen <i>IrDancerLowerLimit</i> und <i>IrDancerUpperLimit</i> werden solange verwendet, bis die Teaching-Funktion ausgeführt wurde. Nach manueller Ausführung der Teaching-Funktion werden immer die gespeicherten Endlagen aus der Teaching-Funktion verwendet.
		FALSE Die Endlagen <i>IrDancerLowerLimit</i> und <i>IrDancerUpperLimit</i> werden verwendet.
xUpperPosTeached	BOOL	Referenzierte obere Tänzerendlage
		TRUE Die obere Tänzerendlage bereits referenziert.
xLowerPosTeached	BOOL	Referenzierte obere Tänzerendlage
		TRUE Die untere Tänzerendlage bereits referenziert.
IrDancerUpperLimit	LREAL	Analoger Wert für die obere Tänzerlagegrenze Initialwert: 10 000 000
IrDancerLowerLimit	LREAL	Analoger Wert für die untere Tänzerlagegrenze Initialwert: 0

Bezeichner	Datentyp	Beschreibung
IrDancerMaterialLength	LREAL	Länge des Materials im Tänzer - Einheit: mm - Initialwert: 0 - Mit dem Wert 0 wird die Betrachtung der Tänzerbewegung deaktiviert.
IrDancerInPosWindowScaled	LREAL	Fenster für die Sollposition des Tänzers, um das Statusbit <i>xDancerReachedSetPos</i> anzusteuern. - Einheit: x 100 % (1.0 = 100 %) - Initialwert: 0.2 (20 %)
rFiltTimeActDancerPosIn	REAL	PT1-Filterzeit für den Eingang <i>IrActDancerPosIn</i> - Einheit: s - Initialwert: 0.005
rFiltTimeActDancerVelComp	REAL	Filterzeit für die Geschwindigkeitskompensation - Einheit: s - Initialwert: 0
wWebBreakMode	WORD	Modus für die Bahnrißüberwachung Initialwert: 1
		0 Bahnrißüberwachung aus der Durchmesserberechnung und der Tänzerlage
		1 Bahnrißüberwachung nur aus der Tänzerlage
		2 Bahnrißüberwachung nur aus der Durchmesserberechnung
dwSelectTensCurve	DWORD	Auswahl der Kennlinie für die Zugkraftsteuerung Initialwert: 0
		0 Linearer Zugkraftverlauf
		1 Linearer Drehmomentverlauf
		2 Zugkraftverlauf nach vorgegebener Kennlinie
alrTensCurve	ARRAY [1...65] OF LREAL	Kennlinie für die Zugkraftsteuerung bestehend aus 65 Werten. Hier müssen alle 65 Werte, gleichmäßig verteilt über den Durchmesserbereich der Parameter <i>IrMinDiam</i> und <i>IrMaxDiam</i> (minimaler und maximaler Durchmesser) angegeben werden.
rFiltTimeAccSpd	REAL	PT1-Filterzeit für die Drehzahl der Wickelwelle zur Beschleunigungskompensation - Einheit: s - Initialwert: 0.005
IrAccCmpsDeadBandTrq Scaled	LREAL	Nacheilbereich (Dead Band) für das aktuelle Beschleunigungsmoment - Einheit: x 100 % (1.00 = 100 %); Bezugsgröße : Motor-Bemessungsdrehmoment [Nm] - Initialwert: 0.1 (10 %)
IrAccCmpsGainAcc	LREAL	Verstärkungsfaktor für das Beschleunigungsmoment in positive Richtung - Einheit: x 100 % (1.00 = 100 %) - Wertebereich: 0 ... 2 (0 ... 200 %) - Initialwert: 1.05 (105 %)
IrAccCmpsGainDec	LREAL	Verstärkungsfaktor für das Beschleunigungsmoment in negative Richtung - Einheit: x 100 % (1.00 = 100 %) - Wertebereich: 0 ... 2 (0 ... 200 %) - Initialwert: 0.95 (95 %)
IrConstMIInertia	LREAL	Konstantes Massenträgheitsmoment an der Wickelwelle - Einheit: kgcm² - Initialwert: 1
IrMaxMIInertia	LREAL	Maximal zulässiges Massenträgheitsmoment an der Wickelwelle - Einheit: kgcm² - Initialwert: 10
rFiltTimeIdentMIInertiaSpd	REAL	PT1-Filterzeit für die Drehzahl der Wickelwelle während der Identifikation des Massenträgheitsmoments - Einheit: s - Initialwert: 0.01

Bezeichner	Datentyp	Beschreibung						
rFiltTimIdentMInertiaTrq	REAL	PT1-Filterzeit für das Drehmoment der Wickelwelle während der Identifikation des Massenträgheitsmoments - Einheit: s - Initialwert: 0.005						
IrIdentMInertiaMaxSpd Scaled	LREAL	Maximale Drehzahl der Wickelwelle während der Massenträgheitsidentifikation - Einheit: x 100 % (1.0 = 100 % = <i>IrWndSpdRef</i>) - Wertebereich: 0 ... 1 (0 ... 100 %) - Initialwert: 0.2 (20 %)						
IrIdentMInertiaMaxTrq Scaled	LREAL	Maximales Drehmoment der Wickelwelle während der Massenträgheitsidentifikation - Einheit: x 100 % (1.0 = 100 %) - Bezugsgröße: Motor-Bemessungsdrehmoment - Wertebereich: 0 ... 1 (0 ... 100 %) - Initialwert: 0.2 (20 %)						
eAdaptSpdCtrlGainMode	ENUM	Modusauswahl zur Adaption der Drehzahlreglerverstärkung. Initialwert: 2 <table><tr><td>0</td><td>DiamToSquare; $VP = f(d^2)$</td></tr><tr><td>1</td><td>Diam; $VP = f(d)$</td></tr><tr><td>2</td><td>Inertia; $VP = f(J)$</td></tr></table>	0	DiamToSquare; $VP = f(d^2)$	1	Diam; $VP = f(d)$	2	Inertia; $VP = f(J)$
0	DiamToSquare; $VP = f(d^2)$							
1	Diam; $VP = f(d)$							
2	Inertia; $VP = f(J)$							
IrAdaptSpdCtrlLowLimit	LREAL	Untere Begrenzung der Drehzahlreglerverstärkung im Antrieb. Der Adaptionswert <i>IrSetSpdCtrlGainAdaptOut</i> darf nicht kleiner sein als der Wert in Parameter <i>IrAdaptSpdCtrlLowLimit</i> . - Wertebereich: 0 bis 1 - Initialwert: 1.0 (100 %)						
alrSpdCtrlGainAdaptX	ARRAY [1...9] OF LREAL	Kennlinienfunktion für die Drehzahlregelungsverstärkung Die X-Achse entspricht dem aktuell berechneten Adaptionsfaktor für die Anpassung der Drehzahlreglerverstärkung. - Einheit: x 100 % (1 = 100 % = Parameter <i>IrMaxMInertia</i>) - Initialwerte: [0.0, 0.2, 0.4, 0.6, 0.8, 1.0, 1.2, 1.4, 1.6]						
alrSpdCtrlGainAdaptY	ARRAY [1...9] OF LREAL	Kennlinienfunktion für die Drehzahlregelungsverstärkung Die Y-Achse entspricht dem Verstärkungsfaktor des Drehzahlreglers. - Einheit: x 100 % (1 = 100 %) - Initialwerte: [0.0, 0.2, 0.4, 0.6, 0.8, 1.0, 1.2, 1.4, 1.6]						
IrReducedGainWindow	LREAL	Bereich der Regelabweichung mit reduzierter Verstärkung/Empfindlichkeit. Mit <i>IrReducedGainWindow</i> wird der Toleranzbereich der Regelabweichung festgelegt, in dem <i>IrReducedGain</i> wirkt. Initialwert: 0.0						
IrReducedGain	LREAL	Verstärkung der Regelabweichung im Bereich der reduzierten Empfindlichkeit Initialwert: 0.0						
eDancerCtrlStopMode	L_TW2P_DancerCtrlStopMode	Modus, in dem der Wickelprozess (Zustand "DANCERCTRL") beendet wird. Initialwert: 0 <table><tr><td>0</td><td>Halt Die Achse wird über die Verzögerung (<i>IrHaltDec</i>) und den Ruck (<i>IrJerk</i>) angehalten.</td></tr><tr><td>1</td><td>Move Abs Absolute Fahrt/Positionierung: Die Achse wird mit der Geschwindigkeit (<i>IrVel</i>), Beschleunigung (<i>IrAcc</i>), Verzögerung (<i>IrDec</i>) und den Ruck (<i>IrJerk</i>) in die Zielposition (<i>IrPos_Dist</i>) gefahren.</td></tr><tr><td>2</td><td>Move Rel Relative Fahrt/Positionierung: Die Achse wird mit der Geschwindigkeit (<i>IrVel</i>), Beschleunigung (<i>IrAcc</i>), Verzögerung (<i>IrDec</i>) und den Ruck (<i>IrJerk</i>) nach der gefahrenen Wegstrecke (<i>IrPos_Dist</i>) in den Stillstand geführt.</td></tr></table>	0	Halt Die Achse wird über die Verzögerung (<i>IrHaltDec</i>) und den Ruck (<i>IrJerk</i>) angehalten.	1	Move Abs Absolute Fahrt/Positionierung: Die Achse wird mit der Geschwindigkeit (<i>IrVel</i>), Beschleunigung (<i>IrAcc</i>), Verzögerung (<i>IrDec</i>) und den Ruck (<i>IrJerk</i>) in die Zielposition (<i>IrPos_Dist</i>) gefahren.	2	Move Rel Relative Fahrt/Positionierung: Die Achse wird mit der Geschwindigkeit (<i>IrVel</i>), Beschleunigung (<i>IrAcc</i>), Verzögerung (<i>IrDec</i>) und den Ruck (<i>IrJerk</i>) nach der gefahrenen Wegstrecke (<i>IrPos_Dist</i>) in den Stillstand geführt.
0	Halt Die Achse wird über die Verzögerung (<i>IrHaltDec</i>) und den Ruck (<i>IrJerk</i>) angehalten.							
1	Move Abs Absolute Fahrt/Positionierung: Die Achse wird mit der Geschwindigkeit (<i>IrVel</i>), Beschleunigung (<i>IrAcc</i>), Verzögerung (<i>IrDec</i>) und den Ruck (<i>IrJerk</i>) in die Zielposition (<i>IrPos_Dist</i>) gefahren.							
2	Move Rel Relative Fahrt/Positionierung: Die Achse wird mit der Geschwindigkeit (<i>IrVel</i>), Beschleunigung (<i>IrAcc</i>), Verzögerung (<i>IrDec</i>) und den Ruck (<i>IrJerk</i>) nach der gefahrenen Wegstrecke (<i>IrPos_Dist</i>) in den Stillstand geführt.							

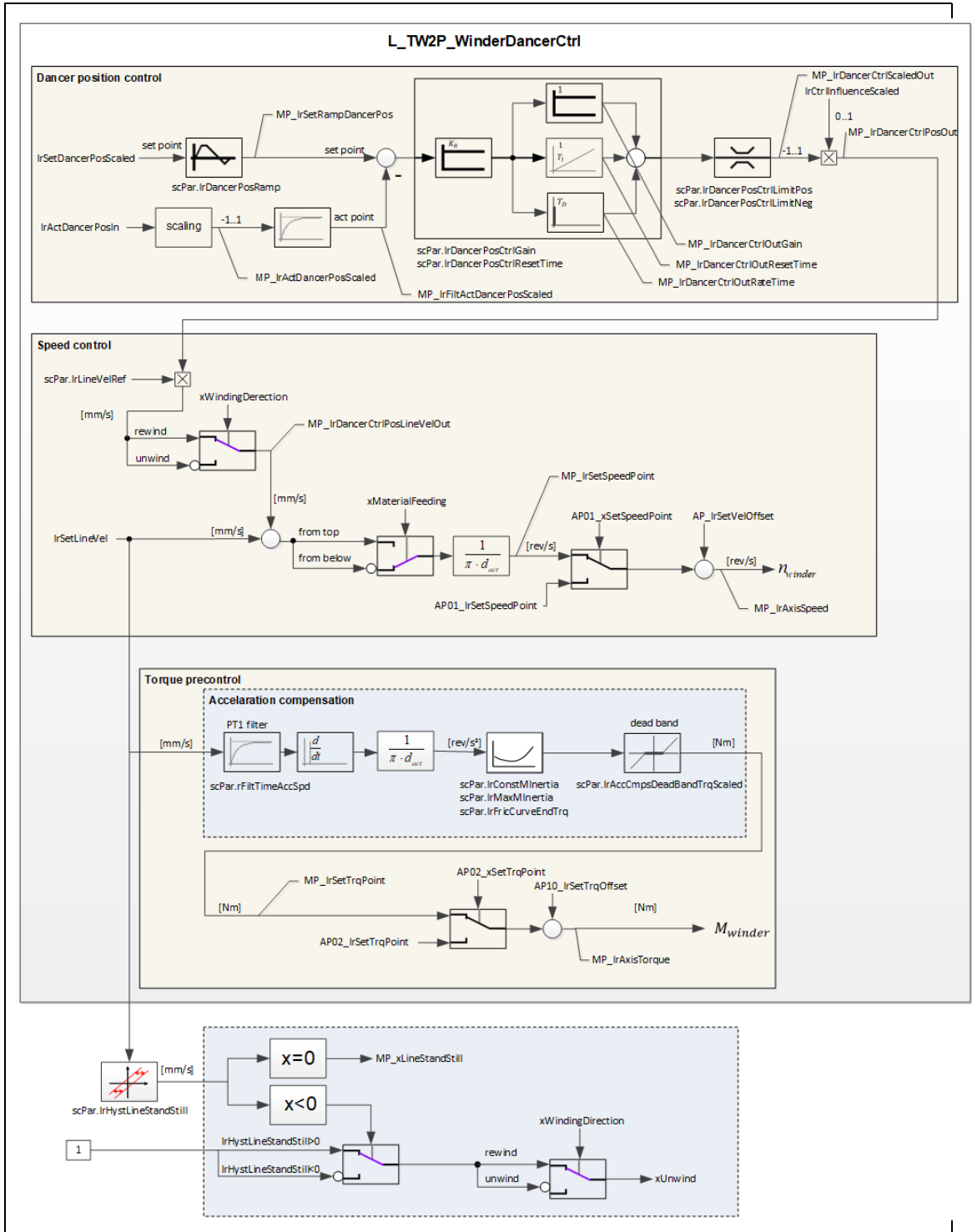
Bezeichner	Datentyp	Beschreibung
IrIdentDiamVel	LREAL	Umfangsgeschwindigkeit der Wickelwelle für die Identifikation des Durchmessers. • Einheit: mm/s • Initialwert: 10
IrIdentDiamAcc	LREAL	Umfangsbeschleunigung der Wickelwelle für die Identifikation des Durchmessers. • Einheit: mm/s ² • Initialwert: 100
IrIdentDiamDec	LREAL	Umfangsverzögerung der Wickelwelle für die Identifikation des Durchmessers. • Einheit: mm/s ² • Initialwert: 100
IrIdentDiamJerk	LREAL	Umfangsruck der Wickelwelle für die Identifikation des Durchmessers. • Einheit: mm/s ³ • Initialwert: 0
wIdentDiamCalcCycles	WORD	Anzahl der Berechnungszyklen für eine erfolgreiche Identifikation • Initialwert: 2
IrIdentDiamMaxDancerPos	LREAL	Skalierte Endlage des Tänzers während der Identifikation. Die Identifikationsfahrt wird abgebrochen, wenn die Endlage überschritten wird. Einstellbereich: -1 ... 1 Initialwert: 0.5
IrIdentDiamSpdCtrlGain	LREAL	Die Drehzahlregler-Verstärkung wird eingestellt, wenn während der Durchmesser-Identifikation die Drehzahlregler-Verstärkung $xAdaptSpdCtrlGain = TRUE$ ist. Einstellbereich: 0 ... 1 Initialwert: 0.5
IrHystlineStandstill	LREAL	Hystereseweite für die Liniengeschwindigkeit Dadurch wird die verrauschte Liniengeschwindigkeit und die Richtungserkennung stabilisiert. Nach Ermittlung der maximalen Rauschamplitude des Eingangssignals muss dieser Wert eingestellt werden. Einheit: mm/s Initialwert: 0

6



7 Signalflussplan

In der folgenden Abbildung ist der Hauptsignalfluss der Funktion dargestellt:



[7-1] Signalfluss des Technologiemoduls

7.1

Struktur des Signalflusses

L_TW2P_scSF_WinderDancerCtrlState

Die Inhalte der Struktur sind nur lesbar und bieten eine praktische Diagnosemöglichkeit innerhalb des Signalflusses.

► [Signalflussplan](#) (33)

Bezeichner	Datentyp	Beschreibung
MP_IrWndVelRef	LREAL	Nenndrehzahl des Wickelballens (abtriebsseitig) • Einheit: units/s
MP_IrWndSpeedRef	LREAL	Nenndrehzahl des Wickelballens (abtriebsseitig) • Einheit: rev/s
MP_IrRevCounter	LINT	Zählerstand des Materiallängenzählers (ganzzahlige Anzahl der vollen Umdrehungen des Wickelballens)
MP_IrRevCounterResidual	LREAL	Zählerstand des Materiallängenzählers (Restwert nach einer vollen Umdrehung des Wickelballens) • Einheit: 1 x 100 % • 100 % = Eine Umdrehung des Wickelballens
MP_IrAxisSpeed	LREAL	Aktuelle Drehzahl des Wickelballens (abtriebsseitig) • Einheit: rev/s
MP_xDiamCalculated	LREAL	FALSE↔TRUE-Flanke: Der Durchmesser wurde berechnet. Der Messpunkt wird nur für einen SPS-Zyklus auf TRUE gesetzt.
MP_IrSetDiamUnfiltered	LREAL	Aktueller berechneter Durchmesser • Einheit: mm
MP_IrSetDiamAdapted	LREAL	Aus dem Eingang <i>IrSetDiam</i> resultierender Durchmesser • Einheit: mm
MP_IrSetDiamScaledOut	LREAL	Aus dem Eingang <i>IrSetDiam</i> resultierender skaliertem Durchmesser • Einheit: 1 x 100 % • 100 % <i>IrMaxDiam</i>
MP_IrSetLineVelScaledOut	LREAL	Skalierte Liniengeschwindigkeit • Einheit: 1 x 100 % • 100 % <i>IrLineVelRef</i>
MP_IrActTotalTrqScaled	LREAL	Aus dem gemessenen Strom das berechnete Drehmoment (skaliert) • Einheit: x 100 % • 100 % Nenndrehmoment
MP_IrSetSpeedPoint	LREAL	Resultierender Drehzahlsollwert • Einheit: rev/s
MP_IrDancerPosScaled0To1	LREAL	Skalierte Tänzerlage • Einheit: 1 x 100 % • Wertebereich: -1 ... 1 (-100 ... 100 %) • -100 % = untere Tänzerendlage • 100 % = obere Tänzerendlage
MP_IrDancerCompVelOut	LREAL	Resultierende Geschwindigkeit des Tänzers • Einheit: 1 x 100 %/s
MP_IrFiltDancerCompVelOut	LREAL	Gefilterte resultierende Geschwindigkeit des Tänzers • Einheit: 1 x 100 %/s
MP_IrFiltActDancerPos Scaled	LREAL	Aktuelle skalierte gefilterte Tänzerposition • Einheit: 1 x 100 % • Wertebereich: -1 ... 1 (-100 ... 100 %)
MP_IrSetRampDancerPos	LREAL	Sollposition des Tänzers nach dem Rampengenerator Diese Position ist die Führungsgröße des Wickelantriebs. • Einheit: 1 x 100 % • Wertebereich: -1 ... 1 (-100 ... 100 %)

Bezeichner	Datentyp	Beschreibung
MP_IrDancerCtrlPosOut	LREAL	Stellgröße des Tänzerlagereglers • Einheit: 1 x 100 % • Wertebereich: -1 ... 1 (-100 ... 100 %)
MP_IrDancerCtrlLineVelOut	LREAL	Stellgröße des Tänzerlagereglers umgerechnet in die Liniengeschwindigkeit • Einheit: mm/s
MP_IrDancerCtrlOutGain	LREAL	Skalierte Stellgröße aus dem proportionalen Anteil (P-Anteil) des Tänzerlagereglers
MP_IrDancerCtrlOutReset Time	LREAL	Skalierte Stellgröße aus dem integrierenden Anteil (I-Anteil) des Tänzerlagereglers
MP_IrDancerCtrlOutRate Time	LREAL	Skalierte Stellgröße aus dem differenzierenden Anteil (D-Anteil) des Tänzerlagereglers
MP_IrDancerCtrlScaledOut	LREAL	Skalierte Stellgröße des Tänzerlagereglers • Einheit: 1 x 100 % • Wertebereich: -1 ... 1 (-100 ... 100 %)
MP_IrAxisTorque	LREAL	Drehmoment des Wickelballens (abtriebsseitig) • Einheit: Nm
MP_IrSetTrqPoint	LREAL	Drehomentvorsteuerwert des Wickelballens (abtriebsseitig) • Einheit: Nm
MP_IrAccCompTrq	LREAL	Resultierendes Drehmoment aus der Beschleunigungskompensation • Einheit: Nm
MP_IrAccTrqScaledOut	LREAL	Resultierendes Drehmoment aus der Reibungskompensation • Einheit: Nm
MP_xLineStandStill	BOOL	TRUE Stillstand der Linie wurde erkannt.

7.2

Struktur der Angriffspunkte

L_TW2P_scAP_WinderDancerCtrlState

Über die Angriffspunkte (AP) können Signale beeinflusst werden. Im Initialzustand haben die Angriffspunkte keine Wirkung.

Jeder Angriffspunkt wirkt als ein alternativer Zweig und wird über eine ODER-Verknüpfung oder einen Schalter aktiviert.

Bezeichner	Datentyp	Beschreibung
AP_xSetVelOffset	BOOL	Freigabe des Angriffspunktes <i>AP_IrVelOffset</i>
	TRUE	Der Angriffspunkt überschreibt die Werte an der Zugriffstelle im Signalfluss.
AP_IrVelOffset	LREAL	Zyklische Vorgabe des Offset für die Geschwindigkeit der Wicklerachse bezogen auf die Wickelwelle (Getriebeausgangsseite) • Einheit: units/s Der Offset-Wert wird ohne Rampengenerator sofort und sprunghaft eingestellt!
AP_xSetTrqOffset	BOOL	Freigabe des Angriffspunktes <i>AP_IrSetTrqOffset</i>
	TRUE	Der Angriffspunkt überschreibt die Werte an der Zugriffstelle im Signalfluss.
AP_IrSetTrqOffset	LREAL	Zyklische Vorgabe des Offset für das Drehmoment der Wicklerachse bezogen auf die Wickelwelle (Getriebeausgangsseite) • Einheit: Nm Der Offset-Wert wird ohne Rampengenerator sofort und sprunghaft eingestellt!
AP_xSetVelOffsetDiamCalc	BOOL	Freigabe des Angriffspunktes <i>AP_IrSetVelOffsetDiamCalc</i>
	TRUE	Der Angriffspunkt überschreibt die Werte an der Zugriffstelle im Signalfluss
AP_IrVelOffsetDiamCalc	LREAL	Zyklische Vorgabe des Offset für die Geschwindigkeit der Wicklerachse bezogen auf die Wickelwelle (Getriebeausgangsseite) für die Berechnung des Durchmessers • Einheit: units/s Der Offset-Wert wird ohne Rampengenerator sofort und sprunghaft eingestellt!
AP_xSetSpeedPointDiamCalc	BOOL	Freigabe des Angriffspunktes <i>AP_IrSetSpeedPointDiamCalc</i>
	TRUE	Der Angriffspunkt überschreibt die Werte an der Zugriffstelle im Signalfluss
AP_IrSetSpeedPointDiamCalc	LREAL	Zyklisches Laden des Drehzahlsollwertes für den Antrieb • Einheit: rev/s Der Offset-Wert wird ohne Rampengenerator sofort und sprunghaft eingestellt!
AP_xSetSpeedPoint	BOOL	Freigabe des Angriffspunktes <i>AP_IrSetSpeedPoint</i>
	TRUE	Der Angriffspunkt überschreibt die Werte an der Zugriffstelle im Signalfluss.
AP_IrSetSpeedPoint	LREAL	Drehzahlsollwert für den Wickelantrieb • Einheit: rev/s
AP_xSetTrqPoint	BOOL	Freigabe des Angriffspunktes <i>AP_IrSetTrqPoint</i>
	TRUE	Der Angriffspunkt überschreibt die Werte an der Zugriffstelle im Signalfluss.
AP_IrSetTrqPoint	LREAL	Drehmomentsollwert für die Vorsteuerung • Einheit: Nm

8 Beschreibung der Basisfunktionen

Übersicht der Basisfunktionen

- [Handfahren \(Jogging\)](#) (📖 38)
- [Referenzfahrt \(Homing\)](#) (📖 40)
- [QSP Application](#) (📖 43)
- [Halt](#) (📖 44)
- [Absolute Positionierung](#) (📖 45)
- [Relative Positionierung](#) (📖 47)
- [Endlosfahrt mit definierter Geschwindigkeit](#) (📖 50)
- [Abbruch und Beendigung einer Positionierung](#) (📖 53)
- [Override für Geschwindigkeit](#) (📖 56)

8.1

Handfahren (Jogging)**Voraussetzung**

- Die Achse ist freigegeben (*xReadyForMotion* = TRUE).

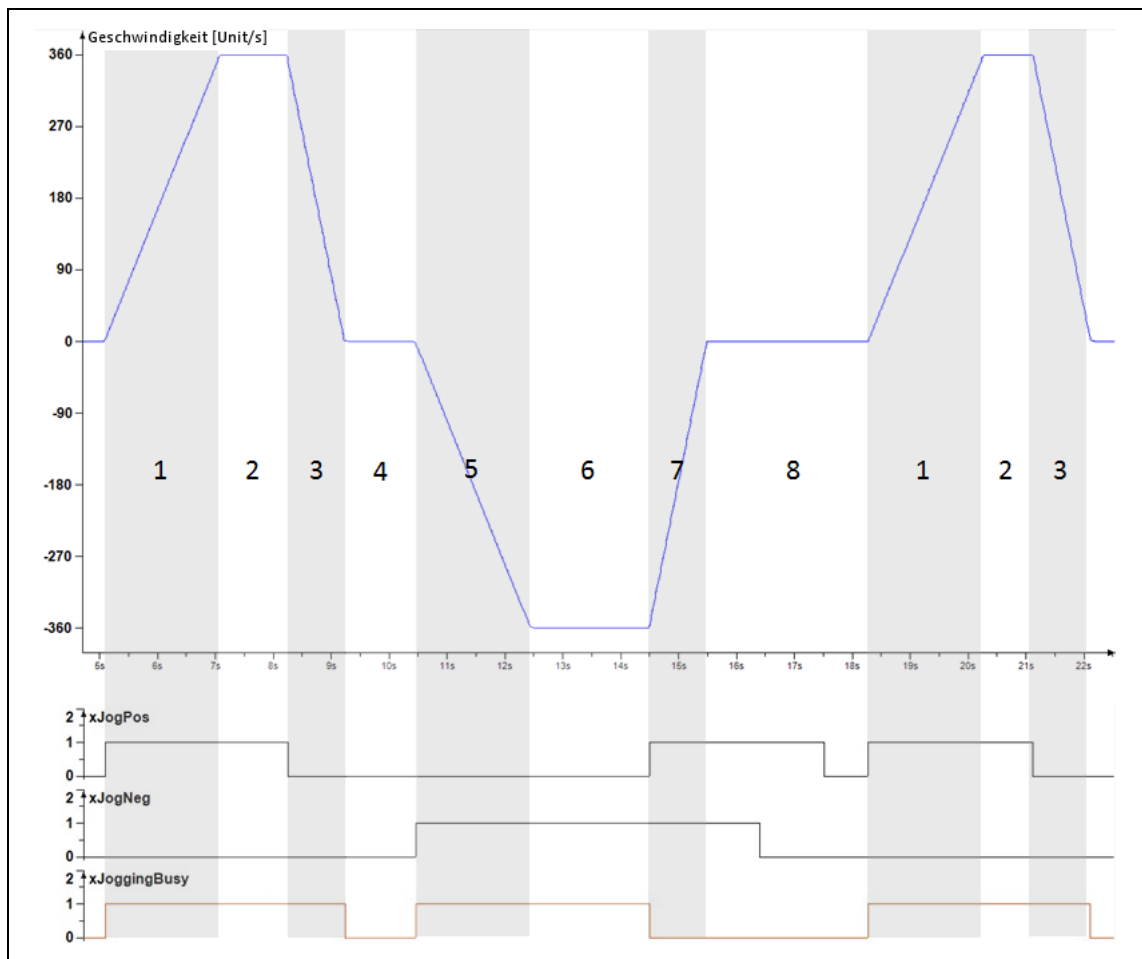
Ausführung

Mit dem Eingang *xJogPos* = TRUE wird die Achse in positive Richtung und mit dem Eingang *xJogNeg* = TRUE in negative Richtung gefahren. Die Achse wird solange gefahren, wie *xJogPos* = TRUE oder *xJogNeg* = TRUE gesetzt bleibt.

Der laufende Fahrbefehl kann nicht durch den anderen Jog-Befehl abgelöst werden. Wenn *xJogPos* = TRUE und *xJogNegPos* = TRUE sind, bremst die Achse in den Stillstand und der Statusausgang *eTMState* wechselt in den Zustand "ERROR".

Erst wenn beide Eingänge zurückgesetzt wurden, wechselt das Technologiemodul wieder zurück in den Zustand "STANDBY".

Fahrtdiagramm



Nr	xJogPos	xJogNeg	xJogging Busy	eTMState	INFO
1	FALSE → TRUE	FALSE	FALSE → TRUE	STANDBY → JOGGING	Die Achse beschleunigt mit 180 units/s ² auf die positive Handfahrgeschwindigkeit 360 units/s.
2	TRUE	FALSE	TRUE	JOGGING	Die Achse verfährt konstant mit 360 units/s.
3	FALSE → TRUE	FALSE	TRUE	JOGGING	Die Achse wird mit 360 units/s ² in den Stillstand geführt.
4	FALSE	FALSE	TRUE → FALSE	JOGGING → STANDBY	Die Achse hat den Stillstand erreicht.
5	FALSE	FALSE → TRUE	FALSE → TRUE	STANDBY → JOGGING	Die Achse beschleunigt mit 180 units/s ² auf die negative Handfahrgeschwindigkeit -360 units/s.
6	FALSE	TRUE	TRUE	JOGGING	Die Achse verfährt konstant mit -360 units/s.
7	TRUE	TRUE	TRUE → FALSE	ERROR	Die Achse wird mit 360 units/s ² in den Stillstand geführt.
8	FALSE	FALSE	FALSE	ERROR → STANDBY	Die Achse steht still und wechselt in den Zustand "STANDBY", wenn beide Eingänge xJogPos und xJogNeg den Wert FALSE haben.

Einzustellende Parameter

Die Parameter für das Handfahren befinden sich in der AXIS_REF in Subindex 181 ... 184.

Die Parameterwerte können während des Betriebs verändert werden. Sie werden bei erneutem Setzen der Eingänge *xJogPos* = TRUE oder *xJogNeg* = TRUE übernommen und sofort wirksam.

Die Handfahrparameter aus dem SPS-Programm können auch über den Baustein **L_MC1P_SetBasicParameter** geschrieben werden.

8.2

Referenzfahrt (Homing)**Voraussetzungen**

- Des Technologiemodul befindet sich im Zustand "STANDBY".
- Die Achse ist freigegeben (*xReadyForMotion* = TRUE).

Ausführung

Mit einer steigenden Flanke (*FALSE* → *TRUE*) am Eingang *xHomeExecute* wird die Referenzfahrt gestartet (Ausgang *xHomingBusy* = TRUE). Die Achse fährt solange, bis die Referenzposition erreicht ist.

Nach Erreichen des Referenzschalters wird *xisHomed* = TRUE.

Bei erfolgreicher Beendigung der Referenzfahrt wechselt *xHomingDone* = TRUE.

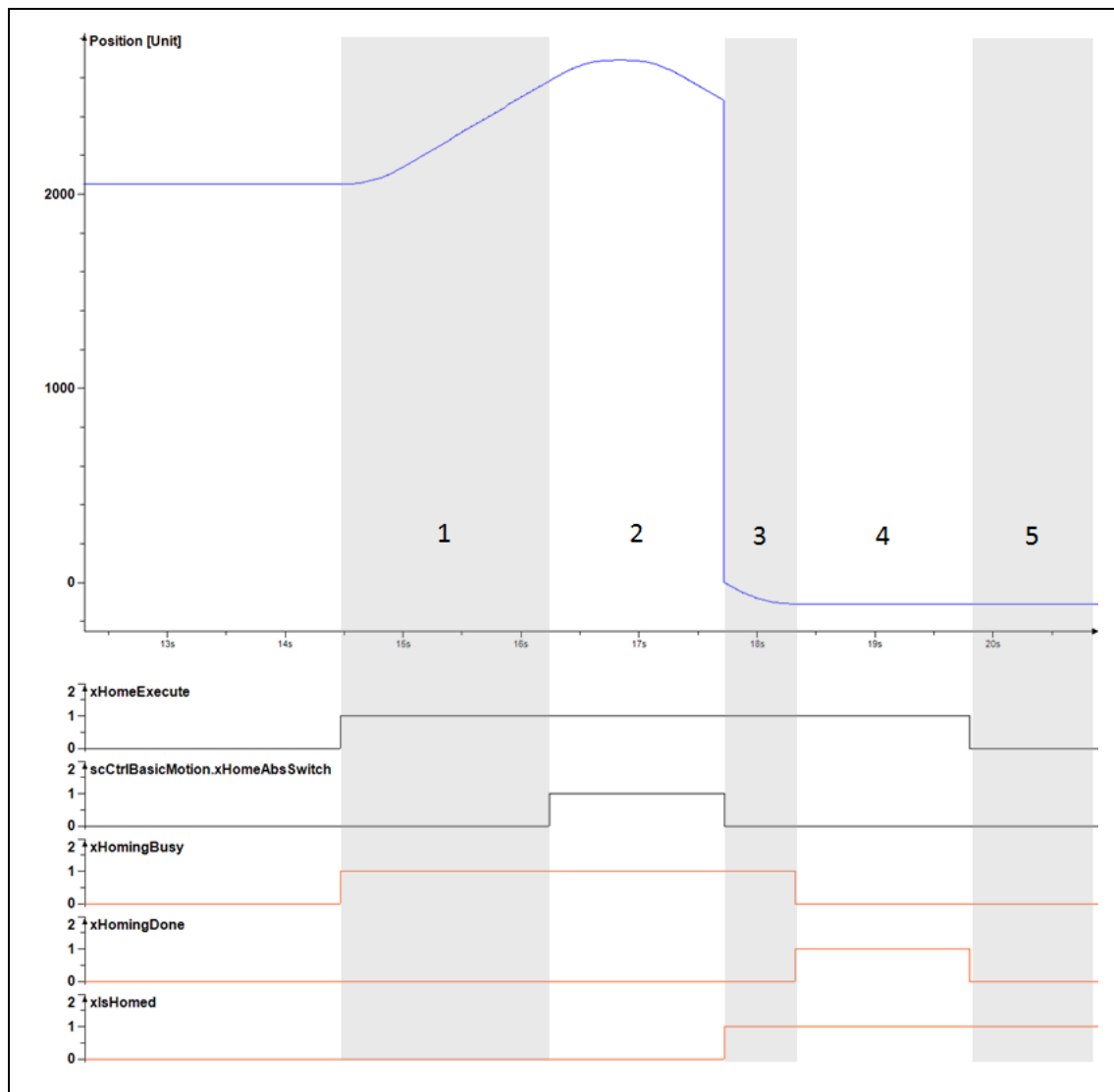
Wird *xHomeExecute* = FALSE gesetzt, dann wird in den Zustand "STANDBY" gewechselt und *xHomingDone* = FALSE gesetzt.

Abbruch der Referenzfahrt

Über den Eingang *xHalt* wird die Funktion abgebrochen.

Die Referenzfahrt wird nicht unterbrochen, wenn vorzeitig *xHomeExecute* = FALSE gesetzt wird.

Fahrtdiagramm



[8-1] Referenzfahrt im Modul CwRpCcwRn mit der Referenzposition "0"

Nr	xHome Execute	xHome AbsSwitch	xHoming Busy	xHoming Done	xlsHomed	eTMState	INFO
1	FALSE → TRUE	FALSE	FALSE → TRUE	FALSE	FALSE	STANDBY → HOMING	Die Achse startet die Referenzfahrt in positive Richtung.
2	TRUE	TRUE → FALSE	TRUE	FALSE	FALSE	HOMING	Die Achse wechselt die Drehrichtung aufgrund der positiven Flanke von xHomeAbsSwitch.
3	TRUE	FALSE → TRUE	TRUE	FALSE	FALSE → TRUE	HOMING	Die negative Flanke von xHomeAbsSwitch setzt die Position der Achse auf die Referenzposition 0 units und die Achse wird in den Stillstand geführt.
4	TRUE	FALSE	TRUE → FALSE	FALSE → TRUE	TRUE	HOMING	Sobald die Achse den Stillstand erreicht hat, wird xHomingBusy = FALSE und xHomingDone = TRUE gesetzt. xHomingDone ist TRUE, solange xHomeExecute = TRUE gesetzt ist, aber immer mindestens für einen Taktzyklus.
5	TRUE → FALSE	FALSE	FALSE	TRUE → FALSE	TRUE	HOMING → STANDBY	Die Achse ist betriebsbereit und kann absolut positioniert werden.

Einzustellende Parameter

Die Parameter für die Referenzfahrt befinden sich in der AXIS_REF in Subindex 70 ... 83.

Die Handfahrparameter aus dem SPS-Programm können auch über den Baustein **L_MC1P_SetBasicParameter** geschrieben werden.

8.3 QSP Application

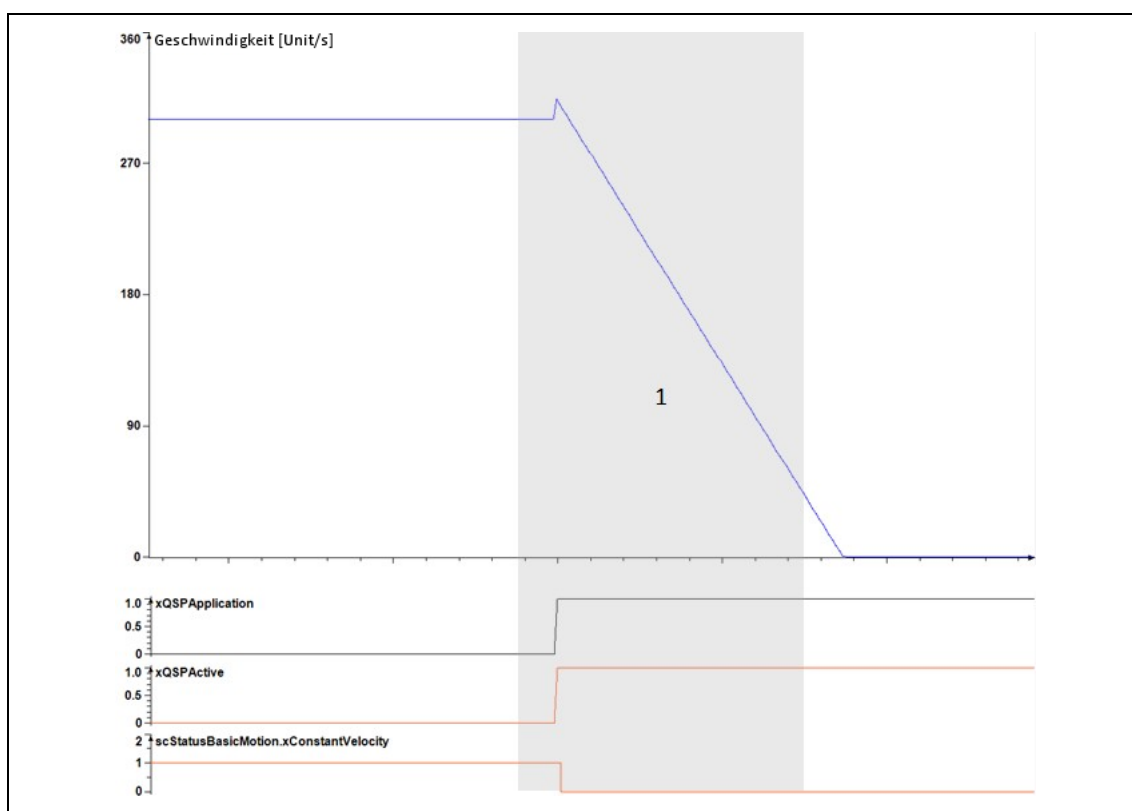
Voraussetzung

- Die Achse ist freigegeben ($xReadyForMotion = TRUE$).

Ausführung

Mit einer steigenden Flanke ($FALSE \rightarrow TRUE$) am Eingang $xQSPApplication$ wird die aktuelle Bewegung der Achse unterbrochen und aufgrund ihres letzten aktuellen Geschwindigkeitswertes an der QSP-Application-Verzögerungsrampe in den Stillstand geführt. Der Parameter für die QSP-Application-Verzögerungsrampe wird in der $AXIS_REF$ eingestellt.

Fahrtdiagramm



Nr	xQSP Application	xQSPActive	xConstant Velocity	eTMState	INFO
1	FALSE $\rightarrow$ TRUE	FALSE $\rightarrow$ TRUE	TRUE $\rightarrow$ FALSE	MOVEVEL $\rightarrow$ STOP	Der Geschwindigkeitssollwert der Achse wird auf die letzte aktuelle Geschwindigkeit gesetzt und über die QSP-Application-Verzögerungsrampe in den Stillstand geführt.

Einzustellende Parameter

Die Parameter für den "QSP Application" befinden sich in der $AXIS_REF$ in Subindex 48 und 49.

Die Handfahrparameter aus dem SPS-Programm können auch über den Baustein **L_MC1P_SetBasicParameter** geschrieben werden.

8.4

Halt

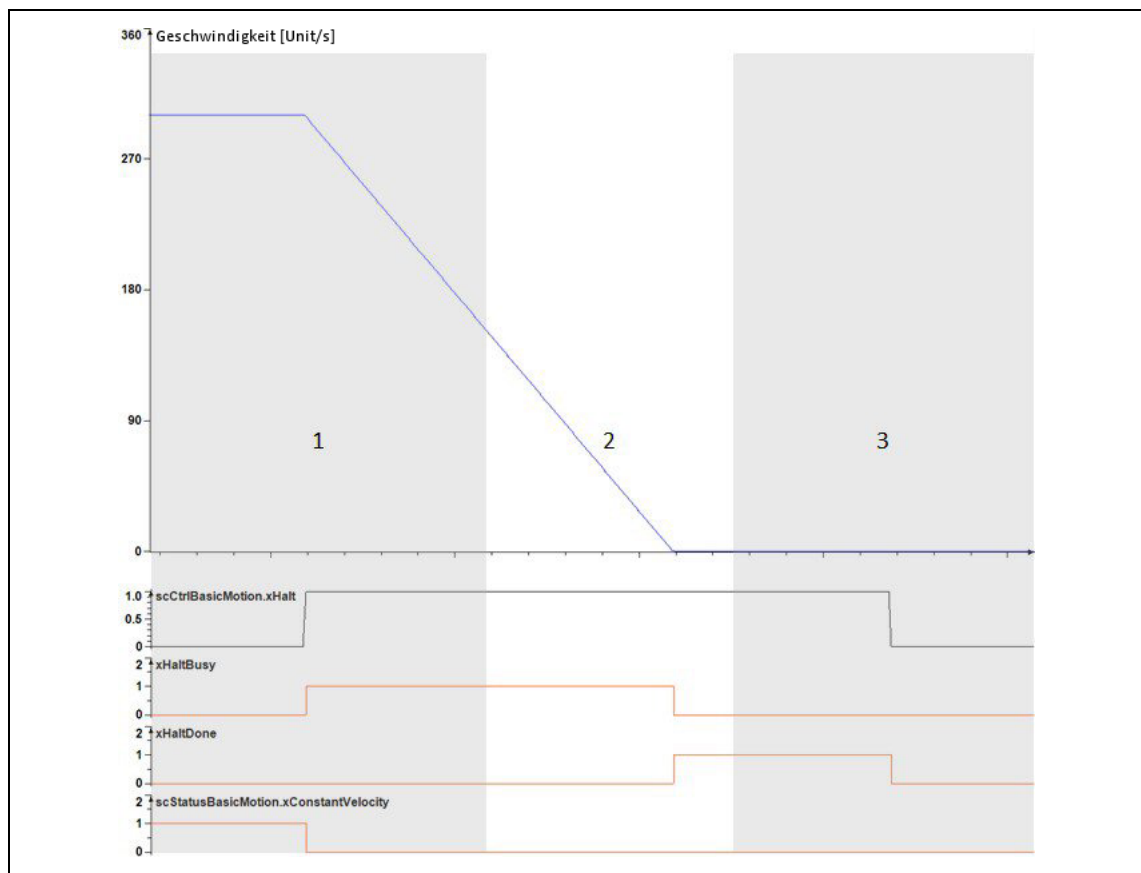
Voraussetzung

- Die Achse ist freigegeben ($xReadyForMotion = TRUE$).

Ausführung

Mit einer steigenden Flanke ($FALSE \rightarrow TRUE$) am Eingang $xHalt$ wird die aktuelle Bewegung der Achse unterbrochen und mit der Halt-Verzögerungsrampe in den Stillstand geführt.

Fahrtdiagramm

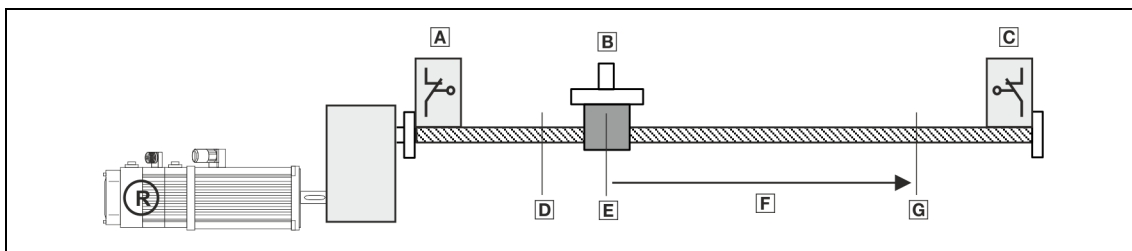


Nr	$xHalt$	$xHaltBusy$	$xHaltDone$	$xConstantVelocity$	eTMState	INFO
1	$FALSE \rightarrow TRUE$	$FALSE \rightarrow TRUE$	$FALSE$	$TRUE \rightarrow FALSE$	$MOVEVEL \rightarrow STOP$	Die Achse verzögert mit der Halt-Verzögerungsrampe: $xHaltBusy = TRUE$ $xConstantVelocity = FALSE$.
2	$TRUE$	$TRUE \rightarrow FALSE$	$FALSE \rightarrow TRUE$	$FALSE$	$STOP$	Die Achse wird in den Stillstand geführt: $xHaltBusy = FALSE$ $xHaltDone = TRUE$
3	$TRUE \rightarrow FALSE$	$FALSE \rightarrow TRUE$	$TRUE \rightarrow FALSE$	$FALSE$	$STOP \rightarrow STANDBY$	$xHalt = FALSE$ $xHaltDone = TRUE \rightarrow FALSE$.

Einzustellende Parameter

Die Parameter für den Halt befinden sich in der $AXIS_REF$ in Subindex 186 und 187.

8.5 Absolute Positionierung



[8-2] Beispiel: Absolute Positionierung einer linearen Achse

Pos.	Beschreibung
A	Negativer Hardware-Endschalter
B	Last (z. B. Schlitten)
C	Positiver Hardware-Endschalter
D	Nullposition des Maßsystems
E	Aktuelle Istposition
F	Zu fahrende Strecke
G	Zielposition (Eingang <i>IrPos_Dist</i>)

Bei der absoluten Positionierung wird die Zielposition im Eingang *IrPos_Dist* (G) als eindeutige absolute Position, bezogen auf die Nullposition des Maßsystems (D), vorgegeben. Die zu fahrende Wegstrecke (F) ergibt sich aus der Distanz zwischen der aktuellen Istposition (E) und der absoluten Zielposition (G).

Voraussetzungen

- Die Achse ist freigegeben (*xReadyForMotion* = TRUE).
- Die Achse ist referenziert (*xIsHomed* = TRUE).

Ausführung

Mit einer steigenden Flanke (FALSE→TRUE) am Eingang *xMoveAbsExecute* wird eine Fahrt auf die absolute Zielposition im Eingang *IrPos_Dist* gestartet, gemäß der Parameter für das Bewegungsprofil (siehe unten). Das Technologiemodul wechselt in den Zustand "MOVEABS".

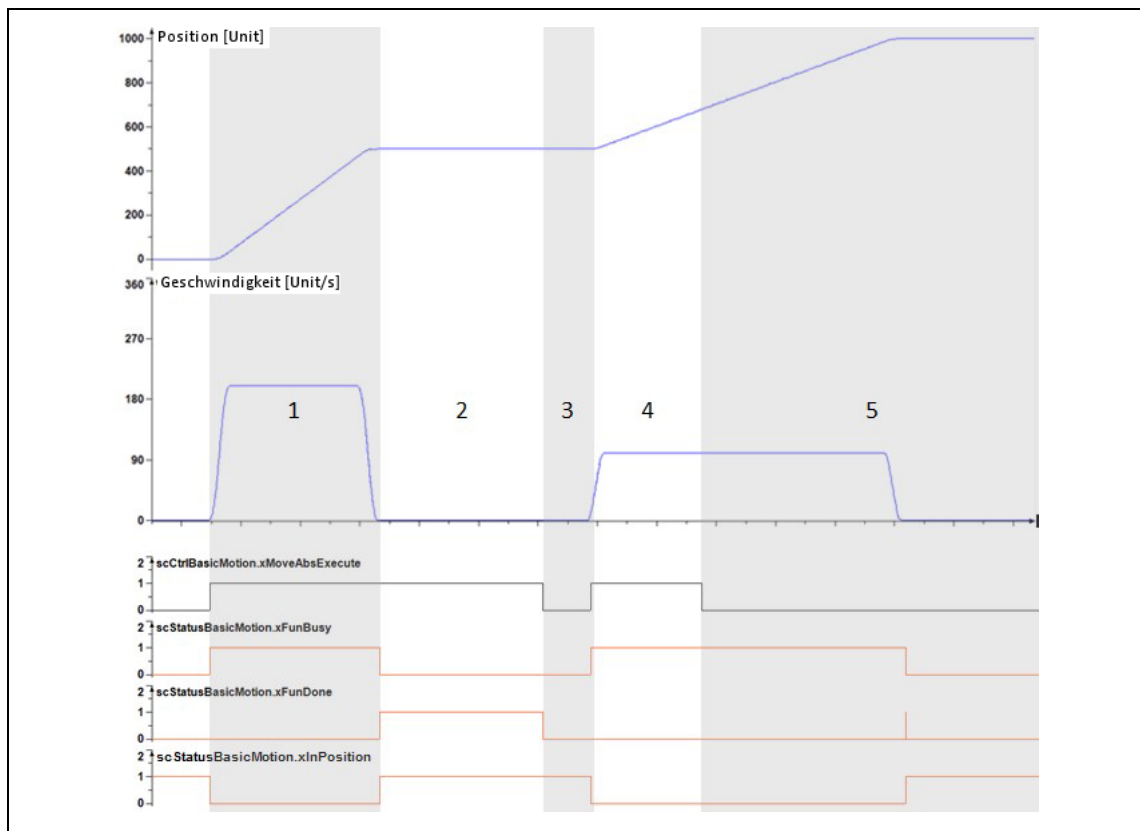
Zum Erreichen der Zielposition wird die Achse über den Parameter *IrDec* in den Stillstand geführt. Die Achse kommt genau in der Zielposition zum Stillstand.

Nach dem Erreichen der Zielposition wird zurück in den Zustand "STANDBY" gewechselt (*xMoveAbsExecute* = FALSE).

Während der Fahrt kann die absolute Zielposition im Eingang *IrPos_Dist* geändert werden. Hierbei ist ein erneutes Auslösen des Execute-Eingangs für die Positionierung erforderlich.

Eine Positionierung kann unterbrochen werden über *xAbort*, *xHalt* oder *xQSPApplication*. Über *xFinishPositioning* kann die Positionierung dann wieder fortgesetzt und beendet werden.

Fahrtdiagramm



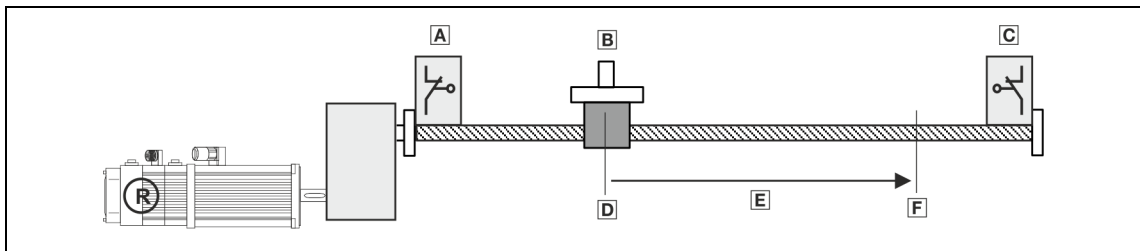
Nr	xMoveAbsExecute	xFunBusy	xFunDone	xInPosition	eTMState	INFO
1	FALSE → TRUE	FALSE → TRUE	FALSE	TRUE → FALSE	STANDBY → MOVEABS	Die Achse startet die Positionierfahrt auf die absolute Zielposition 500 units.
2	TRUE	TRUE → FALSE	FALSE → TRUE	FALSE → TRUE	MOVEABS → STANDBY	Nach Erreichen der Zielposition wird <i>xInPosition</i> = TRUE gesetzt.
3	TRUE → FALSE	FALSE	TRUE → FALSE	TRUE	STANDBY	Sobald <i>xMoveAbsExecute</i> = FALSE gesetzt wird, wechselt das Statussignal <i>xFunDone</i> von TRUE → FALSE.
4	FALSE → TRUE	FALSE → TRUE	FALSE	TRUE → FALSE	STANDBY → MOVEABS	Die Achse startet die Positionierfahrt auf die absolute Zielposition 1000 units.
5	TRUE → FALSE	TRUE → FALSE	FALSE → TRUE → FALSE	FALSE → TRUE	MOVEABS → STANDBY	Nach Erreichen der Zielposition ist <i>xFunDone</i> = TRUE für genau einen Takt, weil während der Positionierung <i>xMoveAbsExecute</i> = FALSE angefordert wurde.

Einzustellende Parameter

- *IrVel*, Initialwert: 3600 [units/s]
- *IrAcc*, Initialwert: 36000 [units/s²]
- *IrDcc*, Initialwert: 36000 [units/s²]
- *IrJerk*, Initialwert: 0 [units/s³]

Weitere Parameter für die absolute Positionierung befinden sich in der AXIS_REF in Subindex 130 ... 137.

8.6 Relative Positionierung



[8-3] Beispiel: Relative Positionierung einer linearen Achse

Pos.	Beschreibung
A	Negativer Hardware-Endschalter
B	Last (z. B. Schlitten)
C	Positiver Hardware-Endschalter
D	Aktuelle Istposition
E	Zu fahrende Strecke (= Eingang <i>IrPos_Dist</i>)
F	Zielposition (Eingang aktuelle Istposition + <i>IrPos_Dist</i>)

Bei der relativen Positionierung entspricht die vorgegebene Zielposition im Eingang *IrPos_Dist* der zu fahrenden Wegstrecke ($F = E$). Der Antrieb fährt also genau die vorgegebene Distanz ab der aktuellen Istposition (D).

Voraussetzung

- Die Achse ist freigegeben (*xReadyForMotion* = TRUE).

Ausführung

Mit einer steigenden Flanke ($FALSE \rightarrow TRUE$) am Eingang *xMoveRelExecute* wird eine Fahrt auf die relative Zielposition (Eingang *IrPos_Dist*) gestartet, gemäß der Parameter für das Bewegungsprofil (siehe unten). Das Technologiemodul wechselt in den Zustand "MOVEREL":

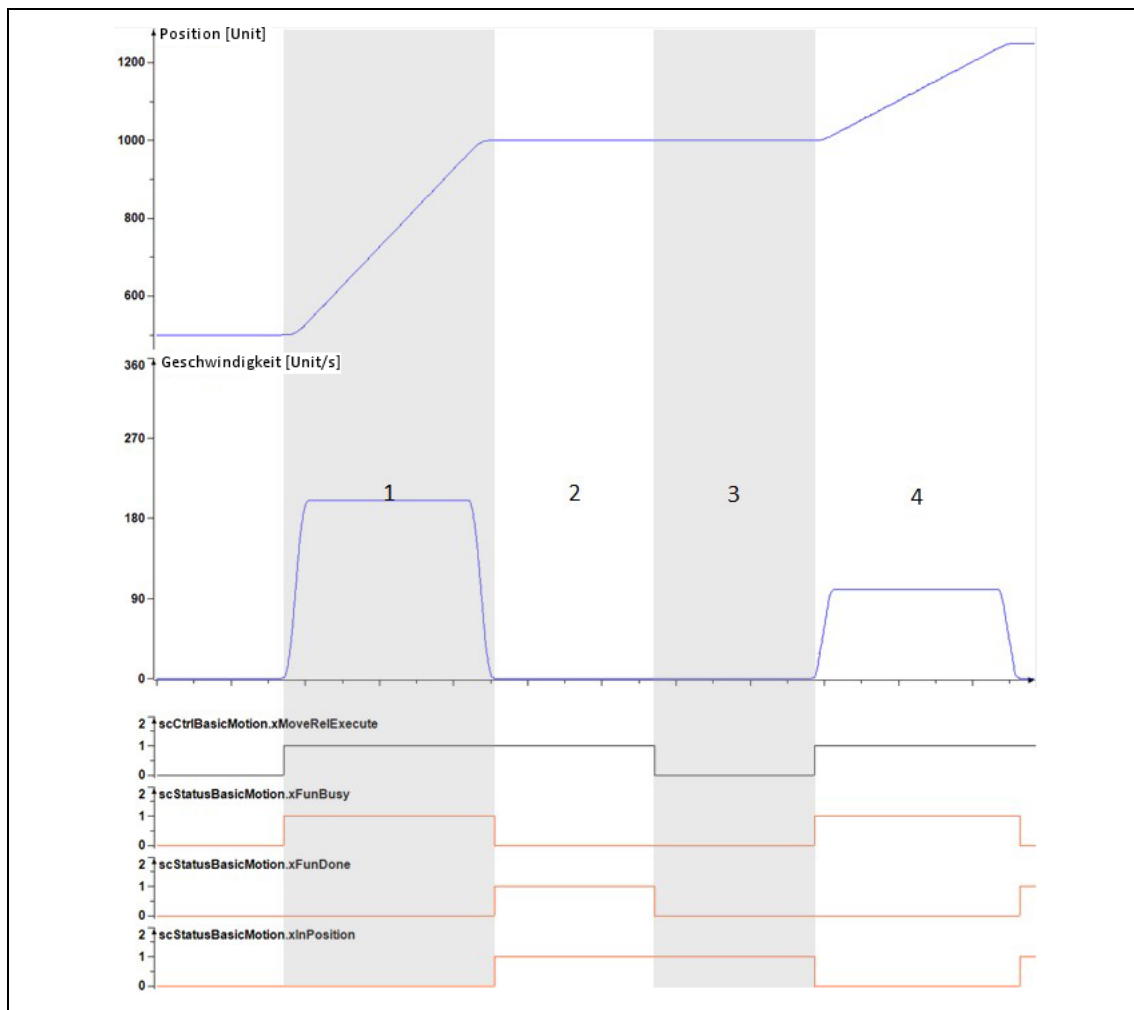
Zum Erreichen der Zielposition wird die Achse über den Parameter *IrDec* in den Stillstand geführt. Die Achse kommt genau in der Zielposition zum Stillstand.

Nach dem Erreichen der Zielposition wird zurück in den Zustand "STANDBY" gewechselt (*xMoveRelExecute* = FALSE).

Während der Fahrt kann die absolute Zielposition im Eingang *IrPos_Dist* geändert werden. Hierbei ist ein erneutes Auslösen des Execute-Eingangs für die Positionierung erforderlich.

Eine Positionierung kann unterbrochen werden über *xAbort*, *xHalt* oder *xQSPApplication*. Über *xFinishPositioning* kann die Positionierung dann wieder fortgesetzt und beendet werden.

Fahrtdiagramm



Nr	xMoveRelExecute	xFunBusy	xFunDone	xInPosition	eTMState	INFO
1	FALSE ↗ TRUE	FALSE ↗ TRUE	FALSE	FALSE	STANDBY → MOVEREL	Die Achse startet die Positionierfahrt auf die relative Zielposition 500 units.
2	TRUE	TRUE ↘ FALSE	FALSE ↗ TRUE	FALSE ↗ TRUE	MOVEREL → STANDBY	Sobald die relative Zielposition erreicht ist, wird <i>xInPosition</i> = TRUE und <i>xFunDone</i> = TRUE gesetzt.
3	FALSE ↗ TRUE	FALSE -	TRUE ↘ FALSE	TRUE	STANDBY	Sobald <i>xMoveRelExecute</i> = FALSE gesetzt wird, wechselt <i>xFunDone</i> von TRUE ↘ FALSE.
4	FALSE ↗ TRUE	FALSE ↗ TRUE ↘ FALSE	TRUE ↘ FALSE ↗ TRUE	TRUE ↘ FALSE ↗ TRUE	STANDBY → MOVEREL → STANDBY	Die Achse startet die Positionierfahrt auf die relative Zielposition 250 units. Nach Erreichen der relativen Zielposition wird <i>xInPosition</i> = TRUE und <i>xFunDone</i> = TRUE gesetzt.

Einzustellende Parameter

- *IrVel*, Initialwert: 3600 [units/s]
- *IrAcc*, Initialwert: 36000 [units/s²]
- *IrDcc*, Initialwert: 36000 [units/s²]
- *IrJerk*, Initialwert: 0 [units/s³]

Weitere Parameter für die relative Positionierung befinden sich in der AXIS_REF in Subindex 130 ... 137.

8.7 Endlosfahrt mit definierter Geschwindigkeit

Voraussetzung

- Die Achse ist freigegeben (*xReadyForMotion* = TRUE).

Ausführung

Mit einer steigenden Flanke (FALSE→TRUE) am Eingang *xMoveVelExecute* wird die Endlosfahrt gestartet, gemäß der Parameter für das Bewegungsprofil (siehe unten). Das Technologiemodul wechselt in den Zustand "MOVEVEL".

Die Achse fährt solange bis eine Software-Endlage erreicht wurde oder bis ein Hardware-Endschalter angefahren wurde:

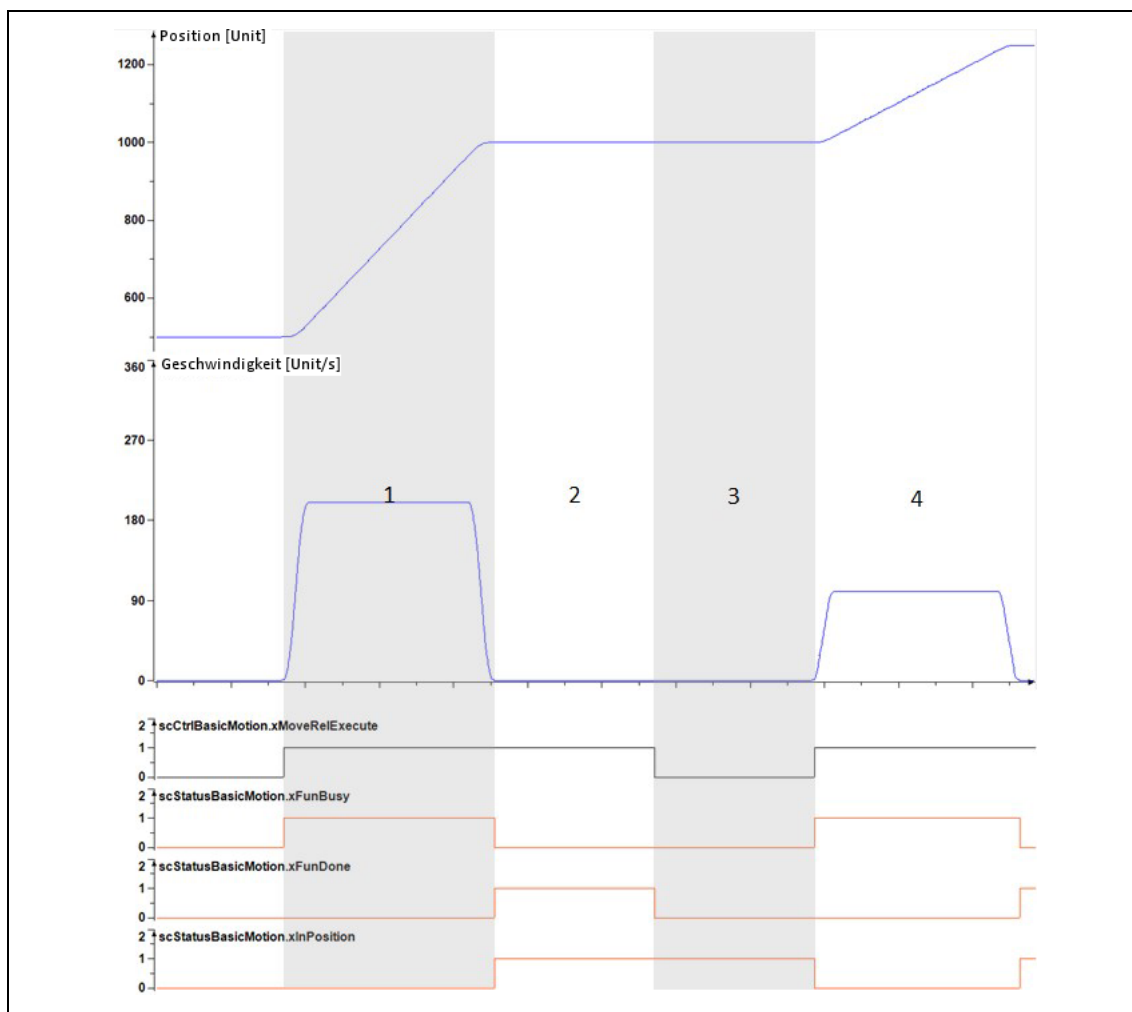
- Eingang *xDisableSWLimit* = FALSE
- Ausgang *xSwLimitEnabled* = TRUE
- Ausgang *xSwLimitSwitchPos/Neg* = TRUE
- Ausgang *xHwLimitSwitchPos/Neg* = TRUE

Bei Erreichen einer Software-Endlage wird die Achse über den Parameter *IrDec* in den Stillstand geführt.

Bei Erreichen eines Hardware-Endschalters wird die Achse über die QSP-Application-Verzögerungsrampe in den Stillstand geführt.

Danach wird zurück in den Zustand "STANDBY" gewechselt.

Fahrdiagramm



Nr	xMoveVel Execute	xAbort	xFunBusy	xFunDone	xConstant Velocity	eTMState	INFO
1	FALSE ↗ TRUE ↘ FALSE	FALSE	FALSE ↗ TRUE	FALSE	FALSE	STANDBY → MOVEVEL	Die Achse beschleunigt auf die Zielgeschwindigkeit von 200 units/s.
2	FALSE	FALSE	TRUE	FALSE ↗ TRUE	FALSE ↗ TRUE	MOVEVEL	Nach Erreichen der Zielgeschwindigkeit, wechseln <i>xFunDone</i> von FALSE ↗ TRUE und <i>xConstantVelocity</i> von FALSE ↗ TRUE.
3	FALSE	FALSE ↗ TRUE	TRUE ↘ FALSE	TRUE ↘ FALSE ↗ TRUE	TRUE ↘ FALSE	MOVEVEL → STANDBY	Die Geschwindigkeitsfahrt wird durch <i>xAbsort</i> = TRUE beendet und mit der Profilverzögerung in den Stillstand geführt. Nach Erreichen des Stillstands wird <i>xFunBusy</i> = FALSE gesetzt.

Einzustellende Parameter

- *IrVel*, Initialwert: 3600 [units/s]
- *IrAcc*, Initialwert: 36000 [units/s²]
- *IrDcc*, Initialwert: 36000 [units/s²]
- *IrJerk*, Initialwert: 0 [units/s³]

Weitere Parameter für die Endlosfahrt mit definierter Geschwindigkeit befinden sich in der AXIS_REF in Subindex 50 ... 53 und Subindex 104 ... 105.

8.8 Abbruch und Beendigung einer Positionierung

Voraussetzung

- Die Achse ist freigegeben (*xReadyForMotion* = TRUE).

Ausführung

Eine aktive Positionierung wird mit einer steigenden Flanke (FALSE↗TRUE) am Eingang *xAbort* abgebrochen.

Die Achse fährt mit der Verzögerungsrampe *scPar.IrDec* in den Stillstand.

- Ausnahme: TM **SpeedControl** und TM **TablePositioning**

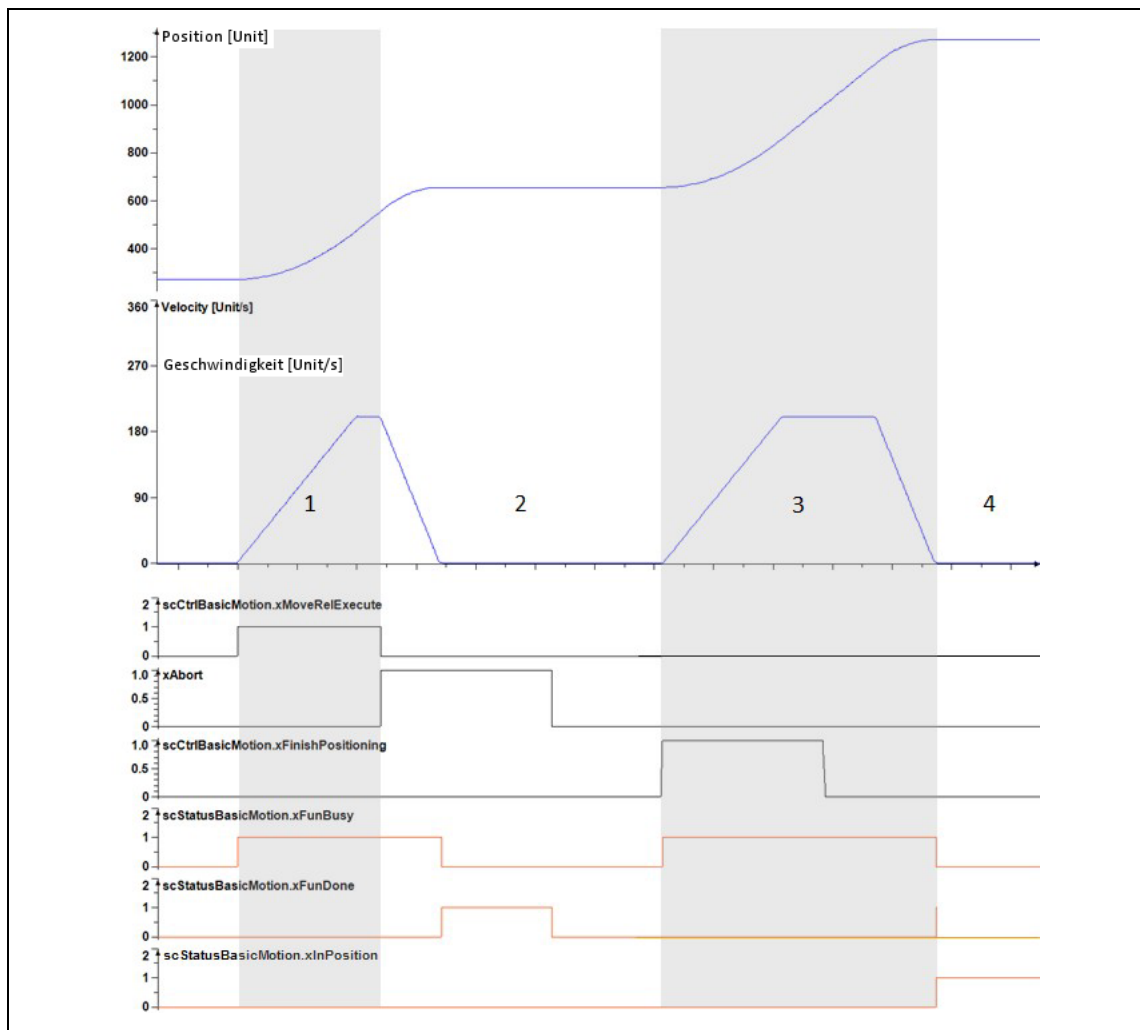
Beide Technologiemodule stellen mehrere Profildatensätze zur Verfügung von denen einer aktiv sein kann.

Wenn einer dieser Profildatensätze aktiv ist, fährt die Achse mit der Verzögerungsrampe dieses aktiven Profildatensatzes in den Stillstand.

Wenn kein Profildatensatz aktiv ist, wird die standardmäßige Verzögerungsrampe *scPar.IrDec* verwendet.

Die Positionierung wird beendet mit einer steigenden Flanke (FALSE↗TRUE) am Eingang *xFinishPositioning*.

Fahrdiagramm



Nr	xMoveRel Execute	xAbort	xFinish Positioning	xInPosition	eTMState	INFO
1	FALSE ↗ TRUE	FALSE	FALSE	FALSE	STANDBY → MOVEREL	Die Achse startet die Positionierfahrt mit der relativen Zielposition 1000 units.
2	TRUE ↘ FALSE	TRUE ↘ FALSE	FALSE	FALSE	MOVEREL → STOP → STANDBY	Die Positionierfahrt wird mit <i>xAbsort</i> = TRUE abgebrochen. Die Achse wird mit der Profilverzögerung in den Stillstand geführt.
3	FALSE	FALSE ↗ TRUE	FALSE ↗ TRUE ↘ FALSE	FALSE	STANDBY → MOVEREL	Mit <i>xFinishPositioning</i> = TRUE wird die Positionierung fortgesetzt.
4	FALSE	FALSE	FALSE	FALSE ↗ TRUE	MOVEREL → STANDBY	Sobald die relative Zielposition erreicht ist, werden <i>xInPosition</i> und <i>xFunDone</i> auf TRUE gesetzt.
	FALSE ↗ TRUE	FALSE ↗ TRUE ↘ FALSE	TRUE ↘ FALSE ↗ TRUE	TRUE ↘ FALSE ↗ TRUE	STANDBY → MOVEREL → STANDBY	Die Achse startet die Positionierfahrt mit der relativen Zielposition 250 units. Nach Erreichen der relativen Zielposition wird <i>sxInPosition</i> = TRUE und <i>xFunDone</i> = TRUE gesetzt.

8.9 Override für Geschwindigkeit

Voraussetzung

- Die Achse ist freigegeben (*xReadyForMotion* = TRUE).

Ausführung

Um Overrides für die aktiven Profildaten einstellen zu können, stehen am Technologiemodul die folgenden Eingänge zur Verfügung:

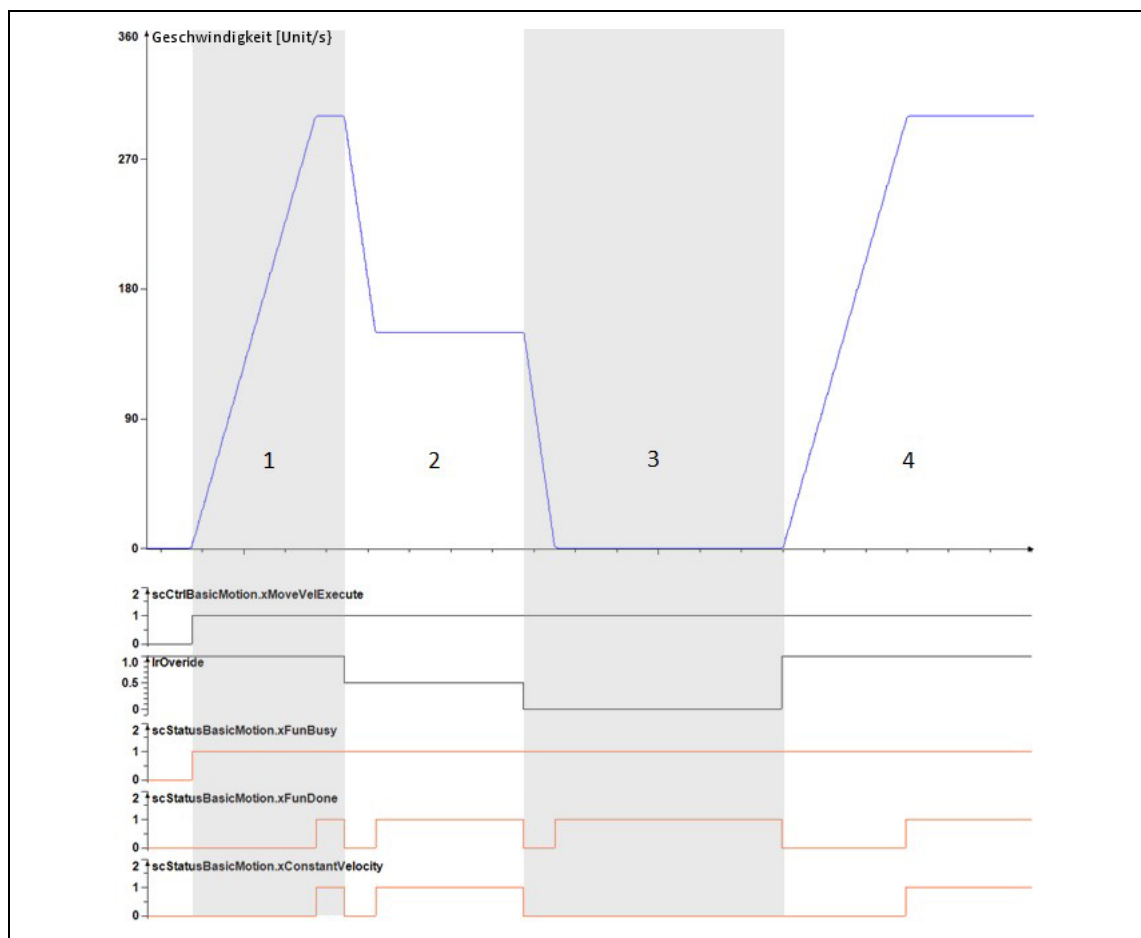
- *IrOverride*, Initialwert: 1.0
- *scCtrlBasicMotion.IrOverrideAcc*, Initialwert: 1.0 (100 %)
- *scCtrlBasicMotion.IrOverrideJerk*, Initialwert: 1.0 (100 %)

Für profilgenerierende Funktionen können separate Overrides für Geschwindigkeit, Beschleunigung, Verzögerung und Ruck eingestellt werden.

Die eingestellten Werte wirken als Faktor und werden mit dem jeweils aktiven Wert multipliziert.

Beispiel

In diesem Beispiel wird die Verwendung des Eingangs *IrOverride* gezeigt.



Nr	xMoveVel Execute	IrOverride	xFunBusy	xFunDone	xConstant Velocity	eTMState	INFO
1	FALSE↗ TRUE	1	FALSE↗ TRUE	FALSE↗ TRUE	FALSE↗ TRUE	STANDBY→ MOVEVEL	Die Achse beschleunigt auf die Zielgeschwindigkeit von 300 units/s.
2	TRUE	0.5	TRUE	TRUE↘ FALSE↗ TRUE	TRUE↘ FALSE↗ TRUE	MOVEVEL	Der Override-Wert wird auf 0.5 gesetzt und die Achse verzögert mit der Profilverzögerung auf 150 units/s
3	TRUE	0	TRUE	TRUE↘ FALSE↗ TRUE	TRUE↘ FALSE	MOVEVEL	Der Override-Wert wird auf 0 gesetzt und die Achse wird mit der Profilverzögerung auf 0 units/s in den Stillstand geführt.
4	TRUE	1	TRUE	TRUE↘ FALSE↗ TRUE	FALSE↗ TRUE	MOVEVEL	Der Override-Wert wird auf 1 gesetzt und die Achse beschleunigt mit der Profilbeschleunigung auf 300 units/s.

9

Beschreibung der Technologiefunktionen

9.1

Festlegung der Wickelrichtung (Aufwickeln/Abwickeln)

Damit die Vorsteuergrößen, die Störgrößenkompensation und das Korrektursignal des Tänzerlagereglers immer in die erforderliche Richtung wirken, ist eine einmalige Festlegung der "normalen Wickelrichtung" erforderlich.

Über den Eingang *xWindingDirection* können Sie einstellen, ob der Wickelantrieb – bezogen auf die normale Materialflussrichtung mit positiver Liniengeschwindigkeit – als Abwickler oder als Aufwickler arbeiten soll.

- *xWindingDirection* = TRUE: Abwickler (Das Material wird abgewickelt.)
- *xWindingDirection* = FALSE: Aufwickler (Das Material wird aufgewickelt.)

⚠️ WARNUNG!

Personenschäden

Verletzungsgefahr besteht durch Änderung der Einstellung am Eingang *xWindingDirection* während einer Bewegung. Dies kann zu einem Drehzahlsprung der Wicklerachse führen.

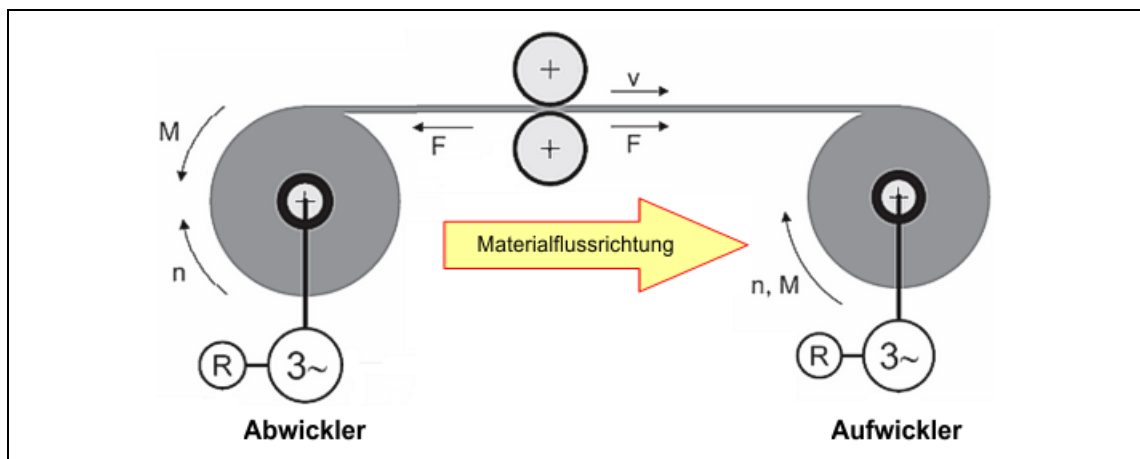
Mögliche Folgen

Tod oder schwere Verletzungen

Schutzmaßnahmen

Die Einstellung des Eingangs *xWindingDirection* nur ändern, wenn das Technologiemodul nicht ausgeführt wird und der Umrichter gesperrt ist:

- Eingang *xEnable* = FALSE
- Eingang *xEnableOperation* = FALSE



[9-1] Wirkrichtung von Drehzahl und Drehmoment in Abhängigkeit des Materialflusses

Automatische Erkennung der Wickelrichtung

Nach Festlegung der Wickelrichtung ist es möglich, die Wickelantriebe mit einer negativen Liniengeschwindigkeit auch in entgegengesetzter Richtung zu betreiben. Ein Eingriff in den Signalfluss ist bei Umkehrung der Materialflussrichtung nicht erforderlich. Die aktuelle Wickelrichtung wird am Ausgang *xUnwind* ausgegeben.

Sonderfall: Erkennung der Wickelrichtung bei stehender Linie

- Abwickler: Eingang *xWindingDirectionStandstill* = TRUE
- Aufwickler: Eingang *xWindingDirectionStandstill* = FALSE.

9.2

Festlegung der Materialzuführung an den Wickler

Über den Eingang *xMaterialFeeding* legen Sie fest, ob das Material von oben oder von unten an den Wickler geführt wird.

► [Funktionsbaustein L_TW2P_WinderDancerCtrlState](#) (17)

⚠ WARNUNG!

Personenschäden

Verletzungsgefahr besteht durch Änderung der Einstellung am Eingang *xMaterialFeeding* während einer Bewegung. Dies kann zu einem Drehzahlsprung der Wicklerachse führen.

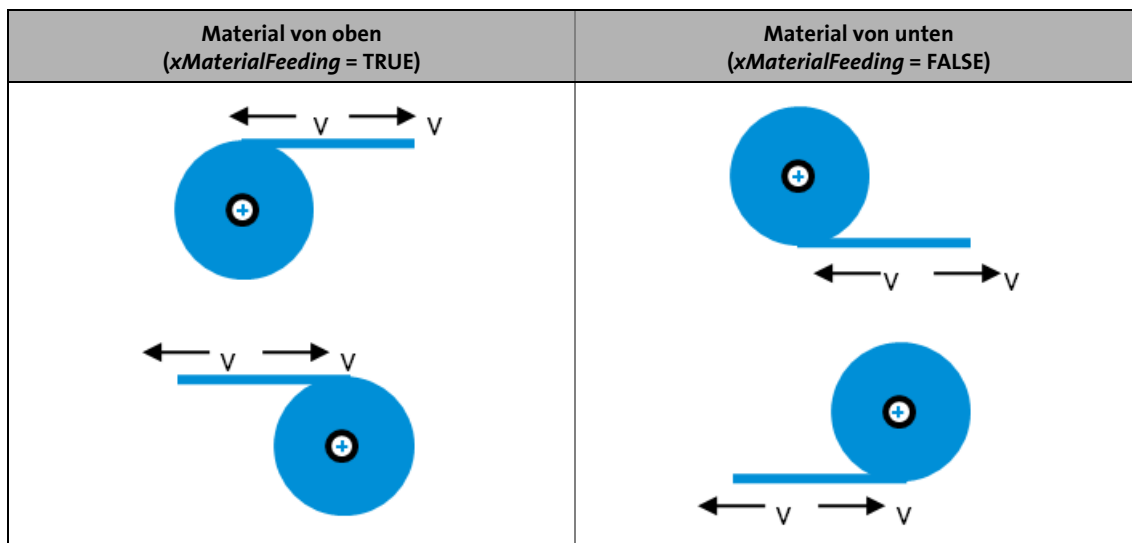
Mögliche Folgen

Tod oder schwere Verletzungen

Schutzmaßnahmen

Die Einstellung des Eingangs *xMaterialFeeding* nur ändern, wenn das Technologiemodul nicht ausgeführt wird und der Umrichter gesperrt ist:

- Eingang *xEnable* = FALSE
- Eingang *xEnableOperation* = FALSE



Die grundsätzliche Anpassung der Wickelwellendrehrichtung an den Materialfluss erfolgt über die Motoranbaurichtung.

Einzustellende Parameter

Die grundsätzliche Anpassung der Wickelwellendrehrichtung an den Materialfluss erfolgt über die Motoranbaurichtung:

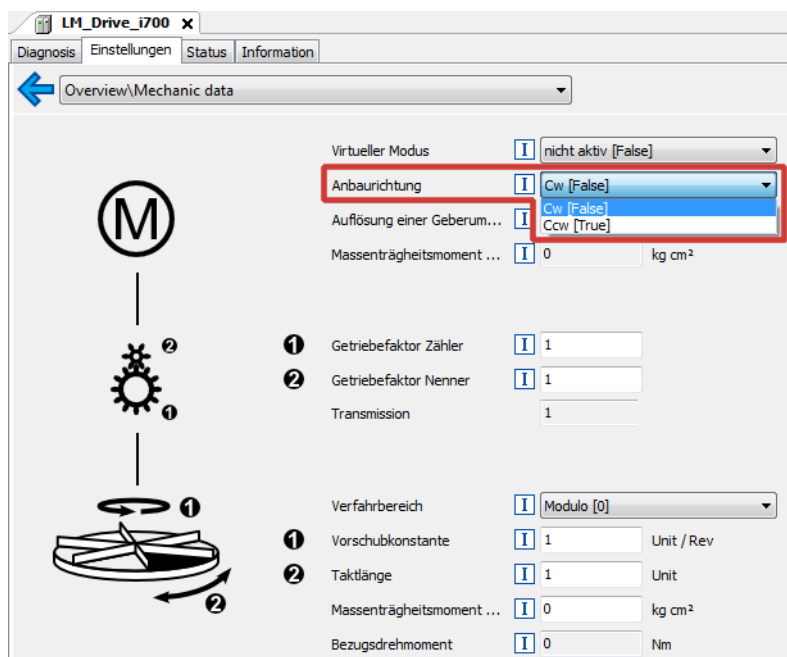
Index:Subindex	Parametername	Display
0x5050:010	Motoranbaurichtung	x

Die Parameter für die Wickelwellendrehrichtung befinden sich in der Parameterstruktur

► [L_TW2P_scPar_WinderDancerCtrlState](#) (26)

Einstellungen im »PLC Designer«

Die Drehrichtung der Achse stellen Sie im »PLC Designer« unter der Registerkarte **Einstellungen** ein:



9.3 Leitwertquelle für die Durchmesserberechnung

Das Technologiemodul arbeitet immer mit der Liniengeschwindigkeit am Eingang *IrSetLineVel*.

Für die Berechnung des Durchmessers kann ein separater Encoder für die Messung der Liniengeschwindigkeit zwischen dem Tänzer und der Wickelachse verwendet werden. In diesem Fall muss die Liniengeschwindigkeit für die Durchmesserberechnung aus dem Encoder am Eingang *IrSetLineVelDiamCalc* verschaltet werden und der Parameter *xLineVelDiamCalc* = TRUE gesetzt werden. Dadurch ist die Berücksichtigung der Tänzerbewegung nicht mehr notwendig – der Parameter *IrDancerMaterialLength* = 0 muss gesetzt werden.

Wird kein Encoder verwendet, so muss der Parameter *xLineVelDiamCalc* = FALSE gesetzt werden. Dadurch wird für die Berechnung des Durchmessers die Liniengeschwindigkeit am Eingang *IrSetLineVel* verwendet. Um die Berechnung des Durchmessers zu optimieren, kann die Tänzerbewegung berücksichtigt werden.

► [Durchmesserberechnung mit Berücksichtigung der Tänzerbewegung](#) (📖 69)

Einzustellende Parameter

Die Parameter für die Durchmesserberechnung mit oder ohne Encoder befinden sich in der Parameterstruktur, siehe Auszug:

Bezeichner	Datentyp	Beschreibung
xLineVelDiamCalc	BOOL	Berechnung des Durchmessers • Initialwert: FALSE
		TRUE Für die Berechnung des Durchmessers wird die Geschwindigkeit aus dem Eingang <i>IrSetLineVelDiamCalc</i> verwendet.
		FALSE Für die Berechnung des Durchmessers wird die Geschwindigkeit aus dem Eingang <i>IrSetLineVel</i> verwendet.
IrDancerMaterialLength	LREAL	Länge des Materials im Tänzer • Einheit: mm • Initialwert: 0 • Mit dem Wert "0" wird die Betrachtung der Tänzerbewegung deaktiviert.

► [L_TW2P_scPar_WinderDancerCtrlState](#) (📖 26)

9.4 Drehzahlvorsteuerung

Die Solldrehzahl für die Drehzahlvorsteuerung wird durch Division der Liniengeschwindigkeit am Eingang *IrSetLineVel* mit dem aktuellen Durchmesser und der Zahl π berechnet:

Berechnung der Solldrehzahl für die Drehzahlvorsteuerung		
$n_{\text{Set}} = \frac{v_{\text{Line}}}{d_{\text{act}} \times \pi}$		
Formelzeichen	Beschreibung	Maßeinheit
n_{Set}	Solldrehzahl für die Drehzahlvorsteuerung	rev/s
v_{Line}	Liniengeschwindigkeit am Eingang <i>IrSetLineVel</i>	mm/s
d_{act}	Aktueller Durchmesser	mm

Damit die Wicklersolldrehzahl mit der Motorsolldrehzahl und dem Liniengeschwindigkeitssignal übereinstimmen, ist die passende Einstellung für die Motorbezugsdrehzahl zwingend erforderlich. Deshalb erfolgt die Berechnung und Parametrierung automatisch und wird nicht dem Anwender überlassen.

Die normierte Wicklersolldrehzahl am Ausgang *IrWndSpdRef* bezieht sich auf die Motordrehzahl, die bei minimalem Durchmesser d_{min} erforderlich ist, um die Bezugsliniengeschwindigkeit am Umfang des Wickelballens zu erreichen.



So prüfen Sie die Drehzahlvorsteuerung:

- Laden Sie den Durchmesserrechner mit dem minimalen Durchmesser (d_{min}):
 - Eingang *IrSetDiam* = 0 (oder $\leq d_{\text{min}}$)
 - Eingang *xLoadDiam* = TRUE
- Bei der Synchronisierung auf die Liniengeschwindigkeit mit dem Eingang *xSyncln* = TRUE folgt die Wicklerachse dem Liniengeschwindigkeitssollwert ausschließlich drehzahl geregelt, ohne dass die Tänzerlage korrigiert wird.
 - [Synchronisierung auf die Liniengeschwindigkeit](#) (81)
- Starten Sie den Liniengeschwindigkeits-Master und erhöhen die Geschwindigkeit, z. B. bis auf 50 %. Der Wickler muss nun mit der Hälfte der Referenzdrehzahl, die am Ausgang *IrWndSpdRef* berechnet wird, drehen.
- Die Umfangsgeschwindigkeit des Wicklers muss nun der Hälfte der Referenz *IrLineVelRef* entsprechen. Das aktuelle Liniengeschwindigkeitssignal wird im Ausgang des Technologiemoduls *IrSetLineVelScaledOut* = 0.5 [x 100 %] = 50 % angezeigt.

Bei falscher Geschwindigkeit oder Drehrichtung prüfen Sie die oben aufgeführte Festlegung der Systemdaten.

9.5 Durchmesserberechnung

Der aktuelle Durchmesser wird durch Division der Liniengeschwindigkeit mit der Wicklerdrehzahl und der Zahl π berechnet:

Berechnung des aktuellen Durchmessers		
$d_{\text{act}} = \frac{v_{\text{Linie}}}{n_{\text{Winder}} \times \pi}$		
Formelzeichen	Beschreibung	Maßeinheit
d_{act}	Aktueller Durchmesser	mm
v_{Linie}	Liniengeschwindigkeit	mm/s
n_{Wickler}	Wicklerdrehzahl	rev/s

Tatsächlich werden bei der Berechnung keine Momentanwerte für die Geschwindigkeit und Drehzahl verwendet sondern aufintegrierte Werte. Hierdurch erfolgt eine Mittelwertbildung. Die Anzahl der Umdrehungen, nach der eine Neuberechnung des Durchmessers erfolgt, wird über den Parameter *IrDiamCalcRegularDist* bestimmt. Der Initialwert dieses Parameters ist auf 1 Wickelwellenumdrehung eingestellt.

Bei dünnen Materialien und unförmigen Wickelballen kann die Mittelwertbildung auf längere Distanzen erweitert werden. Eine geringere Berechnungsdistanz wird mit dem Parameter *IrDiamCalcReducedDist* festgelegt. Initialwert ist hier 1/10 einer Wicklerwellenumdrehung.

Für schnelle Durchmesseränderungen von *IrDiamCalcRegularDist* kann durch das Setzen des Eingangs *xDiamCalcReduced* = TRUE auf den schnellen Berechnungsmodus umgeschaltet werden. Die kleinere Berechnungsdistanz wird mit dem Parameter *IrDiamCalcReducedDist* eingestellt. Als Initialwert muss hier 1/10 Wicklerwellenumdrehung vorgegeben werden.

Diese kleinere Berechnungsdistanz wird auch automatisch durch Laden eines Startdurchmessers aktiviert. Dieser Zustand bleibt solange erhalten, bis ein neuer Durchmesserwert berechnet wurde. Diese Funktion wird benötigt, wenn der reale Durchmesser des Wickelballens von dem geladenen Durchmesser stark abweichen kann. Damit dreht die Wickelwelle nur um eine kurze Distanz mit "falschem" Durchmesser. Nach der Durchmesserberechnung ist wieder ein passender Wert vorhanden.

Als Basis-Parameter müssen der minimale und maximale Durchmesser (Parameter *IrMinDiam*, *IrMaxDiam*) festgelegt werden. Diese Werte wirken einerseits als Begrenzung und andererseits beziehen sich alle prozentualen oder skalierten Durchmesserangaben auf den maximalen Durchmesser.

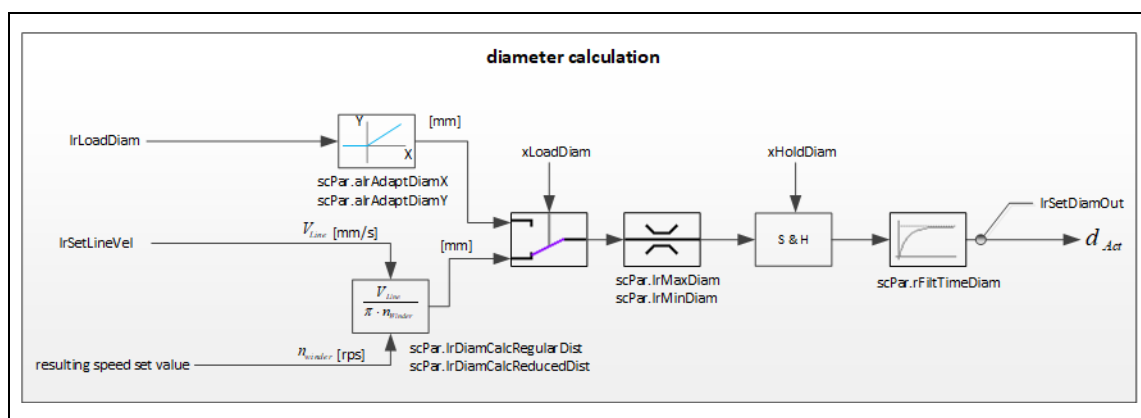
Einzustellende Parameter

Die Parameter für die Durchmesserberechnung befinden sich in der Parameterstruktur, siehe Auszug:

Bezeichner	Datentyp	Beschreibung
IrMaxDiam	LREAL	Maximaler Durchmesser • Einheit: mm • Initialwert: 180
IrMinDiam	LREAL	Minimaler Durchmesser • Einheit: mm • Initialwert: 50
IrDiamCalcRegularDist	LREAL	Reguläre Berechnungsdistanz für Durchmesser • Einheit: rev • Initialwert: 1
IrDiamCalcReducedDist	LREAL	Verkürzte Berechnungsdistanz für Durchmesser • Einheit: rev • Initialwert: 0.1

► [L_TW2P_scPar_WinderDancerCtrlState](#) (26)

Signalfluss zur Berechnung des Durchmessers



9.6

Durchmesser halten

In einigen Betriebszuständen des Wicklers, in denen die Liniengeschwindigkeit nicht der Umfangsgeschwindigkeit des Wickelballens entspricht, kann der aktuelle Durchmesser nicht aus der Liniengeschwindigkeit und der Motordrehzahl berechnet werden. In diesem Fall muss die Berechnung neuer Werte unterbunden werden; der Durchmesserwert wird auf dem alten Wert gehalten.

Wenn der Durchmesser gehalten wird, ist der Ausgang *xHoldDiamActive* = TRUE

Dies erfolgt automatisch bei folgenden Bedingungen:

- Liniengeschwindigkeit < Minimale Liniengeschwindigkeit
- In den Zuständen "STOP", "ERROR", "READY", "JOGGING" und "SYNCLINEVEL".

Für das anwenderseitige Halten des Durchmessers setzen Sie den Eingang *xHoldDiam* = TRUE.

Bei verrauschter Ist-Liniengeschwindigkeit kann die Funktion "Durchmesser halten" toggeln.

Empfehlung: Die Hysteresefunktion nutzen über den Parameter *IrHystlineStandstill*.

Über den Parameter *IrHystlineStandstill* wird die Hysteresenweite festgelegt. Die aktuelle Liniengeschwindigkeit wird über die Hysteresenweite bewertet. Die aus der Hysteresefunktion resultierende Istgeschwindigkeit wird mit der minimalen Liniengeschwindigkeit verglichen.

Einzustellende Parameter

Bezeichner	Datentyp	Beschreibung
IrMinLineVel	LREAL	Minimale Liniengeschwindigkeit Bis zu dieser Geschwindigkeit wird der Durchmesser gehalten. • Einheit: mm/s • Initialwert: 1
xHoldDiam	BOOL	Aktuellen Durchmesser halten Initialwert: FALSE
		TRUE Der aktuelle Durchmesser wird gehalten.
		FALSE Der aktuelle Durchmesser wird nicht gehalten.
IrHystlineStandstill	LREAL	Hysteresenweite für die Liniengeschwindigkeit Dadurch wird die verrauschte Liniengeschwindigkeit und die Richtungserkennung stabilisiert. Nach Ermittlung der maximalen Rauschamplitude des Eingangssignals muss dieser Wert eingestellt werden. Einheit: mm/s Initialwert: 0

► [L_TW2P_scPar_WinderDancerCtrlState](#) (26)

9.7 Durchmesser über einen Sensor vorgeben

Zu Beginn eines Wickelvorgangs kann es erforderlich sein, einen Startdurchmesser vorzugeben oder das Signal eines Durchmessersensors zu verwenden, damit die Sollwertbildung für den Wickelantrieb und auch die Störgrößenkompensation passend starten können.

Mit dem Eingang *IrLoadDiam* können Sie einen Startdurchmesser festlegen, der mit *xLoadDiam* = TRUE mit höchster Priorität übernommen und zyklisch geladen wird.

Ebenso kann ein externer Durchmesserwert, z. B. von einem Ultraschallsensor, auf den Eingang *IrSetDiam* geschaltet werden. Dieser Analogwert kann über eine Kurvenfunktion $Y = f(x)$ adaptiert werden. Die Kurvenfunktion wird mit neun Stützpunkten über die Parameter *alrAdaptDiamX*[1...9] und *alrAdaptDiamY*[1...9] eingestellt. Damit der Analogwert als Startdurchmesser verwendet wird, ist der adaptierte Kurvenverlauf mit *alrAdaptDiamY* = *alrAdaptDiamX* initialisiert. Das Sensorsignal kann auch permanent geladen werden.

In der Praxis werden häufig Ultraschallsensoren mit analoger Ausgangsgröße (0 ... 10 V) zur Erfassung eines Durchmessers eingesetzt. Über eine Kurvenfunktion $Y = f(x)$ können eventuelle Nichtlinearitäten eines solchen Sensors kompensiert werden. Die Funktion wirkt auf den Wert, der über den Messpunkt *MP_IrSetDiamUnfiltered* diagnostiziert werden kann. Die Kurvenfunktion wird mit neun Stützpunkten über die Parameter *alrAdaptDiamX* und *alrAdaptDiamY* eingestellt. Mit der Standardeinstellung der Kurvenfunktion erfolgt keine Adaption. Der Startdurchmesser wird 1:1 durchgereicht.

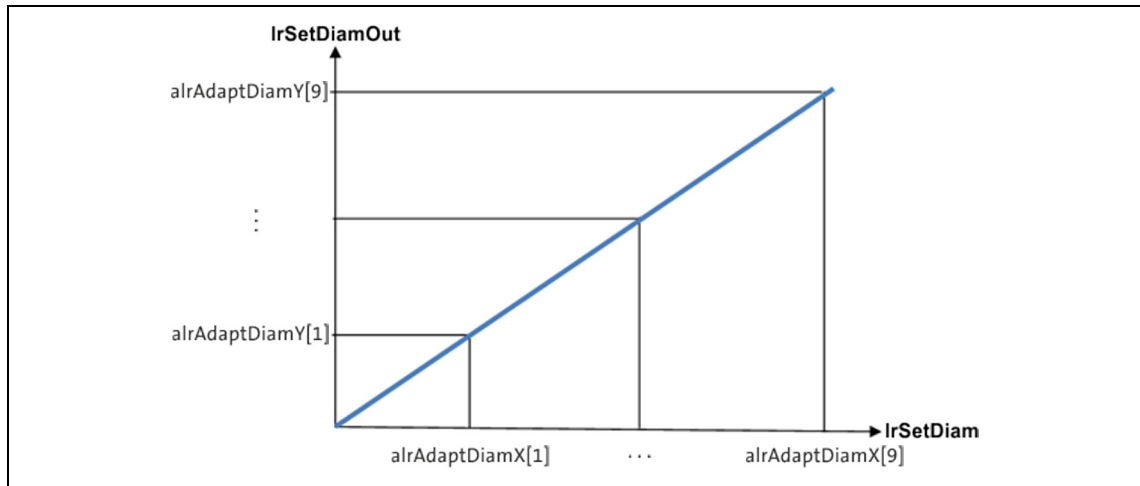
Einzustellende Parameter

Bezeichner	Datentyp	Beschreibung
<i>IrLoadDiam</i>	LREAL	Vorgabe eines (Start-)Durchmessers Der Durchmesser wird zyklisch geladen, wenn der Eingang <i>xLoadDiam</i> = TRUE gesetzt ist. • Einheit: mm • Initialwert: 0
<i>alrAdaptDiamX</i> ARRAY [1...9] OF LREAL		Stützpunkte der Kurvenfunktion für das Laden des Durchmessers • Werte, die am analogen Eingang <i>IrSetDiam</i> anliegen können. • Einheit: mm • Initialwerte: 0, 100, 200, 300, 400, 500, 600, 700, 800
<i>alrAdaptDiamY</i> ARRAY [1...9] OF LREAL		Stützpunkte der Kurvenfunktion für das Laden des Durchmessers • Funktionswerte für den Durchmesser • Einheit: mm • Initialwerte: 0, 100, 200, 300, 400, 500, 600, 700, 800

► [L_TW2P_scPar_WinderDancerCtrlState](#) (26)

Beispiel: Laden eines Durchmessers über eine Kurvenfunktion**Tipp!**

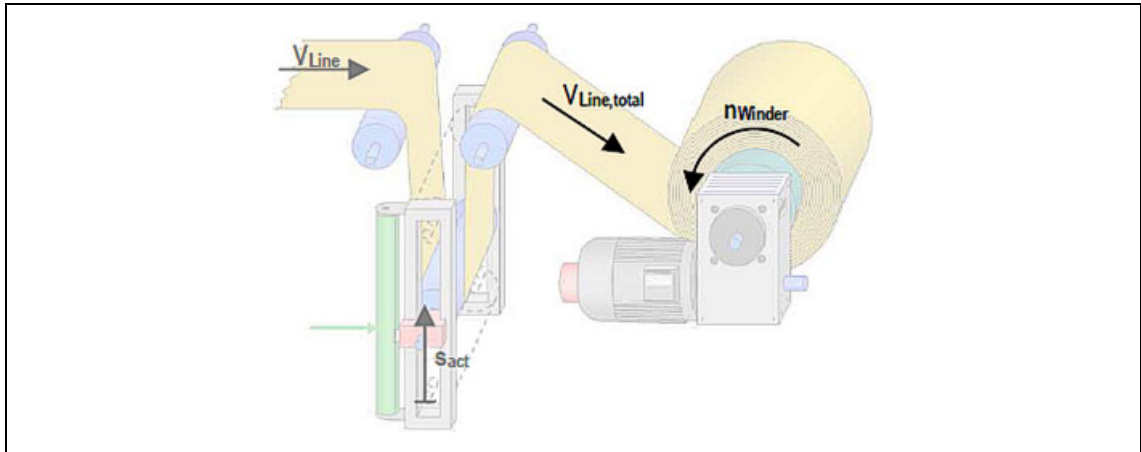
Falls der reale Durchmesser zum Laden nicht bekannt ist (kein Sensor, keine Bedieneingabe möglich), dann sollte ein mittlerer Durchmesser geladen werden.



9.8 Durchmesserberechnung mit Berücksichtigung der Tänzerbewegung

Um die Tänzerposition korrigieren zu können, muss sich der Wickler am Umfang schneller oder langsamer bewegen als die Linie. Wenn hierbei die Umfangsgeschwindigkeit deutlich gegenüber der Liniengeschwindigkeit vor dem Tänzer zu oder abnimmt, dann sollte die resultierende Umfangsgeschwindigkeit $V_{\text{Line;total}}$ für die Durchmesserberechnung verwendet werden.

In der Regel ist das bei Anwendungen der Fall, in denen größere Längen Material in der Tänzermechanik gespeichert werden.



[9-2] Resultierende Umfangsgeschwindigkeit bei Änderung der Tänzerposition

Die Geschwindigkeit, die sich aus der Bewegung des Tänzers ergibt, kann aus der Differenzierung der Tänzerposition ermittelt werden. Die maximal gespeicherte Materiallänge entspricht einer Änderung der Tänzerposition um 200 %.

Speichervolumen im Tänzer

Das Speichervolumen wird mit dem Parameter *IrDancerMaterialLength* definiert und ergibt sich aus dem doppelten Weg zwischen den beiden Grenzlagen multipliziert mit der Anzahl der Materialumschlingungen.

Die Tänzerposition wird über einen PT1-Filter mit der Zeitkonstanten in Parameter *rFiltTimeActDancerVelComp* gefiltert.

Diese Position wird differenziert. Nachfolgend wird die Zusatzgeschwindigkeit aus der Tänzerbewegung mit der eigentlichen Liniengeschwindigkeit für die Durchmesserberechnung *IrSetLineVel* zusammengeführt.

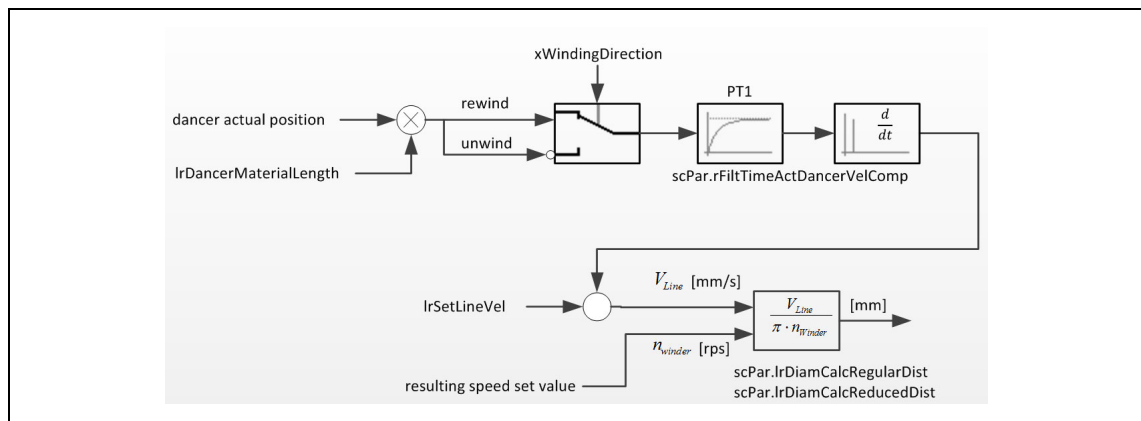
Damit die Kompensation wirksam wird, ist eine passende Materiallänge in Parameter *IrDancerMaterialLength* einzustellen. Mit dem Wert 0 wird keine Zusatzgeschwindigkeit gebildet. Es findet somit keine Kompensation statt.

Einzustellende Parameter

Bezeichner	Datentyp	Beschreibung
IrDancerMaterialLength	LREAL	Länge des Materials im Tänzer • Einheit: mm • Initialwert: 0 • Mit dem Wert 0 wird die Betrachtung der Tänzerbewegung deaktiviert.
rFiltTimeActDancerVelComp	REAL	Filterzeit für die Geschwindigkeitskompensation • Einheit: s • Initialwert: 0

► [L_TW2P_scPar_WinderDancerCtrlState](#) (26)

Signalfluss mit Tänzerkompensation



9.9 Identifikation des Durchmessers durch Anheben des Tänzers

Startet man den Wickelprozess mit einem falschen Durchmesser, kann das zu sehr großen Instabilitäten führen, da weder die Drehzahlvorsteuergröße noch die Adaption der Drehzahlreglerverstärkung passt.

In Anlagen, in denen auf Wicklerrollen mit unbekannten Durchmessern gewechselt wird, sollte man deren Durchmesser vorher identifizieren. Andernfalls kann es zu großen Instabilitäten im Wickelprozess kommen, weil weder die Drehzahl-Vorsteuergröße noch die Drehzahl-Reglerverstärkung passt.

Durch Anheben des Tänzers bei Liniengeschwindigkeit = Null kann ein Wicklerrollen-Durchmesser identifiziert werden, indem die Tänzerbewegung im Liniengeschwindigkeitssignal berücksichtigt wird.

Während der Durchmesser-Identifikation wird bei einer aktiven Adaption der Drehzahlreglerverstärkung ($xAdaptSpdCtrlGain = TRUE$) diese auf den Wert in Parameter $lrIdentDiamSpCtrlGain$ gesetzt.

Voraussetzungen

- Die [State Machine](#) (32) des Technologiemoduls befindet sich im Zustand "STANDBY".
- Die Achse ist freigegeben (Eingang $xRegulatorOn = TRUE$).
- Der Geschwindigkeits-Master der Anlage ist im Stillstand: Liniengeschwindigkeit = 0
- Die im Tänzer gespeicherte Materiallänge, mit dem für die Identifikation verfügbaren Tänzerhub muss ausreichen, um auch bei maximalem Durchmesser eine Wickelwellenbewegung von mindestens dem 2-fachen Durchmesser-Berechnungszyklus (Parameter $lrDiamCalcReducedDist$) durchführen zu können.
- Der Tänzer befindet sich beim Start des Identifizierungslaufs in einer Position, von der aus ausreichend Weg zurückgelegt werden kann, so dass mindestens 1 bis 2 Berechnungszyklen der Durchmesserberechnung durchlaufen werden können, z. B. in der unteren Endlage.
- Die im Tänzer maximal gespeicherte Materiallänge wird in [mm] in Parameter $lrDancerMaterialLength$ eingestellt.
- Die Tänzerendlagen sind passend parametrisiert oder über die Teaching-Funktion angelern worden, so dass die Position am Ausgang $lrActDancerPosScaled$ von -1.0 bis 1.0 (-100 % bis 100 %) skaliert ausgegeben wird.
- Die Bewegungsparameter für die Bewegung während der Identifikation sind passend zu den mechanischen Voraussetzungen eingestellt.

Identifikation starten

Mit einer FALSE→TRUE-Flanke am Eingang $xExecutIdentDiam$ startet die Identifikation. Anschließend wechselt das Technologiemodul in den Zustand "IDENTDIAMETER".

Funktionsablauf

Der Durchmesser wird auf den Startwert von 50 % des maximalen Durchmessers in Parameter *IrMaxDiam* gesetzt. Aufgrund der Unterschreitung der minimalen Liniengeschwindigkeit (Parameter *IrMinLineVel*), wird das interne Halten des Durchmessers deaktiviert.

Wenn die Drehzahlregleradaption durch Setzen des Eingangs *xAdaptSpdCtrlGain* = TRUE aktiviert ist, dann wird die Drehzahlregleradaption auf den Wert aus dem Parameter *IrIdentDiamSpCtrlGain* gesetzt. Die verkürzte Berechnungsdistanz in Parameter *IrDiamCalcReducedDist* wird intern aktiviert.

Die Bewegung der Wicklerachse startet mit den vorgegebenen Profilparametern (Geschwindigkeit, Beschleunigung, Verzögerung und Ruck).



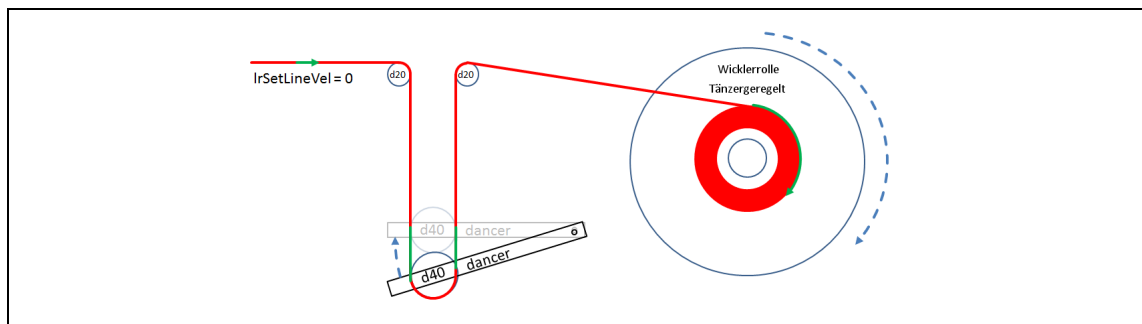
Tipp!

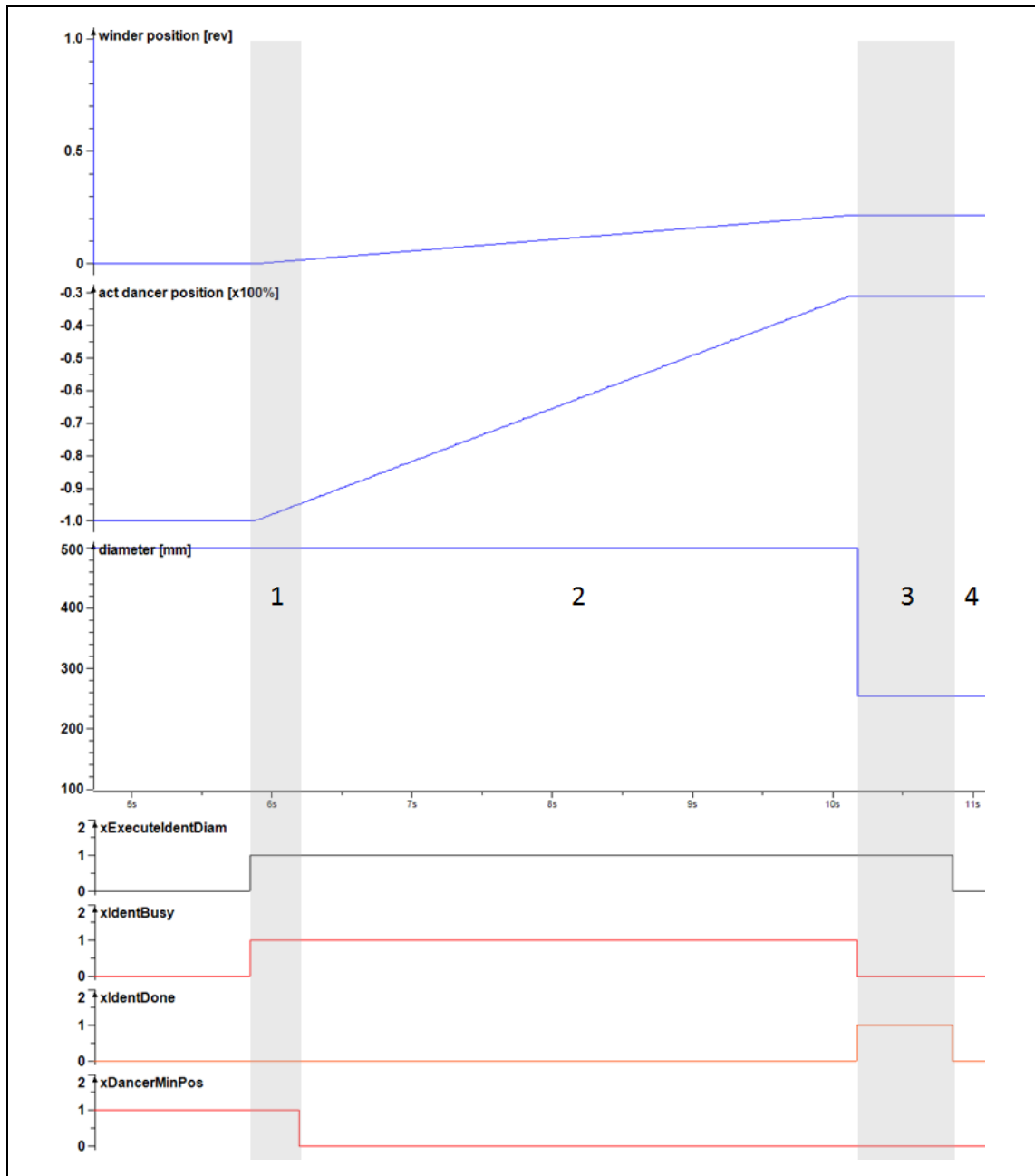
Die Profilparameter beziehen sich auf die Umfangsgeschwindigkeit des Wicklers.

- Geschwindigkeit (*IrIdentDiamVel*) in [mm/s]
- Beschleunigung (*IrIdentDiamAcc*) in [mm/s²]
- Verzögerung (*IrIdentDiamDec*) in [mm/s²]
- Ruck (*IrIdentDiamJerk*) in [mm/s³]

Die aktive Identifikation wird am Status **IdentBusy** = TRUE angezeigt. Während der Identifikation wird durch das Aufwickeln des Wickelballens der Tänzer angehoben. Die Identifikation war erfolgreich, wenn an der Wickelwelle der vorgegebene Durchmesserberechnungszyklus mit der erwarteten Häufigkeit aus Parameter *IrDiamCalcReducedDist* durchlaufen werden konnte. Der Status **IdentBusy** wird dann wieder auf FALSE gesetzt.

Nach erfolgreicher Identifikation kann der Tänzer sofort in die Sollposition geregelt werden, wenn *xDancerCtrl* = TRUE.





Nr	xExecutIdentDiam	xIdentBusy	xIdentDone	xDancerMinPos	eTMState	INFO
1	FALSE → TRUE	FALSE → TRUE	FALSE	TRUE	STANDBY → IDENTDIAMETER	Die Identifizierung des Durchmessers wird beauftragt. Der Aufwickler (von oben) wickelt das Material auf. Somit wird der Tänzer angehoben. Die untere Tänzerendlage ist deaktiviert und dadurch wird die Berechnung des Durchmessers inaktiv.
2	TRUE	TRUE	FALSE	TRUE → FALSE	IDENTDIAMETER	Die untere Tänzerendlage wird inaktiv. Damit wird die Berechnung des Durchmessers über die verkürzte Berechnungsdistanz aktiviert.
3	TRUE	TRUE → FALSE	FALSE → TRUE	FALSE	IDENTDIAMETER → STANDBY	2x verkürzte Berechnungsdistanzen sind von der Wicklerachse abgefahren. Damit wird die Identifizierung beendet. Über den Hub des Tänzers und über die Vorgabe des gespeicherten Materials im Tänzer wird die Länge des aufgewickelten Materials bestimmt. Über die Länge des aufgewickelten Materials und über den Weg der Wicklerachse wird der Durchmesser bestimmt. Der Status Done wird gesetzt.
4	FALSE → TRUE	FALSE	TRUE → FALSE	FALSE	STANDBY	Durch das Rücksetzen des Auftrages für die Identifizierung des Durchmessers wird der Status Done für die Identifizierung ebenfalls zurückgesetzt.

Abbruch Identifikation

Wird während der Identifikation die maximale Tänzerposition in Parameter *IrIdentDiamMaxDancerPos* oder die obere Tänzerendlage in Parameter *IrDancerMaxPosScaled* überschritten, ohne dass die Identifikation erfolgreich abgeschlossen werden konnte, wird die Bewegung mit den Schnellhalt-Parametern gestoppt und der Zustand "ERROR" wird gemeldet (*xError* = TRUE, *eErrorID* = 17165 "ActDancerPosGTMaxDancerPos").

Der anliegende Fehler kann über den Eingang *xResetError* quittiert werden. Das Technologiemodul wechselt in den Zustand "READY", wenn kein Fehler anliegt.

9.10 Materiallängenzähler

Freigabe des Materiallängenzählers

Der Materiallängenzähler wird mit dem Eingang *xEnable* = TRUE aktiviert.

Die Materiallänge wird durch Integration der Liniengeschwindigkeit am Eingang *IrSetLineVel* berechnet und am Ausgang *IrMaterialCounter* (in Millimeter) angezeigt. Je nach Festlegung der Wickelrichtung wird die Materiallänge hoch- oder runtergezählt.

► [Festlegung der Wickelrichtung \(Aufwickeln/Abwickeln\)](#) (📖 58)

Für das Analogsignal der Liniengeschwindigkeit kann der aktuelle Wert der Materiallänge mit einer PT1-Charakteristik gefiltert werden. Die Filterzeit wird mit dem Parameter *rFiltTimeMaterialCounter* eingestellt.

Der aktuelle Wert der Materiallänge wird in den persistenten Daten gespeichert.

► [L_TW2P_PersistentVarWinder](#) (📖 97)

Vorladen des Materiallängenzählers

Für die Initialisierung der Materiallänge kann über den Parameter *IrSetMaterialPos* eine Anfangsmateriallänge eingestellt werden. Mit einer FALSE→TRUE-Flanke am Eingang *xSetMaterialCounter* wird die Anfangsmateriallänge mit höchster Priorität übernommen.

Einzustellende Parameter

Bezeichner	Datentyp	Beschreibung
IrSetMaterialPos	LREAL	Position des Materiallängenzählers Mit einerFALSE→TRUE-Flanke am Eingang <i>xSetMaterialCounter</i> wird der Materiallängenzähler (Ausgang <i>IrMaterialCounter</i>) auf den Wert in <i>IrSetMaterialPos</i> gesetzt. • Einheit: mm • Initialwert: 0
rFiltTimeMaterialCounter	REAL	Filterzeitkonstante für den Materiallängenzähler (Ausgang <i>IrMaterialCounter</i>) • Initialwert: 0 (Filter ist deaktiviert.)

► [L_TW2P_scPar_WinderDancerCtrlState](#) (📖 26)

Quellen für die Materiallängenzählung

Die Materiallängenzählung kann aus einer von drei unterschiedlichen Quellen und nach zwei unterschiedlichen Verfahren erfolgen.

Quelle	Auswahl der Quelle		INFO
	Anschluss an <i>MaterialCounterAxis</i>	<i>xLineVelDiamCalc</i>	
IrSetLineVel	nein	FALSE	Materiallängenzähler über Eingang 'IrSetLineVel'
IrSetLineVelDiamCalc	nein	TRUE	Materiallängenzähler über Eingang 'IrSetLineVelDiamCalc'
MaterialCounterAxis	ja	wird nicht gewertet	Materiallängenzähler über Referenzachse 'MaterialCounterAxis'

9.10.1 Materiallängenzähler über Eingang 'IrSetLineVel'

Voraussetzungen

- Es ist keine Referenzachse am Eingang *MaterialCounterAxis* angeschlossen.
- Parameter *xLineVelDiamCalc* = FALSE

Funktionsweise

In die Materiallängenzählung wird zusätzlich zur Position (*IrSetMaterialPos*) die Liniengeschwindigkeit am Eingang *IrSetLineVel* integriert.

Der resultierende Wert wird als Materiallänge am Ausgang *IrMaterialCounter* angezeigt und persistent gespeichert.

Mit einer FALSE→TRUE-Flanke am Eingang *xSetMaterialCounter* wird die Materiallänge aus dem Parameter *IrSetMaterialPos* geladen. Dabei wird die Materiallänge am Ausgang *IrMaterialCounter* direkt auf den Wert von *IrSetMaterialPos* gesetzt. Die weitere Zählung wird auf den am Ausgang gesetzten Wert der Materiallänge addiert.

Bei einem verrauschten Signal wird die Materialzählung durch die Integration der Liniengeschwindigkeit verfälscht. Hierbei kann eine Drift des Materiallängenzählers beobachtet werden, auch wenn die Linie steht.

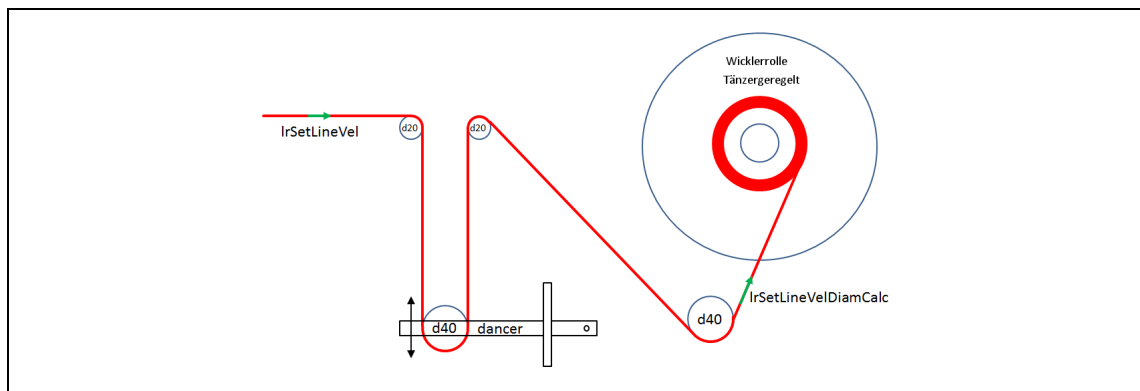
9.10.2 Materiallängenzähler über Eingang 'IrSetLineVelDiamCalc'

Voraussetzungen

- Es ist keine Referenzachse am Eingang *MaterialCounterAxis* angeschlossen.
- Parameter *xLineVelDiamCalc* = FALSE

Funktionsweise

In die Materiallängenzählung wird zusätzlich zur Position (*IrSetMaterialPos*) die Liniengeschwindigkeit am Eingang *IrSetLineVelDiamCalc* integriert. Mit dem Eingang sollte die Geschwindigkeit unmittelbar vor dem Wickelballen und hinter dem Tänzer ausgemessen werden. Damit wird der Einfluss der Tänzerbewegung für die Materiallängenzählung stark reduziert.



Der resultierende Wert wird als Materiallänge am Ausgang *IrMaterialCounter* angezeigt und persistent gespeichert.

Mit einer FALSE→TRUE-Flanke am Eingang *xSetMaterialCounter* wird die Materiallänge aus dem Parameter *IrSetMaterialPos* geladen. Dabei wird die Materiallänge am Ausgang *IrMaterialCounter* direkt auf den Wert von *IrSetMaterialPos* gesetzt. Die weitere Zählung wird auf den am Ausgang gesetzten Wert der Materiallänge addiert.

Bei einem verrauschten Signal wird die Materialzählung durch die Integration der Liniengeschwindigkeit verfälscht. Hierbei kann eine Drift des Materiallängenzählers beobachtet werden, auch wenn die Linie steht.

9.10.3 Materiallängenzähler über Referenzachse 'MaterialCounterAxis'

Voraussetzung

- Eine Referenzachse (Modulo-Achse) ist am Eingang *MaterialCounterAxis* angeschlossen.

Funktionsweise

Als Basis wird die Ermittlung der verlustfreien Anzahl der Umdrehungen für die Materiallängenzählung verwendet. Dieses Verfahren eignet sich für verauschte Signale.

Hierbei werden ganze Umdrehungen und der Bruchteil einer Umdrehung erfasst. Über die Vorschubkonstante der Referenzachse (Modulo-Achse) wird die Materiallänge am Ausgang *IrMaterialCounter* angezeigt.

Die Anzahl der gezählten Umdrehungen kann über den Messpunkt *MP20_IrRevCounter* ausgelesen werden. Der Bruchteil einer Umdrehung wird über den Messpunkt *MP21_IrRevCounterResidual* angezeigt. Die Werte dieser Messpunkte werden persistent gespeichert.

Mit einer FALSE↗TRUE-Flanke am Eingang *xSetMaterialCounter* wird die Materiallänge aus dem Parameter *IrSetMaterialPos* geladen. Dabei wird die Materiallänge über die Vorschubkonstante der Achse in die Anzahl der Umdrehungen umgerechnet und gespeichert.

Die Materiallänge aus dem Parameter *IrSetMaterialPos* wird am Ausgang *IrMaterialCounter* ausgegeben.

Eine genaue Materiallängenzählung kann nur bei einem schlupffreien Messrad erfolgen. Ein schlupfbehaftetes Messrad auf dem Material führt zur Fehlern in der Materiallängenzählung.

9.11 Tippbetrieb mit 1/d-Bewertung



Tipp!

Die Begrenzungsfunktion "Sichere Geschwindigkeit" (SLS1, SLS2 ...) kann die Geschwindigkeit aus der Funktion "Handfahren" (*xLineJogPos*, *xLineJogNeg*) begrenzen, wenn der Parameter "160" (Follower SLS Reaction) der Wicklerachse auf TRUE gesetzt ist.

Bei diesem Betrieb wird ein manuelles Tippen der Linie ausgeführt. Die Funktion unterscheidet sich zur Basic-Motion-Funktion "Handfahren". Die Vorschubkonstante in [mm/rev] ist beim Wickler nicht konstant, da sich der Umfang des Wicklers mit dem aktuellen Durchmesser ändert. Die tatsächliche Drehzahl der Wickelwelle hängt also nicht nur von der parametrisierten Handfahrgeschwindigkeit ab, sondern auch vom aktuellen Durchmesser. Der aktuelle Durchmesser, Ausgang *IrSetDiamOut*, sollte also dem realen Durchmesser entsprechen, wenn der Tippbetrieb angefordert wird, ansonsten könnte die Wickelwelle mit einer nicht erwarteten Drehzahl bewegt werden.

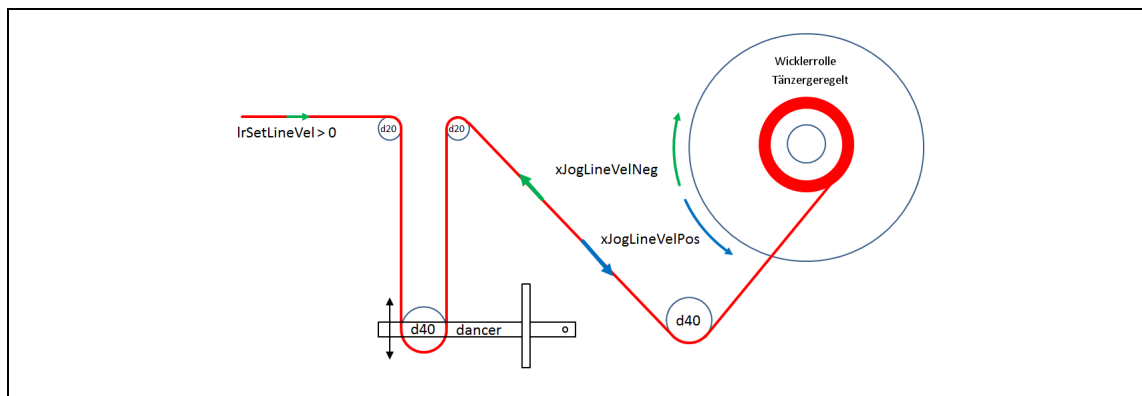
'Positiv Tippen' bedeutet, dass das Material in Materialflussrichtung bewegt wird.

'Negativ Tippen' bedeutet, dass das Material gegen Materialflussrichtung bewegt wird.

Die Drehrichtung der Wickelwelle wird also dabei auch von der Materialführung beeinflusst, die über den Eingang *xMaterialFeeding* festgelegt wird.

Mit dem Eingang *xJogLinePos* = TRUE wird die Linie in positive Richtung und mit dem Eingang *xJogLineNeg* = TRUE in negative Richtung gefahren. Die Linie wird solange gefahren, wie der Eingang TRUE gesetzt bleibt. Der laufende Fahrbefehl kann nicht durch den anderen Jog-Befehl abgelöst werden.

Die parametrierbaren Sollwerte *IrJogLineVel*, *IrJogLineAcc* und *IrJogLineDec* für das Handfahren beziehen sich auf die Umfangsgeschwindigkeit oder Liniengeschwindigkeit und nicht auf die Motordrehzahl.



Einzustellende Parameter

Bezeichner	Datentyp	Beschreibung
IrJogLineAcc	LREAL	Beschleunigung für den Tippbetrieb mit der 1/d-Bewertung. Vorgabe, mit welcher Geschwindigkeitsänderung maximal beschleunigt werden soll. • Einheit: mm/s ² • Initialwert: 100
IrJogLineDec	LREAL	Verzögerung für den Tippbetrieb mit der 1/d-Bewertung. Vorgabe, mit welcher Geschwindigkeitsänderung maximal verzögert werden soll. • Einheit: mm/s ² • Initialwert: 10
IrJogLineVel	LREAL	Geschwindigkeit für den Tippbetrieb mit der 1/d-Bewertung. Vorgabe, mit welcher Geschwindigkeit maximal verfahren werden soll. • Einheit: mm/s • Initialwert: 10

► [L_TW2P_scPar_WinderDancerCtrlState](#) (26)

9.12 Synchronisierung auf die Liniengeschwindigkeit

Im Zuge der Inbetriebnahme, noch ohne Material, kann eine einfache Geschwindigkeitsfolger-Funktion genutzt werden. Hierbei kann geprüft werden, ob die Achse dem Leitwert prinzipiell mit der erwarteten Drehzahl, auch bei sich änderndem Durchmesser, folgt. Als Prüfmittel wird in der Praxis häufig ein Hand-Tacho am Umfang des Wicklers kontaktiert, der die tatsächliche Umfangsgeschwindigkeit anzeigt. Diese kann dann mit der angezeigten *IrSetLineVel* verglichen werden. Der aktuelle Durchmesser *IrSetDiamOut* sollte dem realen Durchmesser entsprechen, wenn die Geschwindigkeitsfolger-Funktion angefordert wird, ansonsten könnte die Wickelwelle mit einer nicht erwarteten Drehzahl bewegt werden.

Die Synchronisierung der Wicklerachse auf die Liniengeschwindigkeit wird mit *xSyncln* = TRUE ausgeführt.

Die Parameter *IrSyncLineAcc* und *IrSyncLineDec* beziehen sich auf die Umfangsgeschwindigkeit oder Liniengeschwindigkeit und nicht auf die Motordrehzahl.

Einzustellende Parameter

Bezeichner	Datentyp	Beschreibung
<i>IrSyncLineAcc</i>	LREAL	Beschleunigung zur Synchronisierung auf die Liniengeschwindigkeit • Einheit: mm/s² • Initialwert: 100
<i>IrSyncLineDec</i>	LREAL	Verzögerung zur Synchronisierung auf die Liniengeschwindigkeit • Einheit: mm/s² • Initialwert: 100

► [L_TW2P_scPar_WinderDancerCtrlState](#) (26)

9.13 Normierung der Tänzerlage

Die Istposition des Tänzers wird in Form eines analogen Signals (0 ... 10 V) an den Controller zurückgeführt. Das analoge Signal muss am Eingang *IrActDancerPosIn* anliegen.

Die Tänzerposition wird über einen PT1-Filter mit der Zeitkonstante *rFiltTimeActDancerPosIn* gefiltert. Die gefilterte Position wird von -1 bis 1 skaliert und am Ausgang *IrActDancerPosScaled* ausgegeben:

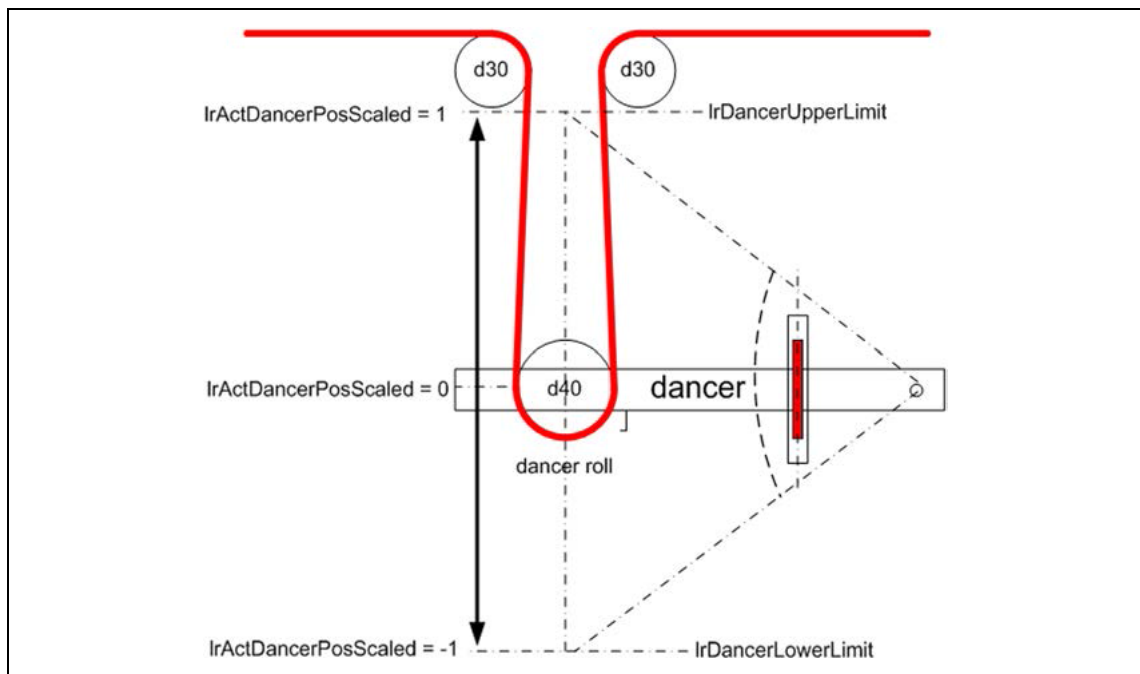
- Die untere Tänzerendlage entspricht *IrActDancerPosScaled* = -1.
- Die obere Tänzerendlage entspricht *IrActDancerPosScaled* = 1.
- Im Tänzer ist die minimale und maximale Materiallänge gespeichert.

Dadurch wird die Vorgabe des Sollwertes und die Überwachung der Tänzerposition vereinfacht.

Quellen zur Festlegung der Tänzerendlagen

Die Wicklertechnologie hat insgesamt zwei Quellen für die Festlegung der Tänzerendlagen.

- Quelle 1: Vorgabe der Tänzerendlagen über Parameter.
Für die Vorgabe der Tänzerendlagen über Parameter muss der Parameter *xTechDancerLimits* = FALSE gesetzt werden. Mit dieser Einstellung werden die Grenzwerte für die obere und untere Tänzerposition (Tänzerendlagen) über die Parameter *IrDancerUpperLimit* und *IrDancerLowerLimit* vorgegeben.
- Quelle 2: Die Tänzerendlagen werden über die Teaching-Funktion erlernt.
Für die Teaching-Funktion der Tänzerendlagen muss der Parameter *xTechDancerLimits* = TRUE gesetzt werden. In dieser Einstellung werden die erlernten Tänzerendlagen verwendet, sobald die Werte für die obere und untere Tänzerendlage aus der Teaching-Funktion zur Verfügung stehen. Solange keine erlernten Werte zur Verfügung stehen, werden immer die Werte für die Tänzerendlagen aus den Parametern vorgegeben.

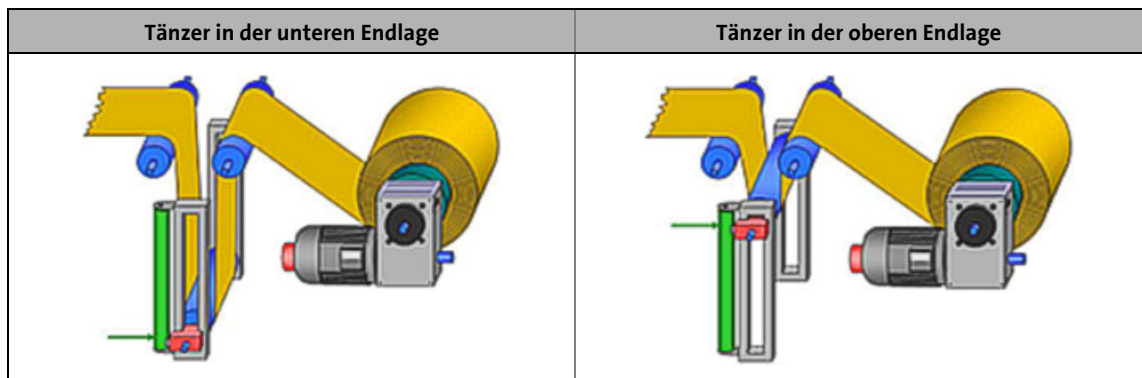


Einzustellende Parameter

Bezeichner	Datentyp	Beschreibung
xTeachDancerLimits	BOOL	Quelle der Tänzerlagebegrenzungen • Initialwert: FALSE
		TRUE Die Endlagen <i>IrDancerLowerLimit</i> und <i>IrDancerUpperLimit</i> werden solange verwendet, bis die Teaching-Funktion ausgeführt wurde. Nach manueller Ausführung der Teaching-Funktion werden immer die gespeicherten Endlagen aus der Teaching-Funktion verwendet.
		FALSE Die Endlagen <i>IrDancerLowerLimit</i> und <i>IrDancerUpperLimit</i> werden verwendet.
IrDancerUpperLimit	LREAL	Analoger Wert für die obere Tänzerlagegrenze • Initialwert: 10 000 000
IrDancerLowerLimit	LREAL	Analoger Wert für die untere Tänzerlagegrenze • Initialwert: 0
rFiltTimeActDancerPosIn	REAL	PT1-Filterzeit für den Eingang <i>IrActDancerPosIn</i> • Einheit: s • Initialwert: 0.005

► [L_TW2P_scPar_WinderDancerCtrlState](#) (26)

9.14 Teaching-Funktion für Tänzerendlagen



So führen Sie die Teaching-Funktion manuell aus:

Hinweis: Bereits erlernte Endlagen werden bei Ausführung der Teaching-Funktion überschrieben.

1. Den Parameter `xTeachDancerLimits` = TRUE setzen.
2. Den Tänzer manuell in die untere Endlage bewegen, zur Vorgabe für maximales Material im Tänzer.
3. Den Eingang `xTeachLowerPos` = TRUE setzen.
Bei steigender Flanke (`FALSE` → `TRUE`) wird der aktuelle Eingangswert `IrActDancerPosIn` gespeichert. Ist der Eingangswert über den Eingang `PersistentVar` verschaltet, wird er in den Retain/Persistent-Speicher in die Variable `IrDancerLowerLimit` geschrieben.
4. Den Tänzer manuell in die obere Endlage bewegen, zur Vorgabe für minimales Material im Tänzer.
5. Den Eingang `xTeachUpperPos` = TRUE setzen.
Bei steigender Flanke (`FALSE` → `TRUE`) wird der aktuelle Eingangswert `IrActDancerPosIn` gespeichert. Ist der Eingangswert über den Eingang `PersistentVar` verschaltet, wird er in den Retain/Persistent-Speicher in die Variable `IrDancerUpperLimit` geschrieben.
6. Nach Abschluss der Teaching-Funktion, die Plausibilität der persistenten Variablen prüfen.
▶ [L TW2P PersistentVarWinder](#) (97)



So prüfen Sie die erlernten Endlagen über die Tänzerlageregelung:

1. Die Reglerfreigabe mit dem Eingang *xEnableOperation* = TRUE setzen.
2. Die Tänzerlageregelung mit dem Eingang *xDancerCtrl* = TRUE aktivieren.
Hinweis: Das Material muss sich im Stillstand befinden.
3. Den skalierten Sollwert für die Tänzerposition am Eingang *IrSetDancerPosScaled* = 1 setzen.
 - Prüfen, ob der Tänzer die obere Endlage erreicht hat.
 - Wurde die obere Endlage nicht erreicht, muss die Teaching-Funktion erneut ausgeführt werden.
4. Den skalierten Sollwert für die Tänzerposition am Eingang *IrSetDancerPosScaled* = -1 setzen.
 - Prüfen, ob der Tänzer die untere Endlage erreicht hat.
 - Wurde die obere Endlage nicht erreicht, muss die Teaching-Funktion erneut ausgeführt werden.

Alternativ zur Teaching-Funktion können Sie die jeweiligen Eingangswerte manuell in die Parameter *IrDancerUpperLimit* und *IrDancerLowerLimit* eintragen.

Einzustellende Parameter

Bezeichner	Datentyp	Beschreibung
xTeachDancerLimits	BOOL	Quelle der Tänzerlagebegrenzungen • Initialwert: FALSE
		TRUE Die Endlagen <i>IrDancerLowerLimit</i> und <i>IrDancerUpperLimit</i> werden solange verwendet, bis die Teaching-Funktion ausgeführt wurde. Nach manueller Ausführung der Teaching-Funktion werden immer die gespeicherten Endlagen aus der Teaching-Funktion verwendet.
		FALSE Die Endlagen <i>IrDancerLowerLimit</i> und <i>IrDancerUpperLimit</i> werden verwendet.
IrDancerUpperLimit	LREAL	Analoger Wert für die obere Tänzerlagegrenze • Initialwert: 10 000 000
IrDancerLowerLimit	LREAL	Analoger Wert für die untere Tänzerlagegrenze • Initialwert: 0
xUpperPosTeached	BOOL	Referenzierte obere Tänzerendlage
		TRUE Die obere Tänzerendlage bereits referenziert.
xLowerPosTeached	BOOL	Referenzierte obere Tänzerendlage
		TRUE Die untere Tänzerendlage bereits referenziert.

► [L_TW2P_scPar_WinderDancerCtrlState](#) (26)

► [L_TW2P_PersistentVarWinder](#) (97)

9.15 Überwachung der Tänzerposition

Für den Betrieb des Wicklers ist die Überwachung der Tänzerposition in folgenden Fällen von Bedeutung:

- Nach Freigabe des Tänzerlagereglers sollte die Maschine erst gestartet werden, wenn sich der Tänzer in der Sollposition befindet: Ausgang *xDancerReachedSetPos* = TRUE.
- Kommt der Tänzer in den Endlagenbereich, kann es im laufenden Betrieb zu einem Bahnrisso kommen. Dies ist wahrscheinlich dann der Fall, wenn sich der Tänzer für längere Zeit in der unteren Endlage befindet.
 - Ausgang *xDancerMinPos* = TRUE: Die untere Endlage ist erreicht.
 - Ausgang *xDancerMaxPos* = TRUE: Die obere Endlage ist erreicht.

Einzustellende Parameter

Bezeichner	Datentyp	Beschreibung
IrDancerMaxPosScaled	LREAL	Maximale Tänzerposition für das Statusbit <i>xDancerMaxPos</i> • Einheit: x 100 % (1 = 100 %) • Initialwert: 0.95 (95 %)
IrDancerMinPosScaled	LREAL	Minimale Tänzerposition für das Statusbit <i>xDancerMinPos</i> • Einheit: x 100 % (1 = 100 %) • Initialwert: -0.95 (-95 %)
IrDancerInPosWindowScaled	LREAL	Fenster für die Sollposition des Tänzers, um das Statusbit <i>xDancerReachedSetPos</i> anzusteuern. • Einheit: x 100 % (1.0 = 100 %) • Initialwert: 0.2 (20 %)

► [L_TW2P_scPar_WinderDancerCtrlState](#) (26)

9.16 PI-Regler für die Tänzerlageregelung

Bei der Tänzerlageregelung wird der Antrieb in Drehzahlregelung betrieben. Durch Aktivierung der Tänzerlageregelung mit dem Eingang $x\text{DancerCtrl} = \text{TRUE}$ wird auf einen Tänzerpositions-Istwert geregelt.

Zur Vorsteuerung dient das Liniengeschwindigkeitssignal multipliziert mit dem Kehrwert des Durchmessers. Die Tänzerposition wird erfasst und mit der Sollposition verglichen. Bei einer Abweichung sorgt der Tänzerlageregler für eine Korrektur des Drehzahlsollwertes.

Der Tänzerregler arbeitet nach einer klassischen PI-Struktur. Die Regelung wird freigegeben mit $x\text{DancerCtrl} = \text{TRUE}$. Solange der Eingang $x\text{DancerCtrl} = \text{FALSE}$ ist, bleibt die Regelabweichung Null und auch der Rampengenerator für den Tänzerlagesollwert wird mit dem Istwert der Tänzerendlage geladen.

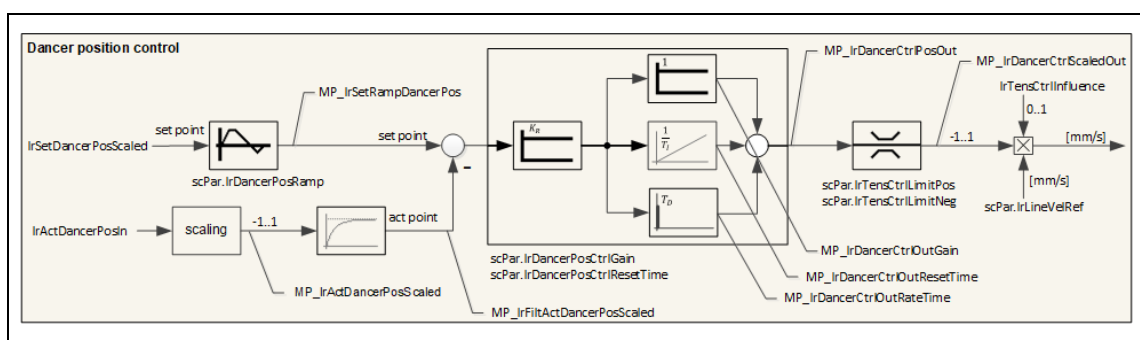
Bei begrenzt aufgelöstem Tänzeristwert-Signal des Eingangs $Ir\text{ActDancerPosIn}$ kann sich eine PT1-Filterung positiv auf die Stabilität des Regelkreises auswirken. Andererseits sollten Totzeiten im Regelkreis vermieden werden, wenn hohe Dynamik verlangt ist. Die Filterzeit ist mit dem Parameter $r\text{FiltTimeActDancerPosIn}$ einstellbar (Standardeinstellung: 5 ms).

Die Stellgröße des PI-Reglers wird noch über einen Limiter geführt, so dass eine Begrenzung über die Parameter $Ir\text{DancerPosCtrlLimitPos}$ und $Ir\text{DancerPosCtrlLimitNeg}$ möglich ist.

Die Stellgröße wirkt additiv im Hauptdrehzahl-Sollwertpfad. Der Einfluss über den Eingang $Ir\text{DancerPosInfluenceScaled}$ wirkt multilikativ auf die Stellgröße des PI-Reglers, damit wird also festgelegt, wie stark die Korrekturgröße wirken darf.

Die Reglerverstärkung wird mit dem Parameter $Ir\text{DancerPosCtrlGain}$ eingestellt. Der I-Anteil des PI-Reglers kann mit dem Parameter $Ir\text{DancerPosCtrlResetTime}$ (Reglernachstellzeit) gesetzt werden. In der Standardeinstellung ist $Ir\text{DancerPosCtrlResetTime} = 0$ (deaktiviert) gesetzt.

Nach Aktivierung der Tänzerlageregelung muss der Tänzer erst in die Sollposition gebracht werden. Damit der Tänzer kontrolliert angehoben wird, wird zuvor der Rampengenerator für den Positionssollwert mit dem aktuellen Positionsiswert geladen. Die Rampe wird mit dem Parameter $Ir\text{DancerPosRamp}$ eingestellt (Standardeinstellung $1 = 100\%/s$). Eine Einblendung des Tänzerreglereinflusses ist dadurch nicht erforderlich. Die Standardeinstellung in Parameter $Ir\text{DancerPosRamp} = 1$ bedeutet, dass der Sollwert von 0 bis 1 innerhalb einer Sekunde geführt wird. Im Wickelbetrieb wird jede Änderung des Sollwertes am Eingang $Ir\text{SetDancerPosScaled}$ über den Rampengenerator geführt.



[9-3] Signalfluss PI-Tänzerlageregler

Mit dem Eingang $x\text{ResetCtrl} = \text{TRUE}$ wird der I-Anteil des PI-Reglers ausgeschaltet und die Stellgröße (Ausgang des Reglers) aus dem I-Anteil über die Rampenfunktion auf '0' geführt. Die Stellgröße aus dem P-Anteil wird nicht beeinflusst.

Einzustellende Parameter

Bezeichner	Datentyp	Beschreibung
IrDancerPosRamp	LREAL	Beschleunigungsrampe für die Tänzerlagesollwerte • Einheit: 1/s • Initialwert: 1
IrDancerPosCtrlGain	LREAL	Reglerverstärkung • Initialwert: 1
IrDancerPosCtrlResetTime	LREAL	Reglernachstellzeit • Einheit: s • Initialwert: 0 (Die Regler-Nachstellzeit ist deaktiviert.)

► [L_TW2P_scPar_WinderDancerCtrlState](#) (26)

9.17 Reduzierte Reglerdynamik bei geringer Regelabweichung

Durch eine reduzierte Reglerdynamik bei geringen Regelabweichungen wird das Dämpfungsverhalten des Regelkreises meist günstig beeinflusst.

Die Regelabweichung ergibt sich aus der Differenz des Soll- und Istwerte, d. h. Differenz aus Eingang *IrSetDancerPosScaled* und Ausgang *IrActDancerPosScaled*.

Mit dem Parameter *IrReducedGain* wird eingestellt, auf welchen Wert die Verstärkung im festgelegten Toleranzbereich reduziert werden soll. Das heißt, im Toleranzbereich wirkt die reduzierte Verstärkung (*IrReducedGain*).

Einzustellende Parameter

Bezeichner	Datentyp	Beschreibung
<i>IrReducedGainWindow</i>	LREAL	Bereich der Regelabweichung mit reduzierter Verstärkung/Empfindlichkeit. Mit <i>IrReducedGainWindow</i> wird der Toleranzbereich der Regelabweichung festgelegt, in dem <i>IrReducedGain</i> wirkt. • Initialwert: 0.0
<i>IrReducedGain</i>	LREAL	Verstärkung der Regelabweichung im Bereich der reduzierten Empfindlichkeit • Initialwert: 0.0

► [L_TW2P_scPar_WinderDancerCtrlState](#) (26)

9.18 Zugkraftsteuerung über Kennlinienfunktion

In Abhängigkeit von der Oberfläche und der Art des Wickelmaterials ist es bei vielen Aufwicklern erforderlich, dass die Zugkraft mit zunehmendem Durchmesser reduziert wird, damit der Wickelballen nicht verschoben wird. Man spricht hierbei von der Wickelcharakteristik oder Zugkraftcharakteristik.

Es ist üblich die Zugkraftbeeinflussung in der Wicklersteuerung vorzunehmen, um den adaptierten Sollwert dann, z. B. auf ein pneumatisches Stellglied, wiederzugeben.

Damit die materialabhängige Charakteristik erreicht wird, wird der eigentliche Zugkraftsollwert aus dem Eingang *IrSetTens* über Kennlinienfunktionen oder durchmesserabhängig bewertet.

Die Adaption kann entsprechend verschiedener Prinzipien erfolgen:

- ▶ [Kennlinie für einen linearen Zugkraftverlauf \(Parameter *dwSelectTensCurve* = 0 \)](#) (📘 91)
- ▶ [Kennlinie für einen linearen Drehmomentverlauf \(Parameter *dwSelectTensCurve* = 1 \)](#) (📘 92)
- ▶ [Freidefinierbare Kennlinie mit 65 Stützstellen \(Parameter *dwSelectTensCurve* = 2 \)](#) (📘 93)

Einzustellende Parameter

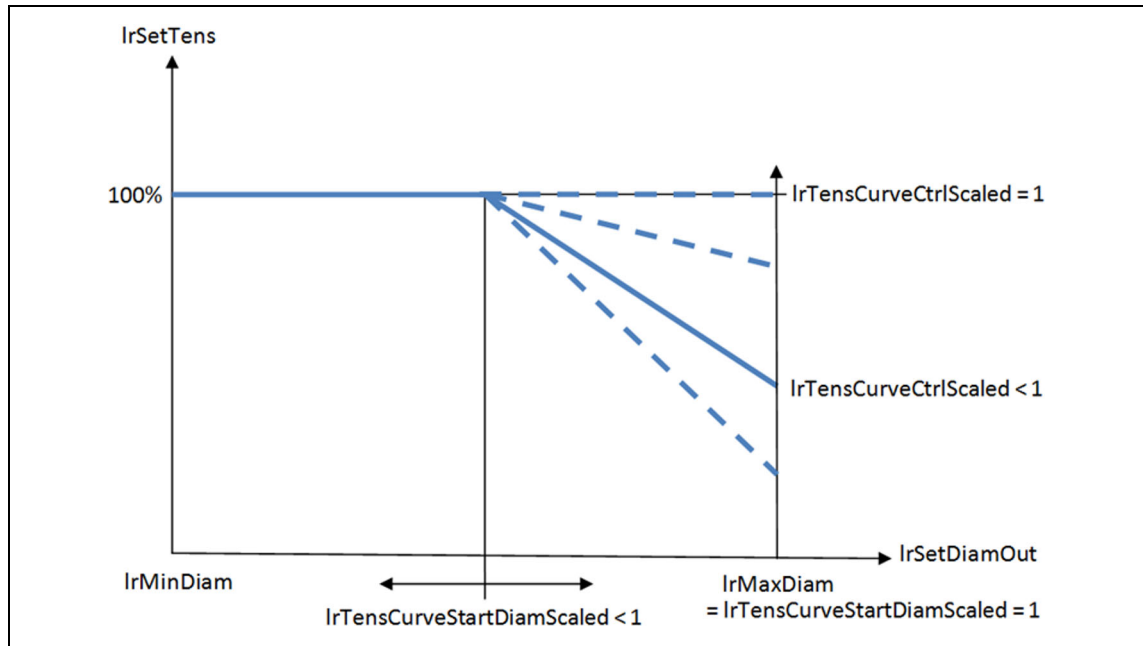
Bezeichner	Datentyp	Beschreibung
<i>dwSelectTensCurve</i>	DWORD	Auswahl der Kennlinie für die Zugkraftsteuerung Initialwert: 0
	0	Linearer Zugkraftverlauf
	1	Linearer Drehmomentverlauf
	2	Zugkraftverlauf nach vorgegebener Kennlinie
<i>alrTensCurve</i>	ARRAY[1..65] OF LREAL	Kennlinie für die Zugkraftsteuerung bestehend aus 65 Werten. Hier müssen alle 65 Werte, gleichmäßig verteilt über den Durchmesserbereich der Parameter <i>IrMinDiam</i> und <i>IrMaxDiam</i> (minimaler und maximaler Durchmesser) angegeben werden.
<i>IrTensCurveStartDiamScaled</i>	LREAL	Anfangspunkt der Kennlinie für die Zugkraftsteuerung - Einheit: x 100 % 1 = 100 % = Parameter <i>IrMaxDiam</i> - Initialwert: 0 (0 %)
<i>IrTensCurveCtrlScaled</i>	LREAL	Steigung der Kennlinie für die Zugkraftsteuerung - Einheit: x 100 % (1 = 100 %) - Initialwert: 0 - Mit dem Wert '1' ergibt sich ein konstanter Zugkraftverlauf und damit ein für den Durchmesser proportional ansteigender Sollwert.
<i>IrMaxDiam</i>	LREAL	Maximaler Durchmesser - Einheit: mm - Initialwert: 180
<i>IrMinDiam</i>	LREAL	Minimale Durchmesser - Einheit: mm - Initialwert: 50

- ▶ [L_TW2P_scPar_WinderDancerCtrlState](#) (📘 26)

9.18.1 Kennlinie für einen linearen Zugkraftverlauf (Parameter $dwSelectTensCurve = 0$)

Die Kennlinie ist gekennzeichnet durch einen Anfangsbereich mit konstanter Bewertung (100 %) und einem zweiten Bereich, in dem die Zugkraft an den Durchmesser angepasst wird.

Mit dem Parameter *IrTensCurveStartDiamScaled* wird festgelegt, bei welchem Durchmesser die Zugkraftabsenkung beginnen soll. Mit dem Parameter *IrTensCurveCtrlScaled* wird die Zugkraft beim maximalen Durchmesser bewertet.

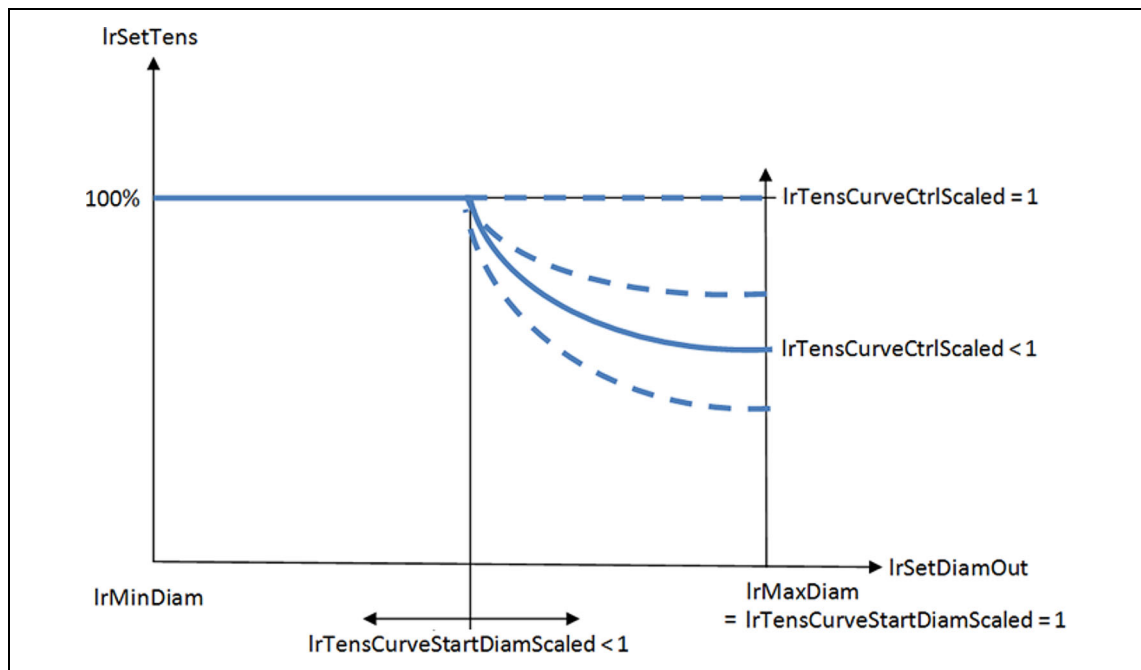


[9-4] Linearer Zugkraftverlauf

9.18.2 Kennlinie für einen linearen Drehmomentverlauf (Parameter $dwSelectTensCurve = 1$)

Die Kennlinie ist gekennzeichnet durch einen Anfangsbereich mit konstanter Bewertung (100 %) und einem zweiten Bereich, in dem die Zugkraft linear über den Drehmomentverlauf angepasst wird. Folglich wird die Kennlinie über nichtlineare Funktionen über den Durchmesser angepasst.

Über den Parameter $IrTensCurveStartDiamScaled$ wird festgelegt, bei welchem Durchmesser die Zugkraftabsenkung beginnt. Mit dem Parameter $IrTensCurveCtrlScaled$ wird der prozentuale Anteil der Zugkraft beim maximalen Durchmesser festgelegt.

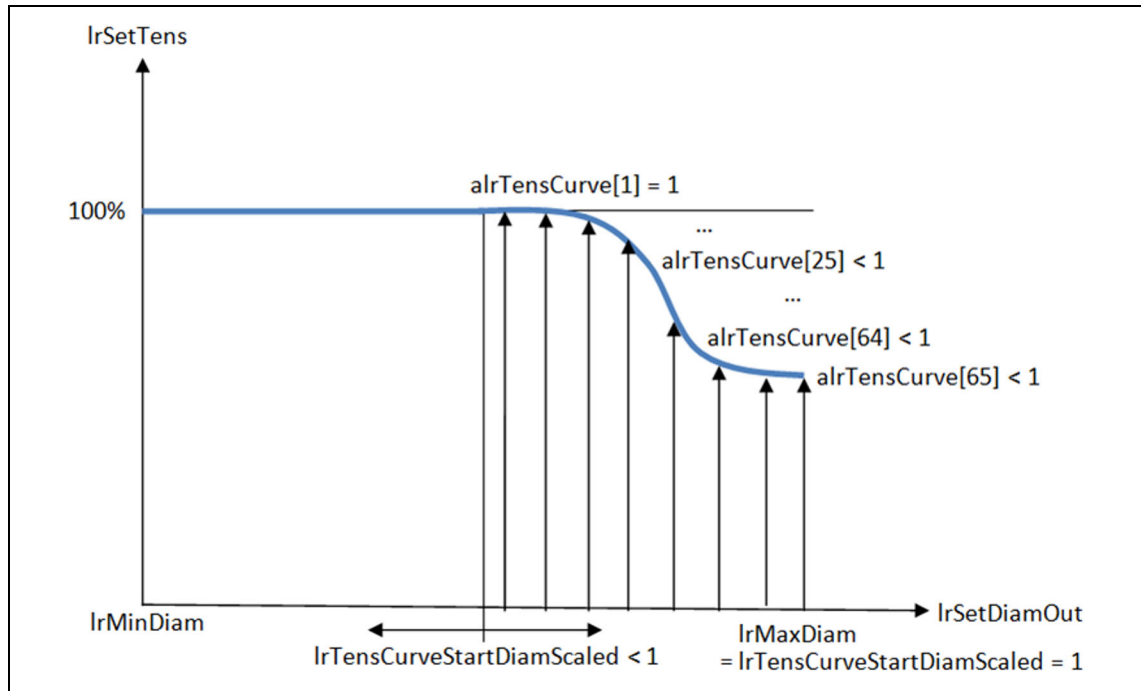


[9-5] Linearer Drehmomentverlauf

9.18.3 Freidefinierbare Kennlinie mit 65 Stützstellen (Parameter `dwSelectTensCurve = 2`)

Die Kennlinie ist parametrierbar über das Array `alrTensCurve`. Es umfasst 65 Werte, gleichmäßig verteilt über den Durchmesserbereich der Parameter `IrMinDiam` und `IrMaxDiam` (minimaler und maximaler Durchmesser).

Für die Kennlinie müssen alle 65 Werte angegeben werden.



[9-6] Freidefinierbare Stützstellen

9.19 Bahnrißüberwachung

Das Technologiemodul bietet zwei Möglichkeiten einer Bahnrißüberwachung.

Über den Parameter *wWebBreakMode* können drei Möglichkeiten einer Bahnrißüberwachung ausgewählt werden:

- Bahnrißüberwachung über die Auswertung der Tänzerposition
- Bahnrißüberwachung über die Änderung des Durchmessers
- Bahnrißüberwachung über Auswertung von Tänzerposition und Durchmesseränderung

Bahnrißüberwachung über die Auswertung der Tänzerposition

Alternativ zur Bahnrißüberwachung auf Basis der Durchmesseränderungsrichtung kann beim tänzerlagegeregelten Wickler auch auf einen Bahnriß geschlossen werden, wenn der Tänzer im Betrieb die untere Endlage erreicht. Die Überwachung wird mit *xWebBreakMonit* = TRUE und dem Parameter *wWebBreakMode* = 1 aktiviert.

Reaktionen auf einen Bahnriß:

- Halten des aktuellen Durchmessers.
- Setzen des Ausgangs *xWebBreak* = TRUE.

Bahnrißüberwachung über die Änderung des Durchmessers

Bei einem Bahnriß entwickelt sich der berechnete Durchmesser entgegen der Wickelrichtung (Abwickeln oder Aufwickeln), da die tatsächliche Umfangsgeschwindigkeit nicht mehr dem Liniengeschwindigkeitssignal entspricht.

Die Überwachung wird mit dem Eingang *xWebBreakMonit* = TRUE aktiviert. Damit ist eine Durchmesseränderung entgegen der Wickelrichtung nur noch innerhalb des im Parameter *IrWebBreakWindow* eingestellten Fensters zulässig.

Der Auf- oder Abwickelbetrieb wird automatisch anhand des Vorzeichens der Liniengeschwindigkeit und der über den Eingang *xWindingDirection* eingestellten Wickelrichtung erkannt.

Die Bahnrißüberwachung darf erst aktiviert werden, wenn der berechnete Durchmesser dem realen Durchmesser entspricht!

Bei aktiver Bahnrißüberwachung (*xWebBreakMonit* = TRUE) wird eine Durchmesseränderung entgegen der über den Ausgang *xUnwind* vorgegebenen Wickelrichtung unterbunden.

Nach dem Laden eines Startdurchmessers, der entgegen der Wickelrichtung deutlich vom realen Durchmesser abweicht, kann dies zum ungewollten Ansprechen der Überwachung führen. So wird beispielsweise beim Aufwickler ein Startdurchmesser von 50 % geladen; der reale Durchmesser beträgt aber nur 45 %. Die Änderung des Durchmesserwertes auf die realen 45 % wird bei aktiver Bahnrißüberwachung verhindert.

Einzustellende Parameter

Bezeichner	Datentyp	Beschreibung
wWebBreakMode	WORD	Modus für die Bahnrißüberwachung • Initialwert: 1
	0	Bahnrißüberwachung aus der Durchmesserberechnung und der Tänzerlage
	1	Bahnrißüberwachung nur aus der Tänzerlage
	2	Bahnrißüberwachung nur aus der Durchmesserberechnung
IrWebBreakWindow	LREAL	Bahnrißfenster Der aktuelle Durchmesser wird mit dem Durchmesser aus dem letzten Berechnungszyklus über das Bahnrißfenster verglichen. • Einheit: x 100 % (1.0 = 100 %) • Initialwert: 0.1 (10 %)

► [L_TW2P_scPar_WinderDancerCtrlState](#) (26)

9.20 Persistente Variablen

Das Technologiemodul bietet die Möglichkeit, die ermittelten Parameter, wie z. B. den Wickeldurchmesser oder die "erlernten" Tänzerendlagen, persistent zu speichern. Dazu müssen im »PLC Designer« folgende Einstellungen für das Technologiemodul ausgeführt werden.

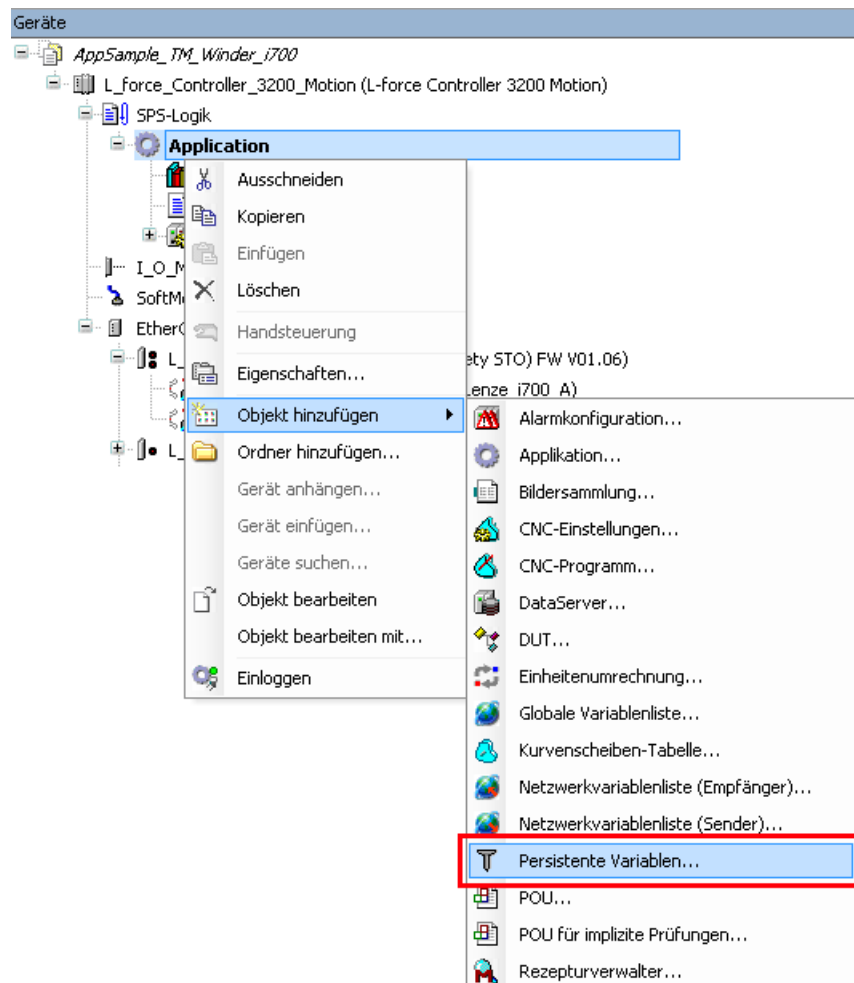


So legen Sie im »PLC Designer« persistente Variablen an:

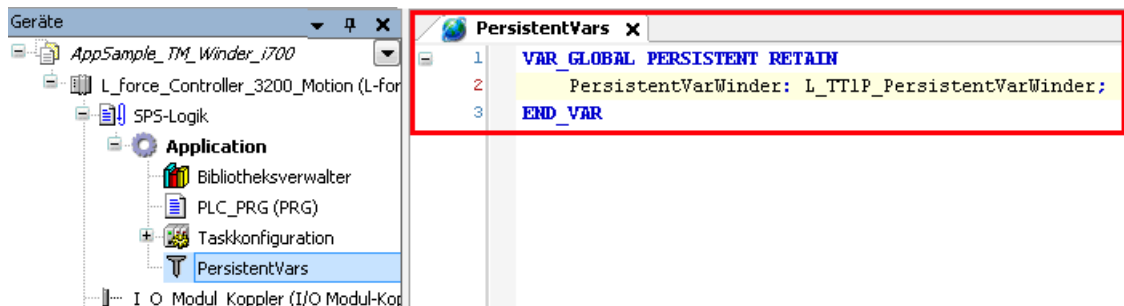
Hinweis!

Diese Vorgehensweise gilt nicht für das ApplicationTemplate, weil dort bereits Strukturen für persistente Daten der Maschinenmodule bereitgestellt werden.

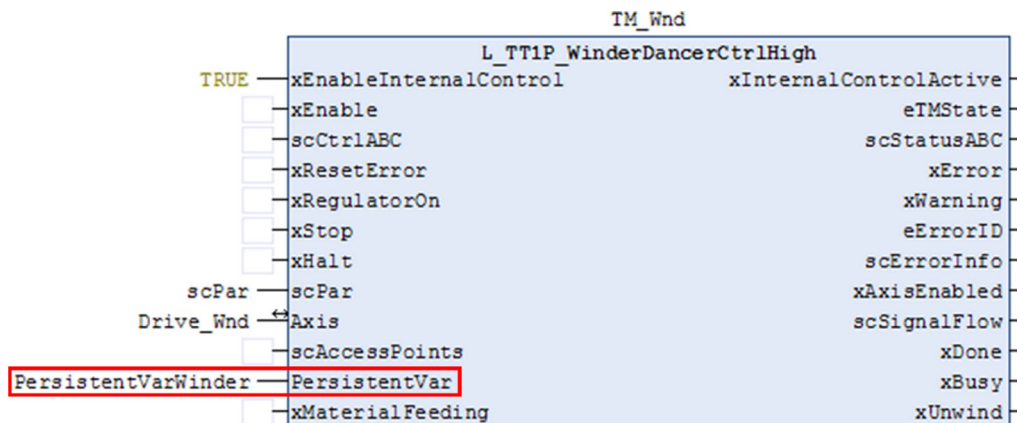
1. Im Kontextmenü zu **Application** mit dem Befehl **Objekt hinzufügen → Persistente Variablen...** die globale Liste für die Verwaltung von persistenten Variablen hinzufügen.



2. Die Referenz der persistenten Variablen "L_TT1P_PersistentVarWinder" in der globalen Struktur der persistenten Variablen instanziiieren.



3. Die Instanz der persistenten Variablen mit dem Eingang *PersistentVar* verschalten.



L_TW2P_PersistentVarWinder

Inhalt der persistenten Variablen innerhalb der Struktur **L_TW2P_PersistentVarWinder**:

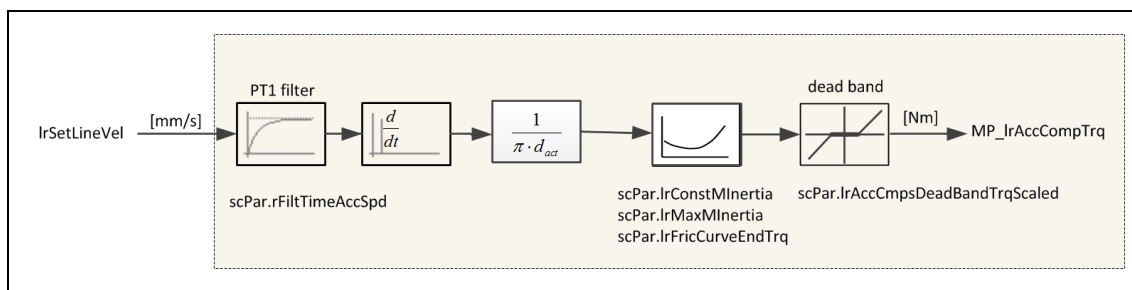
Bezeichner	Datentyp	Beschreibung
IrDiam	LREAL	Der berechnete Durchmesser • Einheit: mm
IrMaterialCounter	LREAL	Zählerstand des Materials • Einheit: mm
liRevCounter	LINT	Anzahl der gezählten Umdrehungen für den Materiallängenzähler • Einheit: rev
IrRevCounterResidual	LREAL	Der Bruchteil einer nicht vollständigen Umdrehung für den Materiallängenzähler • Einheit: rev
xUpperPosTeached	BOOL	Referenzierte obere Tänzerendlage TRUE Die obere Tänzerendlage bereits referenziert.
xLowerPosTeached	BOOL	Referenzierte obere Tänzerendlage TRUE Die untere Tänzerendlage bereits referenziert.
IrDancerUpperLimit	LREAL	Der analoge Sensorwert aus dem Eingang <i>IrActDancerPosIn</i> für die obere Tänzerendlage.
IrDancerLowerLimit	LREAL	Der analoge Sensorwert aus dem Eingang <i>IrActDancerPosIn</i> für die untere Tänzerendlage.

9.21 Beschleunigungskompensation

Die Beschleunigung im Liniengeschwindigkeitssollwert stellt im Wickelprozess eine Störgröße dar. Das Drehmoment, das zur Beschleunigung aufgebracht werden muss, fehlt in der Zugkraft.

Das Beschleunigungsdrehmoment muss also berechnet und als Zusatzdrehmoment vorgesteuert werden.

Berechnung des Beschleunigungsdrehmoments		
$M = 2 \cdot \pi \cdot \left(\frac{\partial n}{\partial t} \right) \cdot (J_{const} + J_{var})$ mit $J_{var} = (J_{max} - J_{const}) \cdot \left(\frac{d_{act}^4 - d_{min}^4}{d_{max}^4 - d_{min}^4} \right) \cdot B$		
Formelzeichen	Beschreibung	Maßeinheit
M	Beschleunigungsdrehmoment	Nm
∂n	(Delta-)Drehzahl des Motors	rev/s
∂t	(Delta-)Zeit	s
J_{const}	Konstantes Massenträgheitsmoment	kgm ²
J_{var}	Variables (durchmesserabhängiges) Massenträgheitsmoment	
J_{max}	Maximales Massenträgheitsmoment	
d_{act}	Aktueller Durchmesser	mm
d_{min}	Minimaler Durchmesser (Hülsendurchmesser)	
d_{max}	Maximaler Durchmesser	
B	Materialbreite	mm



Zur Bildung einer Beschleunigung erfolgt eine Differenzierung des Liniengeschwindigkeitssignals.

In der Praxis ist mit einem nicht ideal, stetig verlaufenden Liniengeschwindigkeitssignal zu rechnen. Je nach Auflösung und Stabilität dieses Signals kann eine Glättung über das PT1-Filter erforderlich sein. Die Filterzeit wird über den Parameter *rFiltTimeAccSpd*.

Die Totgang-Funktion wird zur Rauschunterdrückung in dem berechneten Kompensationsdrehmoment angewendet. Damit wird vermieden, dass kleine Beschleunigungsänderungen, die nur aus der Quantisierung oder dem Rauschen des Liniengeschwindigkeitssignals resultieren, zu einem Kompensationsdrehmoment führen. Der Parameter *IrAccCmpsDeadBandTrqScaled* bestimmt, bis zu welchem Wert der Totgang wirkt.

Die Beschleunigungskompensation wird mit dem Eingang *xAccCmp* = TRUE freigegeben.

Unterschiedliche Materialbreiten oder Materialdichten können prozentual über den Eingang *IrMinertiaAdapt* berücksichtigt werden.

Die Parameter *IrAccCmpsGainAcc* und *IrAccCmpsGainDec* bewerten das berechnete Kompensationsdrehmoment in Abhängigkeit der Wirkrichtung. Zur Vermeidung von Materiallose soll beim Beschleunigen stärker kompensiert werden als beim Verzögern.

Unterschiedliche Materialbreiten oder Materialdichten können prozentual über den Eingang *IrMInertiaAdapt* berücksichtigt werden.

Einzustellende Parameter

Bezeichner	Datentyp	Beschreibung
rFiltTimeAccSpd	REAL	PT1-Filterzeit für die Drehzahl der Wickelwelle zur Beschleunigungskompensation • Einheit: s • Initialwert: 0.005
IrAccCmpsDeadBandTrqScaled	LREAL	Nacheilbereich (Dead Band) für das aktuelle Beschleunigungsmoment • Einheit: x 100 % (1.00 = 100 %); - Bezugsgröße : Motor-Bemessungsdrehmoment [Nm] • Initialwert: 0.1 (10 %)
IrAccCmpsGainAcc	LREAL	Verstärkungsfaktor für das Beschleunigungsmoment in positive Richtung • Einheit: x 100 % (1.00 = 100 %) • Wertebereich: 0 ... 2 (0 ... 200 %) • Initialwert: 1.05 (105 %)
IrAccCmpsGainDec	LREAL	Verstärkungsfaktor für das Beschleunigungsmoment in negative Richtung • Einheit: x 100 % (1.00 = 100 %) • Wertebereich: 0 ... 2 (0 ... 200 %) • Initialwert: 0.95 (95 %)
IrConstMInertia	LREAL	Konstantes Massenträgheitsmoment an der Wickelwelle • Einheit: kgcm² • Initialwert: 1
IrMaxMInertia	LREAL	Maximal zulässiges Massenträgheitsmoment an der Wickelwelle • Einheit: kgcm² • Initialwert: 10

► [L_TW2P_scPar_WinderDancerCtrlState](#) (26)

9.22 Identifikation der Massenträgheitsmomente

Ein passendes Massenträgheitsmoment wird benötigt, wenn die Beschleunigungskompensation oder die Adaption der Drehzahlreglerverstärkung genutzt werden soll.

Dazu muss die Konstante in Parameter *IrConstMinertia* und das maximale Massenträgheitsmoment in Parameter *MaxMinertia* passend parametrieren sein. Im Betrieb wird dann das aktuelle Massenträgheitsmoment in Abhängigkeit des aktuellen Durchmessers automatisch berechnet und an *IrSetMinertiaOut* ausgegeben.

Die Ermittlung des konstanten und maximalen Massenträgheitsmomentes kann per manueller Berechnung oder per Identifikation erfolgen.

Das maximale Massenträgheitsmoment entspricht dem Wickelballen mit dem maximalen Massenträgheitsmoment, für den der Wickler konzipiert ist. Zur Adaption des tatsächlich genutzten Wickelballens, der sich in der Materialdichte, dem maximalen Durchmesser und/oder der Materialbreite unterscheiden kann, stehen jeweils Bewertungsfaktoren bei der Beschleunigungskompensation und der Drehzahlreglerverstärkung zur Verfügung.

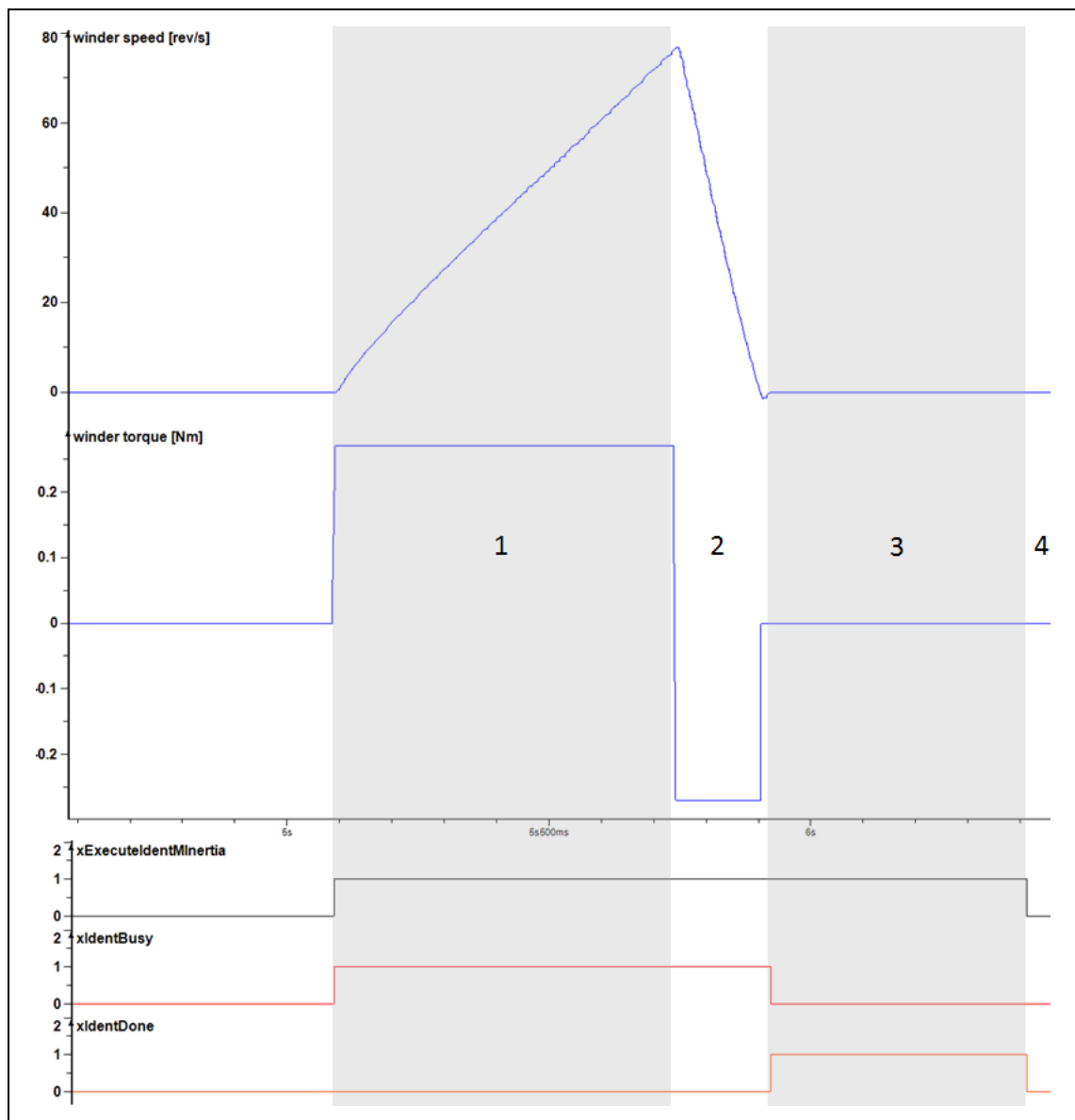
Identifikation des konstanten Massenträgheitsmoments

Die Wickelwelle ist leer (ohne Material).

Mit dem Parameter *IrIdentMinertiaMaxSpdScaled* wird die maximale Motordrehzahl in [max. 100 %] bezogen auf die maximal erreichbare Wicklerdrehzahl *IrWndSpdRef* festgelegt. Typischerweise sind hier Drehzahlen zwischen 50 ... 60 % ausreichend.

Mit dem Parameter *IrIdentMinertiaMaxTrqScaled* wird das Beschleunigungsmoment festgelegt. Dieser Wert muss immer größer sein, als die maximal auftretende Reibung – Empfehlung: 25 %.

Mit einer steigenden Flanke (FALSE → TRUE) am Eingang *xExecutIdentMinertia* erfolgt die Ermittlung des Massenträgheitsmoments. Am Ausgang *IrIdentMinertia* wird das ermittelte Massenträgheitsmoment angezeigt.



Nr	xExecutIdentMinertia	xIdentBusy	xIdentDone	eTMState	INFO
1	FALSE → TRUE	FALSE → TRUE	FALSE	STANDBY → IDENTMINERTIA	Die Achse wird in Drehmomentmodus "CST" geführt. Mit dem positiven Drehmoment, das über die Parametrierung für die Identifizierung festgelegt ist, wird die Achse bis zur definierten identifizierten Geschwindigkeit beschleunigt.
2	TRUE	TRUE	FALSE	IDENTMINERTIA	Die Geschwindigkeit, die über die Parametrierung für die Identifizierung bestimmt ist, ist erreicht. Das Drehmoment wird in negativer Richtung beauftragt. Die Achse wird nun abgebremst.
3	TRUE	TRUE → FALSE	FALSE → TRUE	IDENTMINERTIA → STANDBY	Die Achse erreicht den Stillstand und das Drehmoment wird abgeschaltet. Die erfassten Verläufe werden ausgewertet und die Massenträgheit wird bestimmt.
4	TRUE → FALSE	FALSE	TRUE → FALSE	STANDBY	Bei Rücksetzen der Beauftragung zur Identifizierung wird der Status Done ebenfalls zurückgesetzt.

Die Regelstabilität des Antriebs sowie die Istwertauflösung von Drehzahl und Drehmoment gehen direkt in das Berechnungsergebnis der Identifikation ein. Eine Filterung der aktuellen Werte verbessert häufig das Ergebnis der Identifikation:

- Filterzeit für das aktuelle Drehmoment: $rFiltTimIdentMinertiaTrq$
- Filterzeit für die aktuelle Wicklerdrehzahl: $rFiltTimIdentMinertiaSp$

Identifikation des maximalen Massenträgheitsmoments

Der Wickler ist mit dem maximal möglichen Wickelballen beladen (maximaler Durchmesser und maximale Breite).

Die maximale Motordrehzahl *IrIdentMInertiaMaxSpdScaled* muss so parametrierung werden, dass die maximal zulässige Umfangsgeschwindigkeit des Wicklers nicht überschritten wird (z. B. *IrIdentMInertiaMaxSpdScaled* = 10 % bei $d_{\max}/d_{\min} = 10$) – Empfehlung: 25 %.

Mit einer steigenden Flanke (FALSE $\rightarrow$ TRUE) am Eingang *xExecutIdentMInertia* erfolgt die Ermittlung des Massenträgheitsmoments.

Beendigung der Identifikation

Die Identifikation ist beendet, wenn der Motor wieder den Stillstand erreicht hat, keine Fehler gemeldet wurden und *xDone* auf TRUE gesetzt wurde.

Die ermittelte Trägheit der Wickelwelle (J_{Winder}) wird am Ausgang *IrIdentMInertia* angezeigt und muss auf Plausibilität geprüft werden.

Die Identifikation sollte für beide Fälle mehrmals durchgeführt werden. Dabei kann die Filterzeit für die Drehzahl *rFiltTimIdentMInertiaSpd* variiert werden.

Eine ausgeprägte nichtlineare Reibung im System beeinflusst die Berechnung des Massenträgheitsmoments im Technologiemodul negativ.

Einzustellende Parameter

Bezeichner	Datentyp	Beschreibung
<i>rFiltTimIdentMInertiaSpd</i>	REAL	PT1-Filterzeit für die Drehzahl der Wickelwelle während der Identifikation des Massenträgheitsmoments - Einheit: s - Initialwert: 0.01
<i>rFiltTimIdentMInertiaTrq</i>	REAL	PT1-Filterzeit für das Drehmoment der Wickelwelle während der Identifikation des Massenträgheitsmoments - Einheit: s - Initialwert: 0.005
<i>IrIdentMInertiaMaxSpd Scaled</i>	LREAL	Maximale Drehzahl der Wickelwelle während der Massenträgheitsidentifikation - Einheit: x 100 % (1.0 = 100 % = <i>IrWndSpdRef</i>) - Wertebereich: 0 ... 1 (0 ... 100 %) - Initialwert: 0.2 (20 %)
<i>IrIdentMInertiaMaxTrq Scaled</i>	LREAL	Maximales Drehmoment der Wickelwelle während der Massenträgheitsidentifikation - Einheit: x 100 % (1.0 = 100 %) - Bezugsgröße: Motor-Bemessungsdrehmoment - Wertebereich: 0 ... 1 (0 ... 100 %) - Initialwert: 0.2 (20 %)

► [L_TW2P_scPar_WinderDancerCtrlState](#) (26)

Berechnung des maximalen Massenträgheitsmoments

Steht kein Wickelballen zur Verfügung, so kann das maximale Massenträgheitsmoment wie folgt berechnet werden:

Berechnung des maximalen Massenträgheitsmoments		
Die <u>Dichte</u> des Wickelmaterials ist bekannt:		
$J_{\text{MaxWinder}} = i^2 \cdot J_{\text{Motor}} + \left(\frac{\pi}{32 \cdot 10^8} \right) \cdot B \cdot \rho \cdot (d_{\text{max}}^4 - d_{\text{min}}^4)$		
Die <u>Masse</u> des Wickelmaterials ist bekannt:		
$J_{\text{MaxWinder}} = i^2 \cdot J_{\text{Motor}} + \frac{m \cdot d_{\text{max}}^4}{800}$		
Formelzeichen	Beschreibung	Maßeinheit
$J_{\text{MaxWinder}}$	Maximales Massenträgheitsmoment der Wickelwelle	kgcm ²
J_{Motor}	Massenträgheitsmoment der Motorwelle	kgcm ²
i	Getriebefaktor	
B	Materialbreite	mm
ρ	Materialdichte	kg / dm ³
d_{max}	Maximaler Durchmesser	mm
d_{min}	Minimaler Durchmesser (Hülsendurchmesser)	mm
m	Masse	kg

Das konstante Massenträgheitsmoment wird im Wesentlichen durch den Motor bestimmt. Dessen Massenträgheitsmoment ist im Katalog angegeben.

Die berechneten oder auch identifizierten Werte müssen in den jeweiligen Parameter übertragen werden:

- *IrConstMinertia*
- *IrMaxMinertia*

Um die Werte netzausfallsicher im Gerät zu hinterlegen, muss der Parametersatz gespeichert werden!

9.23 Adaption der Drehzahlreglerverstärkung

Voraussetzung

- Die Wicklerachse muss freigegeben sein (Eingang *xRegulatorOn* = TRUE).

Funktionsweise

Betrachtet man Motor und Wickelballen als ein starres Ein-Masse-System, so verhält sich die optimale Verstärkung des Drehzahlreglers direkt proportional zum Massenträgheitsmoment.

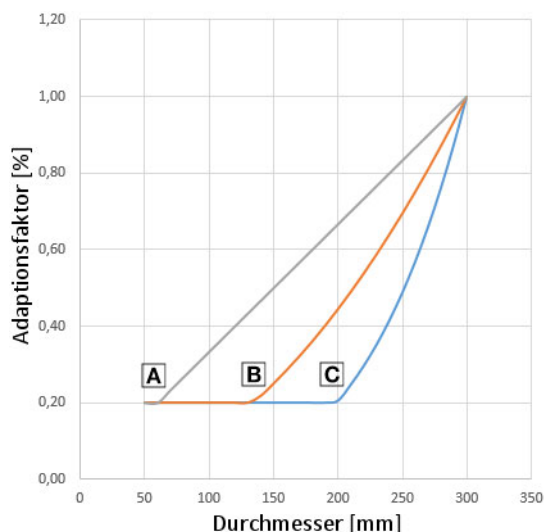
Da sich während des Wickelprozesses das Massenträgheitsmoment meist deutlich verändert, kann es für ein stabiles Regelverhalten erforderlich sein, die Verstärkung des Drehzahlreglers mit dem Massenträgheitsmoment mitzuführen.

- Entsprechend der gewählten Methode in Parameter *eAdaptSpdCtrlGainMode* wird im Technologiemodul ein Adaptionsfaktor gebildet, entweder in direkter Abhängigkeit zum aktuellen Durchmesser oder zum Massenträgheitsmoment.
- Die für den Drehzahlregler eingestellte Verstärkung wird mit dem Faktor in Parameter *IrAdaptSpdCtrlGainFactor* (Parameter *IrAdaptSpdCtrlLowLimit* 0 ... 1) multipliziert. Somit wirkt die eingestellte Verstärkung erst bei maximalem Durchmesser oder maximalem Massenträgheitsmoment.

Folgende Methoden zur Bildung des Adaptionsfaktors sind im Parameter *eAdaptSpdCtrlGainMode* auswählbar:

- 0: *DiamToSquare*: Der Adaptionsfaktor entspricht d_{act}^2 / d_{max}^2
- 1: *Diam*: Der Adaptionsfaktor entspricht d_{act} / d_{max}
- 2: *Inertia*: Der Adaptionsfaktor entspricht dem J_{act} / J_{max}

Beispiel: Adaption der Drehzahlreglerverstärkung mit unterer Begrenzung auf *IrAdaptSpdCtrlLowLimit* = 0.2



A *eAdaptSpdCtrlGainMode* = 0: *DiamToSquare*

B *eAdaptSpdCtrlGainMode* = 1: *Diam*

C *eAdaptSpdCtrlGainMode* = 2: *Inertia*

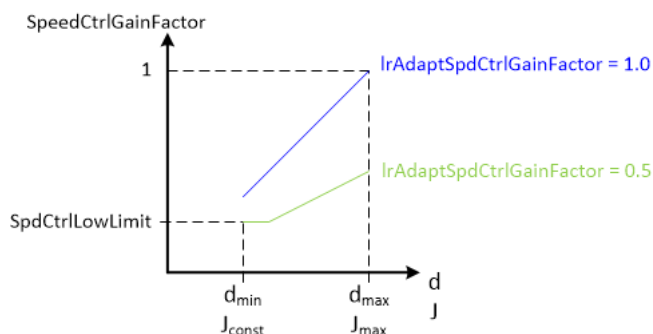
Bei sehr steifen Mechaniken und Wickelmaterialien und minimierten Totzeiten im Regelkreis kann die Adaptionmethode in Parameter *eAdaptSpdCtrlGainMode* = 2 (*Inertia*) angewendet werden.

In diesem Fall müssen die Massenträgheitsmomente in den Parametern *IrConstMIInertia* und *IrMaxMIInertia* plausibel eingetragen werden.

In vielen Anwendungen ist es ausreichend die Standard-Adaptionmethode in Parameter *eAdaptSpdCtrlGainMode* = 0 (*DiamToSquare*) zu verwenden. In diesem Fall hat das Massenträgheitsmoment keinen Einfluss auf die Adaption.

Adaption der Drehzahlregler-Verstärkung aktivieren / deaktivieren

Die Adaption der Drehzahlregler-Verstärkung ist in allen Technologiemodul-Zuständen aktiv, wenn der Parameter *xAdaptSpdCtrlGain* = TRUE gesetzt ist.



Einzustellende Parameter

Bezeichner	Datentyp	Beschreibung
<i>eAdaptSpdCtrlGainMode</i>	ENUM	Modusauswahl zur Adaption der Drehzahlreglerverstärkung. • Initialwert: 2
		0 DiamToSquare; $VP = f(d^2)$
		1 Diam; $VP = f(d)$
		2 Inertia; $VP = f(J)$
<i>IrAdaptSpdCtrlLowLimit</i>	LREAL	Untere Begrenzung der Drehzahlreglerverstärkung im Antrieb. Der Adaptionswert <i>IrSetSpdCtrlGainAdaptOut</i> darf nicht kleiner sein als der Wert in Parameter <i>IrAdaptSpdCtrlLowLimit</i> . • Wertebereich: 0 bis 1 • Initialwert: 0
<i>alrSpdCtrlGainAdaptX</i>	ARRAY [1...9] OF LREAL	Kennlinienfunktion für die Drehzahlregelungsverstärkung Die X-Achse entspricht dem dem aktuell berechneten Apationsfaktor für die Anpassung der Drehzahlreglerverstärkung. • Einheit: x 100 % (1 = 100 % = Parameter <i>IrMaxMIInertia</i>) • Initialwerte: [0.0, 0.2, 0.4, 0.6, 0.8, 1.0, 1.2, 1.4, 1.6]
<i>alrSpdCtrlGainAdaptY</i>	ARRAY [1...9] OF LREAL	Kennlinienfunktion für die Drehzahlregelungsverstärkung Die Y-Achse entspricht dem Verstärkungsfaktor des Drehzahlreglers. • Einheit: x 100 % (1 = 100 %) • Initialwerte: [0.0, 0.2, 0.4, 0.6, 0.8, 1.0, 1.2, 1.4, 1.6]
<i>xAdaptSpdCtrlGain</i>	BOOL	Adaption der Drehzahlreglerverstärkung ein-/ausschalten. • Initialwert: FALSE
		TRUE Adaption der Drehzahlreglerverstärkung freigegeben.
		FALSE Adaption der Drehzahlreglerverstärkung inaktiv.

► [L TW2P_scPar_WinderDancerCtrlState](#) (26)

9.24 Begrenzung der Master-Liniengeschwindigkeit

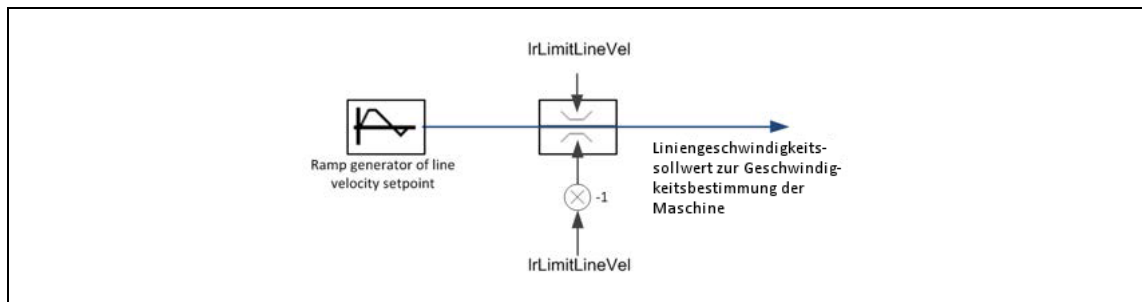
Zur Reduzierung der Antriebsleistung bei kleinen Wicklerdurchmessern oder um zulässige Getriebe-Eintriebsdrehzahlen nicht zu überschreiten, kann es erforderlich sein, die Liniengeschwindigkeit der Anlage zu begrenzen. Die Berechnung der Begrenzung erfolgt im Technologiemodul.

Die maximale Drehzahl der Wickelwelle (abtriebsseitig) wird über den Parameter *IrMaxWndSpd* festgelegt. Mit Eingabe dieses Parameters ist die Begrenzungsfunktion direkt freigegeben.

Am Ausgang *IrLimitLineVel* wird die maximal zugelassen Liniengeschwindigkeit in [mm/s] ausgegeben. Ein Überschreiten der Liniengeschwindigkeit *IrLimitLineVel* bedeutet zwangsläufig auch eine Überschreitung der maximalen Wicklerwellen-Drehzahl in Parameter *IrMaxWndSpd*.

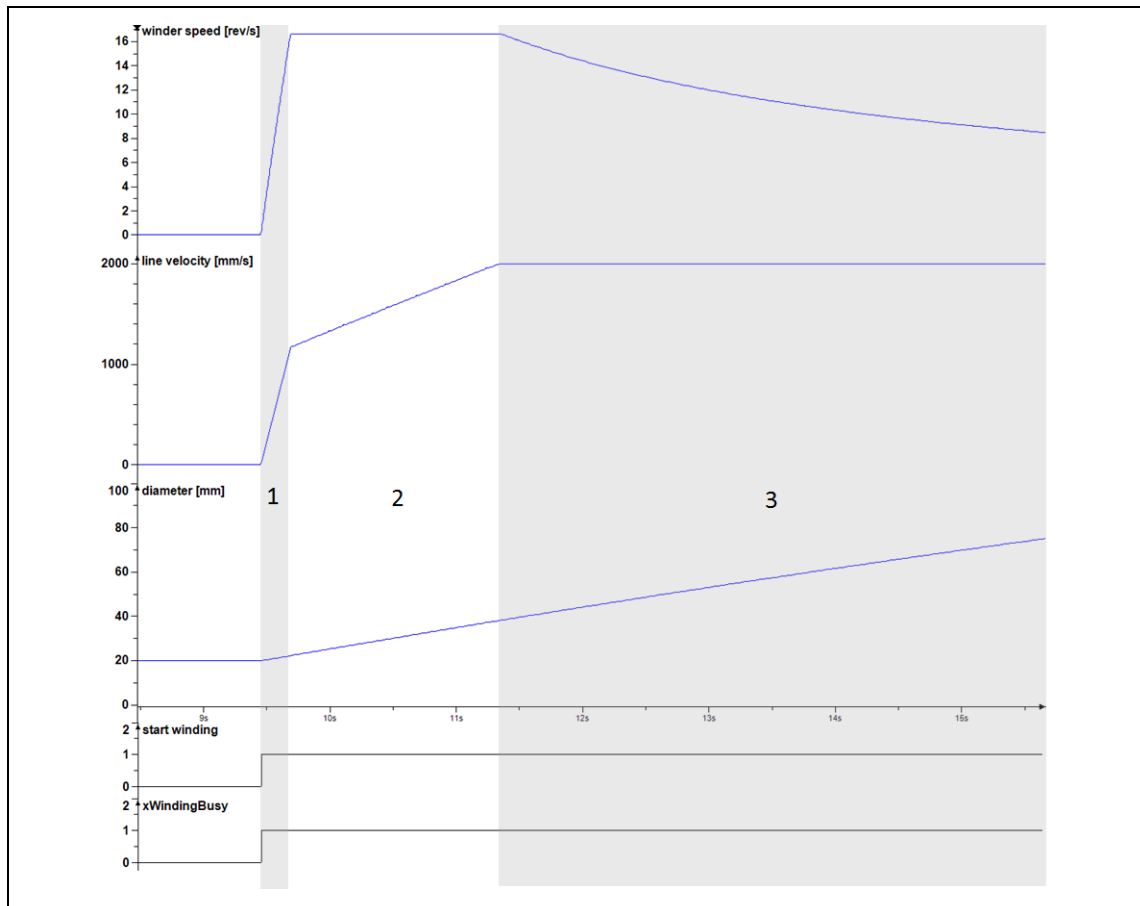
Beispiel

Das Beispiel zeigt die Umsetzung einer Begrenzung des Anlagensollwertes im geschwindigkeitsbestimmenden Antrieb (nicht im Wickler!) unter Verwendung des Ausgangs *IrLimitLineVel* aus dem Technologiemodul.



[9-7] Signalfloss zur Begrenzung der Master-Liniengeschwindigkeit

Folgende Grafik verdeutlicht die notwendige Begrenzung der Liniengeschwindigkeit zwischen Referenz-Liniengeschwindigkeit (V_{ref}) bis maximaler Liniengeschwindigkeit (V_{max}) für einen Aufwickler, der bei minimalem Durchmesser (D_{min}) startet, um die maximal zugelassene Drehzahl der Wickelwelle in Parameter *IrMaxWndSpd* nicht zu überschreiten.



[9-8] Beispiel für die Begrenzung der Liniengeschwindigkeit

Nr	xWindingBusy	Begrenzung ist aktiv	Liniengeschwindigkeit hat das Maximum erreicht	INFO
1	TRUE	FALSE	FALSE	Das TM befindet sich im Wickelbetrieb. Die Liniengeschwindigkeit wird größer.
2	TRUE	TRUE	FALSE	Der Wickelantrieb erreicht die maximal erlaubte Drehzahl. Die Linienbegrenzung aus dem Wickler begrenzt die Liniengeschwindigkeit.
3	TRUE	FALSE	TRUE	Die Linie hat die maximale Geschwindigkeit erreicht und wird konstant verfahren. Der Durchmesser wächst weiterhin. Die Drehzahl des Wicklers wird geringer.

Einzustellende Parameter

Bezeichner	Datentyp	Beschreibung
IrMaxWndSpd	LREAL	Festlegung der maximalen Wickelwellen-Drehzahl (abtriebsseitig). • Einheit: rev/s • Initialwert: 100

► [L_TW2P_scPar_WinderDancerCtrlState](#) (26)

9.25 Beendigung des Wickelprozesses

Zur Beendigung des Wickelprozesses (Zustand "DANCERCTRL") gibt es zwei Möglichkeiten:

A. Eingänge $xDancerCtrl$ = FALSE und $xSyncln$ = TRUE setzen.

Das Technologiemodul wechselt vom Wickelprozess in die synchrone Fahrt. Hierbei wird die Wicklerumfangsgeschwindigkeit auf die Liniengeschwindigkeit (Eingang $IrSetLineVel$) synchronisiert.

Zustandswechsel: "DANCERCTRL" ==> "SYNCLINEVEL"

B. Eingänge $xDancerCtrl$ = FALSE und $xSyncln$ = FALSE setzen.

Der Wickelprozess wird in Abhängigkeit der Einstellung des Parameters $eDancerCtrlStopMode$ beendet.

- $eDancerCtrlStopMode$ = 0: Halt
Die Achse wird über die Verzögerung ($IrHaltDec$) und den Ruck ($IrJerk$) in den Stillstand geführt.
- $eDancerCtrlStopMode$ = 1: Move Abs
Die Achse wird mit der Geschwindigkeit ($IrVel$), Beschleunigung ($IrAcc$), Verzögerung ($IrDec$) und den Ruck ($IrJerk$) in die Zielposition ($IrPos_Dist$) gefahren.
Die Zielposition wird über die Struktur $scCtrlBasicMotion.IrPos_Dist$ am Eingang des Technologiemoduls vorgeben.
- $eDancerCtrlStopMode$ = 2: Move Rel
Die Achse wird mit der Geschwindigkeit ($IrVel$), Beschleunigung ($IrAcc$), Verzögerung ($IrDec$) und den Ruck ($IrJerk$) nach der gefahrenen Wegstrecke ($IrPos_Dist$) in den Stillstand geführt.
Der Weg wird über die Struktur $scCtrlBasicMotion.IrPos_Dist$ am Eingang des Technologiemoduls vorgeben.

Zustandswechsel: "DANCERCTRL" ==> "STOP"

Einzustellende Parameter

Bezeichner	Datentyp	Beschreibung
$eDancerCtrlStopMode$ $L_TW2P_DancerCtrlStopMode$		Modus, in dem der Wickelprozess (Zustand "DANCERCTRL") beendet wird. • Initialwert: 0
	0	Halt Die Achse wird über die Verzögerung ($IrHaltDec$) und den Ruck ($IrJerk$) angehalten.
	1	Move Abs Absolute Fahrt/Positionierung: Die Achse wird mit der Geschwindigkeit ($IrVel$), Beschleunigung ($IrAcc$), Verzögerung ($IrDec$) und den Ruck ($IrJerk$) in die Zielposition ($IrPos_Dist$) gefahren.
	2	Move Rel Relative Fahrt/Positionierung: Die Achse wird mit der Geschwindigkeit ($IrVel$), Beschleunigung ($IrAcc$), Verzögerung ($IrDec$) und den Ruck ($IrJerk$) nach der gefahrenen Wegstrecke ($IrPos_Dist$) in den Stillstand geführt.
$IrVel$	LREAL	Geschwindigkeit für Bewegungen, die über die Struktur scCtrlBasicMotion beauftragt werden. • Einheit: units/s • Initialwert: 3600
$IrAcc$	LREAL	Beschleunigung für Bewegungen, die über die Struktur scCtrlBasicMotion beauftragt werden. Vorgabe, mit welcher Geschwindigkeitsänderung maximal beschleunigt werden soll. • Einheit: units/s ² • Initialwert: 36000

Bezeichner	Datentyp	Beschreibung
IrDec	LREAL	Verzögerung für Bewegungen, die über die Struktur scCtrlBasicMotion beauftragt werden. Vorgabe, mit welcher Geschwindigkeitsänderung maximal bis zum Stillstand verzögert werden soll. • Einheit: units/s^2 • Initialwert: 100
IrJerk	LREAL	Ruck für Bewegungen, die über die Struktur scCtrlBasicMotion beauftragt werden. • Einheit: units/s^3 • Initialwert: 0
IrPos_Dist	LREAL	Absolute Positionierung Absolute Zielposition bezogen auf die Nullposition der Achse. • Einheit: units • Initialwert: 0
		Relative Positionierung Zu verfahrenende Wegstrecke bezogen auf die Sollposition zum Startzeitpunkt des Kommandos. • Einheit: units • Initialwert: 0

► [L_TW2P_scPar_WinderDancerCtrlState](#) (26)

A

Abbruch einer Positionierung [53](#)
Absolute Positionierung [45](#)
Abweichung im Bereich reduzierter Empfindlichkeit [89](#)
Access points [36](#)
Adaption der Drehzahlreglerverstärkung [104](#)
Anlauf der Achsen [15](#)
Anwendungshinweise [8](#)
Ausgänge [23](#)

B

Bahnrißüberwachung [94](#)
Beendigung des Wickelprozesses [108](#)
Beendigung einer Positionierung [53](#)
Begrenzung der Master- Liniengeschwindigkeit [106](#)
Beschleunigungskompensation [98](#)
Betriebsmodus [14](#)

D

Dokumenthistorie [6](#)
Drehzahlreglerverstärkung (Adaption) [104](#)
Drehzahlvorsteuerung [63](#)
Durchmesser halten [66](#)
Durchmesser vorgeben [67](#)
Durchmesserberechnung [64](#)
Durchmesserberechnung mit Korrektur der Tänzerposition [69](#)
Durchmessersensor-Signal [67](#)

E

Eingänge [19](#)
Eingänge und Ausgänge [19](#)
E-Mail an Lenze [112](#)
Endlosfahrt mit definierter Geschwindigkeit [50](#)

F

Feedback an Lenze [112](#)
Funktionen des Technologiemoduls (Übersicht) [13](#)
Funktionsbeschreibung "Winder Dancer-controlled (V2)" [11](#)

G

Geschwindigkeits-Override [56](#)
Gestaltung der Sicherheitshinweise [8](#)

H

Halt [44](#)
Handfahren (Jogging) [38](#)
Hinweise zum Betrieb des Technologiemoduls [14](#)

I

Identifikation der Massenträgheitsmomente [100](#)
Identifikation des Durchmessers [71](#)

K

Korrektur der Tänzerposition [69](#)

L

L_TW2P_scPar_WinderDancerCtrlState [26](#)
L_TW2P_WinderDancerCtrlState [17](#)
Leitwert-Quelle für die Durchmesserberechnung [62](#)

M

Massenträgheitsmomente identifizieren [100](#)
Master-Liniengeschwindigkeit begrenzen [106](#)
Materiallängenzähler [75](#)
Materiallängenzählung (Quellen) [75](#)
Materialzuführung an den Wickler [60](#)
Max. Massenträgheitsmoment berechnen [103](#)

N

Normierung der Tänzerlage [82](#)

O

Override für Geschwindigkeit [56](#)

P

Parameterstruktur L_TW2P_scPar_WinderDancerCtrlState [26](#)
Persistente Variablen [96](#)
PI-Regler für die Tänzerlageregelung [87](#)

Q

QSP Application [43](#)
Quellen für die Materiallängenzählung [75](#)

R

Referenzfahrt (Homing) [40](#)
Regelabweichung im Bereich reduzierter Empfindlichkeit [89](#)
Relative Positionierung [47](#)

S

Sicherheitshinweise [9](#)
Signal vom Durchmessersensor [67](#)
Signalfluss des Technologiemoduls "Winder Dancer-controlled" [33](#)
Signalflussplan [33](#)
Startdurchmesser vorgeben [67](#)
State Machine [32](#)
Struktur der Angriffspunkte
L_TW2P_scAP_WinderDancerCtrlState [36](#)
Struktur des Signalflusses
L_TW2P_scSF_WinderDancerCtrlState [34](#)
Synchronisierung auf die Liniengeschwindigkeit [81](#)

T

Tänzerlage (Normierung) [82](#)
Tänzerposition korrigieren [69](#)
Tänzerpositions-Überwachung [86](#)
Teaching-Funktion für Tänzerendlagen [84](#)

U

Überwachung der Tänzerposition [86](#)

V

Variablenbezeichner [7](#)

Verwendete Konventionen [7](#)

W

Wickelprozess beenden [108](#)

Wickelrichtung (Automatische Erkennung) [59](#)

Wickelrichtung festlegen (Aufwickeln/Abwickeln) [58](#)

Z

Zielgruppe [5](#)

Zugkraftsteuerung über Kennlinienfunktion (Base-Variante)
[90](#)

Zustände [32](#)



Ihre Meinung ist uns wichtig

Wir erstellen diese Anleitung nach bestem Wissen mit dem Ziel, Sie bestmöglich beim Umgang mit unserem Produkt zu unterstützen.

Vielleicht ist uns das nicht überall gelungen. Wenn Sie das feststellen sollten, senden Sie uns Ihre Anregungen und Ihre Kritik in einer kurzen E-Mail an:

feedback-docu@lenze.com

Vielen Dank für Ihre Unterstützung.

Ihr Lenze-Dokumentationsteam

