



# Technologiemodul

Winder Dancer-controlled \_\_\_\_\_

Referenzhandbuch

DE



13551073

**Lenze**

<b>1</b>	<b>Über diese Dokumentation</b>	<b>4</b>
1.1	Dokumenthistorie	6
1.2	Verwendete Konventionen	7
1.3	Definition der verwendeten Hinweise	8
<b>2</b>	<b>Sicherheitshinweise</b>	<b>9</b>
<b>3</b>	<b>Funktionsbeschreibung "Winder Dancer-controlled"</b>	<b>11</b>
3.1	Übersicht der Funktionen	12
3.2	Wichtige Hinweise zum Betrieb des Technologiemoduls	13
3.3	Funktionsbaustein L_TT1P_WinderDancerCtrl[Base/State/High]	15
3.3.1	Eingänge und Ausgänge	16
3.3.2	Eingänge	17
3.3.3	Ausgänge	21
3.3.4	Parameter	23
3.4	State machine	30
3.5	Signalflusspläne	32
3.5.1	Struktur des Signalflusses	34
3.5.2	Struktur der Angriffspunkte	36
3.6	Festlegung der Wickelrichtung (Aufwickeln/Abwickeln)	38
3.7	Automatische Erkennung der Wickelrichtung	38
3.8	Festlegung der Materialzuführung an den Wickler	39
3.9	Leitwert-Quelle für die Durchmesserberechnung	40
3.10	Drehzahlvorsteuerung	41
3.11	Durchmesserberechnung	42
3.12	Durchmesser halten	43
3.13	Durchmesser vorgeben / Signal vom Durchmessersensor	44
3.14	Durchmesserberechnung mit Korrektur der Tänzerposition	45
3.15	Materiallängenzähler	47
3.16	Quellen für die Materiallängenzählung	48
3.16.1	Quelle: Eingang „lrSetLineVel“	48
3.16.2	Quelle: Eingang „lrSetLineVelDiamCalc“	48
3.16.3	Quelle: Eingang „MaterialCounterAxis“ (Referenzachse)	49
3.17	Handfahren (Jogging)	50
3.18	Synchronisierung auf die Liniengeschwindigkeit	52
3.19	Trimmung	53
3.20	Normierung der Tänzerlage	54
3.21	Teaching-Funktion für Tänzerendlagen	55
3.22	Überwachung der Tänzerposition	56
3.23	PI-Regler für die Tänzerlageregelung	57
3.24	Zugkraftsteuerung über Kennlinienfunktion (Base-Variante)	58
3.25	Bahnrißüberwachung	59
3.26	Persistente Variablen	60
3.27	Beschleunigungskompensation	62
3.28	Zugkraftsteuerung über Kennlinienfunktion/Wickelcharakteristik	64
3.29	Identifikation der Massenträgheitsmomente	66
3.30	Adaption der Drehzahlregler-Verstärkung	68
3.30.1	Adaptionsmodus eAdaptSpdCtrlGainMode:= 0 (DiamToSquare)	68
3.30.2	Adaptionsmodus eAdaptSpdCtrlGainMode:= 1 (Diam)	69
3.30.3	Adaptionsmodus eAdaptSpdCtrlGainMode:= 2 (Inertia)	69
3.31	Regelabweichung im Bereich reduzierter Empfindlichkeit	71
3.32	Beendigung des Wickelprozesses	72
3.33	Begrenzung der Master-Liniengeschwindigkeit	73
3.34	Identifikation des Durchmessers durch Anheben des Tänzers	74
3.35	CPU-Auslastung (Beispiel Controller 3231 C)	76

# Inhalt

---

<b>Index</b>	77
Ihre Meinung ist uns wichtig	79

## 1 Über diese Dokumentation

Diese Dokumentation ...

- enthält ausführliche Informationen zu den Funktionalitäten des Technologiemoduls "Winder Dancer-controlled";
- 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 <ul style="list-style-type: none"><li>• Controller-based Automation EtherCAT®</li><li>• Controller-based Automation CANopen®</li><li>• Controller-based Automation PROFIBUS®</li><li>• Controller-based Automation PROFINET®</li></ul>
Referenzhandbücher Online-Hilfen	Lenze-Controller: <ul style="list-style-type: none"><li>• Controller 3200 C</li><li>• Controller c300</li><li>• Controller p300</li><li>• Controller p500</li></ul>
Software-Handbücher Online-Hilfen	Lenze Engineering Tools: <ul style="list-style-type: none"><li>• »PLC Designer« (Programmierung)</li><li>• »Engineer« (Parametrierung, Konfigurierung, Diagnose)</li><li>• »VisiWinNET® Smart« (Visualisierung)</li><li>• »Backup &amp; Restore« (Datensicherung, Wiederherstellung, Aktualisierung)</li></ul>

## 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	
<input type="checkbox"/>	<b>Produktkataloge</b> <ul style="list-style-type: none"><li>• Controller-based Automation</li><li>• Controller</li><li>• Inverter Drives/Servo Drives</li></ul>
Montage und Verdrahtung	
<input type="checkbox"/>	<b>Montageanleitungen</b> <ul style="list-style-type: none"><li>• Controller</li><li>• Kommunikationskarten (MC-xxx)</li><li>• I/O-System 1000 (EPM-Sxxx)</li><li>• Inverter Drives/Servo Drives</li><li>• Kommunikationsmodule</li></ul>
<input type="checkbox"/>	<b>Gerätehandbücher</b> <ul style="list-style-type: none"><li>• Inverter Drives/Servo Drives</li></ul>
Parametrierung / Konfigurierung / Inbetriebnahme	
<input type="checkbox"/>	<b>Online-Hilfe / Referenzhandbücher</b> <ul style="list-style-type: none"><li>• Controller</li><li>• Inverter Drives/Servo Drives</li><li>• I/O-System 1000 (EPM-Sxxx)</li></ul>
<input type="checkbox"/>	<b>Online-Hilfe / Kommunikationshandbücher</b> <ul style="list-style-type: none"><li>• Bussysteme</li><li>• Kommunikationsmodule</li></ul>
Beispielapplikationen und Vorlagen	
<input type="checkbox"/>	<b>Online-Hilfe / Software- und Referenzhandbücher</b> <ul style="list-style-type: none"><li>• Application Sample i700</li><li>• Application Samples 8400/9400</li><li>• FAST Application Template Lenze/PackML</li><li>• FAST Technologiemodule</li></ul>

### 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](http://www.lenze.com)

## Zielgruppe

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

## 1.1


## Dokumenthistorie

Version			Beschreibung
5.2	03/2019	TD06	Fehler korrigiert
5.1	02/2019	TD29	Fehler korrigiert
5.0	05/2018	TD29	Erweitert: <ul style="list-style-type: none"> <li>• <a href="#">Adaption der Drehzahlregler-Verstärkung</a> (☞ 68)</li> </ul> Neu: <ul style="list-style-type: none"> <li>• <a href="#">Begrenzung der Master-Liniengeschwindigkeit</a> (☞ 73)</li> <li>• <a href="#">Identifikation des Durchmessers durch Anheben des Tänzers</a> (☞ 74)</li> </ul>
4.3	05/2017	TD17	<ul style="list-style-type: none"> <li>• Inhaltliche Struktur geändert.</li> <li>• Allgemeine Korrekturen</li> <li>• Abbildung <a href="#">Signalfluss des Technologiomoduls</a> (☞ 33) korrigiert.</li> </ul> Neu: <ul style="list-style-type: none"> <li>• Eingang "MaterialCounterAxis" (AXIS_REF)</li> <li>• <a href="#">Quellen für die Materiallängenzählung</a> (☞ 48)</li> </ul>
4.2	11/2016	TD17	<ul style="list-style-type: none"> <li>• Allgemeine Korrekturen</li> <li>• Angriffspunkte <a href="#">L TT1P_scAP_WinderDancerCtrl [Base/State/High]</a> (☞ 36) ergänzt.</li> </ul>
4.1	04/2016	TD17	<ul style="list-style-type: none"> <li>• Allgemeine Korrekturen</li> <li>• Abbildung <a href="#">State machine</a> (☞ 30) korrigiert.</li> <li>• Abbildung <a href="#">Signalfluss des Technologiomoduls</a> (☞ 33) korrigiert.</li> <li>• Angriffspunkte <a href="#">L TT1P_scAP_WinderDancerCtrl [Base/State/High]</a> (☞ 36) ergänzt.</li> </ul>
4.0	11/2015	TD17	<ul style="list-style-type: none"> <li>• Allgemeine Korrekturen</li> <li>• Neu:  <ul style="list-style-type: none"> <li>▶ <a href="#">Regelabweichung im Bereich reduzierter Empfindlichkeit</a> (☞ 71)</li> <li>▶ <a href="#">Beendigung des Wickelprozesses</a> (☞ 72)</li> </ul> </li> <li>• Inhaltliche Struktur geändert.</li> </ul>
3.0	05/2015	TD17	<ul style="list-style-type: none"> <li>• Allgemeine Korrekturen</li> <li>• Neu: <a href="#">Materiallängenzähler</a> (☞ 47)</li> </ul>
2.0	01/2015	TD17	<ul style="list-style-type: none"> <li>• Allgemeine redaktionelle Überarbeitung</li> <li>• Modularisierung der Inhalte für die »PLC Designer« Online-Hilfe</li> </ul>
1.0	04/2014	TD00	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
Textauszeichnung		
Programmname	» «	»PLC Designer« ...
Variablenbezeichner	<i>kursiv</i>	Durch Setzen von <i>bEnable</i> auf TRUE ...
Funktionsbausteine	<b>fett</b>	Der Funktionsbaustein <b>L_MC1P_AxisBasicControl</b> ...
Funktionsbibliotheken		Die Funktionsbibliothek <b>L_TT1P_TechnolgyModules</b> ...
Quellcode	Schriftart "Corier new"	... dwNumerator := 1; dwDenominator := 1; ...
Symbole		
Seitenverweis	 7)	Verweis auf weiterführenden Informationen: Seitenzahl in PDF-Datei.

### 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 Über diese Dokumentation

## 1.3 Definition der verwendeten Hinweise

### 1.3 Definition der verwendeten Hinweise

Um auf Gefahren und wichtige Informationen hinzuweisen, werden in dieser Dokumentation folgende Signalwörter und Symbole verwendet:

#### Sicherheitshinweise

Aufbau der Sicherheitshinweise:



#### Piktogramm und Signalwort!

(kennzeichnen die Art und die Schwere der Gefahr)

#### Hinweistext

(beschreibt die Gefahr und gibt Hinweise, wie sie vermieden werden kann)

Piktogramm	Signalwort	Bedeutung
	Gefahr!	<b>Gefahr von Personenschäden durch gefährliche elektrische Spannung</b> Hinweis auf eine unmittelbar drohende Gefahr, die den Tod oder schwere Verletzungen zur Folge haben kann, wenn nicht die entsprechenden Maßnahmen getroffen werden.
	Gefahr!	<b>Gefahr von Personenschäden durch eine allgemeine Gefahrenquelle</b> Hinweis auf eine unmittelbar drohende Gefahr, die den Tod oder schwere Verletzungen zur Folge haben kann, wenn nicht die entsprechenden Maßnahmen getroffen werden.
	Stop!	<b>Gefahr von Sachschäden</b> Hinweis auf eine mögliche Gefahr, die Sachschäden zur Folge haben kann, wenn nicht die entsprechenden Maßnahmen getroffen werden.

#### Anwendungshinweise

Piktogramm	Signalwort	Bedeutung
	Hinweis!	Wichtiger Hinweis für die störungsfreie Funktion
	Tipp!	Nützlicher Tipp für zum einfachen Bedienen
		Verweis auf andere Dokumentation



## 2 Sicherheitshinweise

Beachten Sie die Sicherheitshinweise in dieser Dokumentation, wenn Sie ein Automationssystem 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 Automationssystems 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 Automationssystems 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 Invertern 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.



## Stop!

### **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**

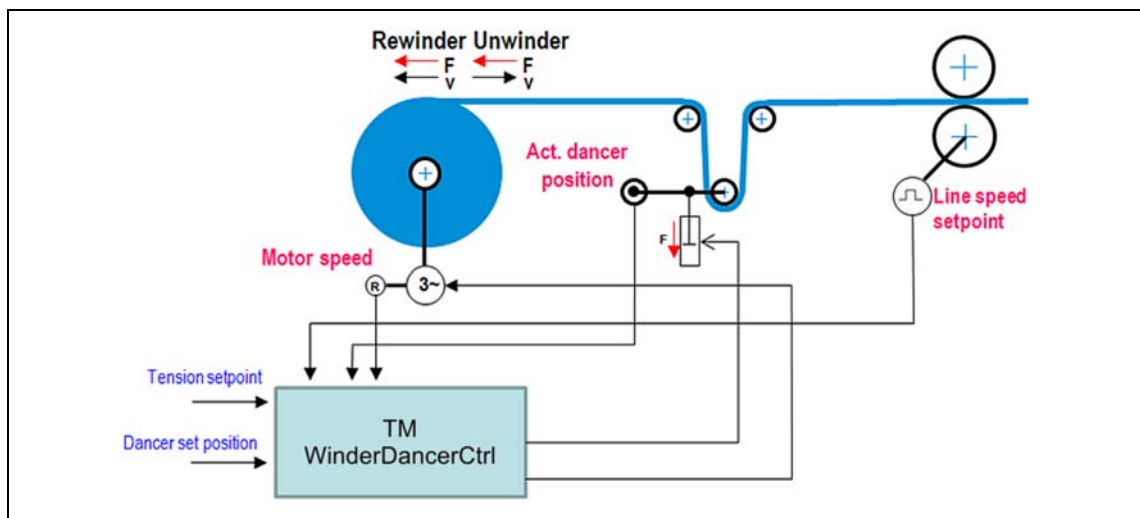
- Vor allen Arbeiten an den Komponenten des Automationssystems immer die Spannungsversorgung ausschalten.
- Elektronische Bauelemente und Kontakte nur berühren, wenn zuvor ESD-Maßnahmen getroffen wurden.
- Anlagen mit eingebauten Invertern ggf. mit zusätzlichen Überwachungs- und Schutzeinrichtungen nach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

### 3 Funktionsbeschreibung "Winder Dancer-controlled"

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

Mit diesem Technologiemodul kann ein tänzerlagegelter Wickelantrieb projiziert werden.



[3-1] Aufbau eines tänzergeregelten Wicklers

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 Tänzerlage-regler für eine Korrektur des Drehzahlsollwertes.

- In der Variante "Base" erfolgt die Tänzerlageregelung mit der Berechnung des Durchmessers. Für die Tänzerlageregelung ist es möglich, einen PI-Regler zu verwenden. Die Tänzerendlagen können über die Parameterstruktur [L TT1P\\_scPar WinderDancerCtrl \[Base/State/High\]](#) ([□ 23](#)) 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.
- In der Variante "State" ist der Funktionsumfang der Base-Variante erweitert. Hierbei 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 Beschleunigungskom-pensation vorgesteuert werden.

- Die Variante "High" bietet ergänzend die Möglichkeit, das Massenträgheitsmoment der Wick-lerachse zu identifizieren und für die Parametrierung des Technologiemoduls zu verwenden. Zudem ist eine Adaption der Drehzahlreglerverstärkung, in Abhängigkeit des aktuellen Massen-trägheitsmoments, im laufenden Betrieb ausführbar.

### 3.1 Übersicht der Funktionen

Neben den Grundfunktionen zur Bedienung des Funktionsbausteins **L\_MC1P\_AxisBasicControl**, der **Stopp-Funktion** und der **Halt-Funktion** bietet das Technologiemodul folgende Funktionalitäten, die den Varianten "Base", "State" und "High" zugeordnet sind:

Funktionalität	Variante		
	Base	State	High
<a href="#">Festlegung der Wickelrichtung (Aufwickeln/Abwickeln) (□ 38)</a>	●	●	●
<a href="#">Automatische Erkennung der Wickelrichtung (□ 38)</a>	●	●	●
<a href="#">Festlegung der Materialzuführung an den Wickler (□ 39)</a>	●	●	●
<a href="#">Leitwert-Quelle für die Durchmesserberechnung (□ 40)</a>	●	●	●
<a href="#">Drehzahlvorsteuerung (□ 41)</a>	●	●	●
<a href="#">Durchmesserberechnung (□ 42)</a>	●	●	●
<a href="#">Durchmesser halten (□ 43)</a>	●	●	●
<a href="#">Durchmesser vorgeben / Signal vom Durchmessersensor (□ 44)</a>	●	●	●
<a href="#">Durchmesserberechnung mit Korrektur der Tänzerposition (□ 45)</a>	●	●	●
<a href="#">Materiallängenzähler (□ 47)</a>	●	●	●
<a href="#">Quellen für die Materiallängenzählung (□ 47)</a>	●	●	●
<a href="#">Handfahren (Jogging) (□ 50)</a>	●	●	●
<a href="#">Synchronisierung auf die Liniengeschwindigkeit (□ 52)</a>	●	●	●
<a href="#">Trimmung (□ 53)</a>	●	●	●
<a href="#">Normierung der Tänzerlage (□ 54)</a>	●	●	●
<a href="#">Teaching-Funktion für Tänzerendlagen (□ 55)</a>	●	●	●
<a href="#">Überwachung der Tänzerposition (□ 56)</a>	●	●	●
<a href="#">PI-Regler für die Tänzerlageregelung (□ 57)</a>	●	●	●
<a href="#">Zugkraftsteuerung über Kennlinienfunktion (Base-Variante) (□ 58)</a>	●	●	●
<a href="#">Bahnrißüberwachung (□ 59)</a>	●	●	●
<a href="#">Persistente Variablen (□ 60)</a>	●	●	●
<a href="#">Begrenzung der Master-Liniengeschwindigkeit (□ 73)</a>	●	●	●
<a href="#">Beschleunigungskompensation (□ 62)</a>		●	●
<a href="#">Zugkraftsteuerung über Kennlinienfunktion/Wickelcharakteristik (□ 64)</a>		●	●
<a href="#">Identifikation der Massenträgheitsmomente (□ 66)</a>			●
<a href="#">Adaption der Drehzahlregler-Verstärkung (□ 68)</a>			●
<a href="#">Identifikation des Durchmessers durch Anheben des Tänzers</a>			●
<a href="#">Regelabweichung im Bereich reduzierter Empfindlichkeit (□ 71)</a>			●
<a href="#">Beendigung des Wickelprozesses (□ 72)</a>			●



#### »PLC Designer« Online-Hilfe

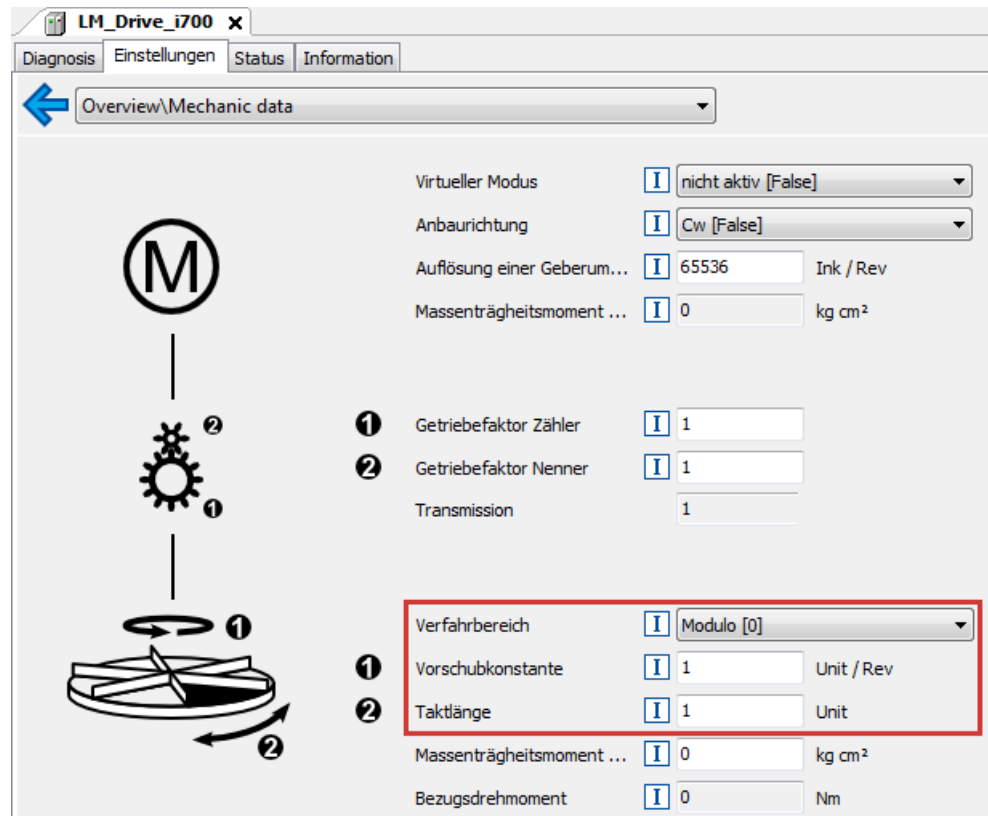
Hier finden Sie ausführliche Informationen zum Funktionsbaustein **L\_MC1P\_AxisBasicControl**, zur **Stopp-Funktion** und zur **Halt-Funktion**.

### 3.2 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:



- Der Vorschub der Wicklerachse wird in der Einheit [revs/s] parametrier.
- Die Geschwindigkeit der Linie wird in der Einheit [mm/s] parametrier.

#### Einstellung des Betriebsmodus

Der Betriebsmodus (Mode of Operation) für die Wickler-Achse muss auf "Zyklisch synchrone Position" (csp) eingestellt werden, da die Achse über den Positions-, Geschwindigkeits- und Drehmomentleitwert geführt wird.

**Kontrollierter Anlauf der Achsen**

Bewegungsbefehle, die im gesperrten Achszustand ( $xAxisEnabled = FALSE$ ) gesetzt werden, müssen nach der Freigabe ( $xRegulatorOn = TRUE$ ) erneut durch eine  $FALSE \nearrow TRUE$ -Flanke aktiviert werden.

So wird verhindert, dass der Antrieb nach der Reglerfreigabe unkontrolliert anläuft.

**Beispiel Handfahren (Jogging) (50):**

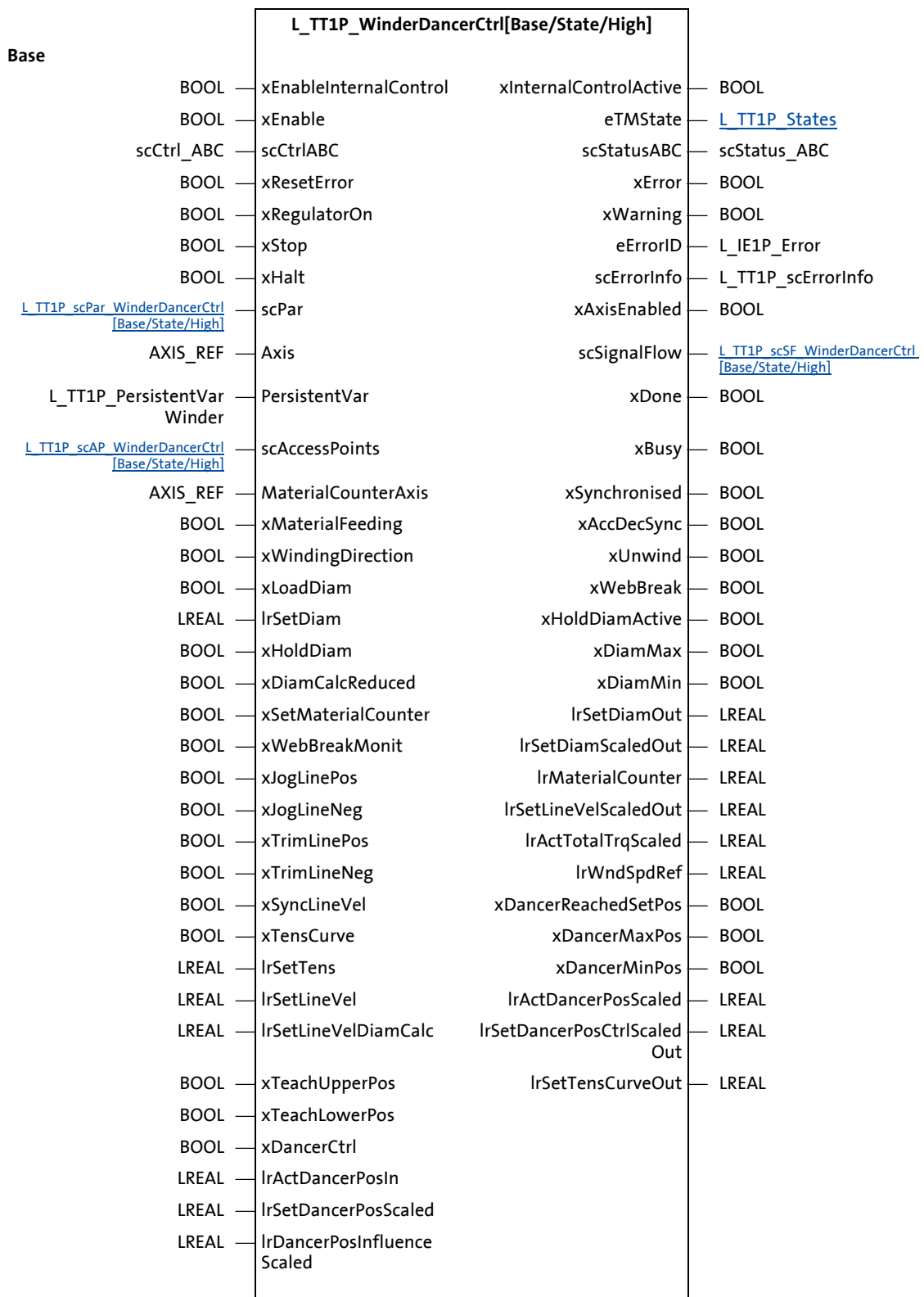
1. Im gesperrten Achszustand ( $xAxisEnabled = FALSE$ ) wird  $xJogPos = TRUE$  gesetzt.
  - $xRegulatorOn = FALSE$  (Achse ist gesperrt.)  
==> Zustand "READY" ( $xAxisEnabled = FALSE$ )
  - $xJogPos = TRUE$  (Handfahren soll ausgeführt werden.)
2. Achse freigeben.
  - $xRegulatorOn = TRUE$   
==> Zustand "READY" ( $xAxisEnabled = TRUE$ )
3. Handfahren ausführen.
  - $xJogPos = FALSE \nearrow TRUE$   
==> Zustand "JOGPOS"

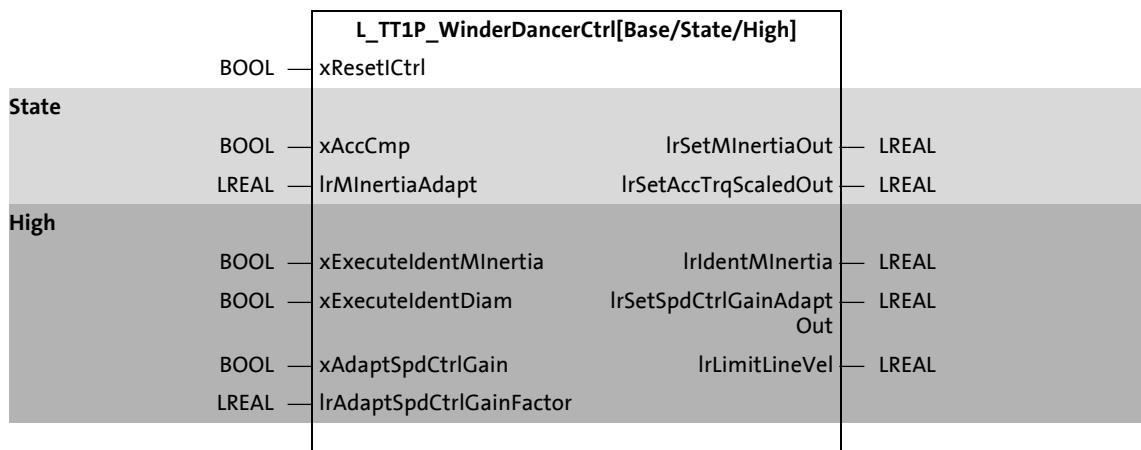
## 3.3

## Funktionsbaustein L\_TT1P\_WinderDancerCtrl[Base/State/High]

Die Abbildung zeigt die Zugehörigkeit der Ein- und Ausgänge für die Varianten "Base", "State" und "High".

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





## 3.3.1

## Eingänge und Ausgänge

Bezeichner	Datentyp	Beschreibung	Verfügbar in Variante		
			Base	State	High
Axis	AXIS_REF	Referenz auf die Achse	●	●	●
PersistentVar L_TT1P_PersistentVar Winder		Referenz auf Persistente Variablen In der Referenz werden folgende Daten verwaltet: <ul style="list-style-type: none"> <li>Berechneter Durchmesser</li> <li>"Erlernte" Tänzerendlagen (siehe <a href="#">Teaching-Funktion für Tänzerendlagen</a> (□ 55))</li> </ul>	●	●	●



## 3.3.2 Eingänge

Bezeichner Datentyp	Beschreibung		Verfügbar in Variante		
			Base	State	High
xEnableInternalControl BOOL	TRUE	In der Visualisierung ist die interne Steuerung der Achse über die Schaltfläche "Internal Control" auswählbar.	●	●	●
xEnable BOOL	Ausführung des Funktionsbausteins		●	●	●
	TRUE	Der Funktionsbaustein wird ausgeführt.			
	FALSE	Der Funktionsbaustein wird nicht ausgeführt.			
scCtrlABC scCtrl_ABC	Eingangsstruktur für den Funktionsbaustein <b>L_MC1P_AxisBasicControl</b> <ul style="list-style-type: none"> <li>scCtrlABC kann im Zustand "Ready" genutzt werden.</li> <li>Liegt eine Anforderung an, wird in den Zustand "Service" gewechselt.</li> <li>Vom Zustand "Service" wird zurück in den Zustand "Ready" gewechselt, wenn keine Anforderung mehr anliegt.</li> </ul>		●	●	●
xResetError BOOL	TRUE	Fehler der Achse oder der Software zurücksetzen.	●	●	●
xRegulatorOn BOOL	TRUE	Reglerfreigabe der Achse aktivieren (über den Funktionsbaustein <b>MC_Power</b> ).	●	●	●
xStop BOOL	TRUE	Aktive Bewegung abbrechen und Achse mit der über den Parameter IrStopDec definierten Verzögerung in den Stillstand führen. <ul style="list-style-type: none"> <li>Ein Wechsel in den Zustand "Stop" erfolgt.</li> <li>Zustand "STOP" wird verlassen, wenn (Not xStop AND Not xHalt) AND eAxisState = StandStill.</li> <li>Der Eingang ist auch bei "Internal Control" aktiv.</li> </ul>	●	●	●
xHalt BOOL	TRUE	Aktive Bewegung abbrechen und Achse mit der über den Parameter IrHaltDec definierten Verzögerung in den Stillstand führen. <ul style="list-style-type: none"> <li>Ein Wechsel in den Zustand "Stop" erfolgt.</li> <li>Das Technologiemodul bleibt im Zustand "Stop", solange xHalt = TRUE (oder xStop = TRUE).</li> <li>Der Zustand "Stop" kann erst verlassen werden, wenn die Achse still steht.</li> </ul>	●	●	●
scPar <a href="#">L_TT1P_scPar_WinderDancerCtrl[Base/State/High]</a>	Die Parameterstruktur enthält die Parameter des Technologiemoduls. Der Datentyp ist abhängig von der verwendeten Variante (Base/State/High).		●	●	●
scAccessPoints <a href="#">L_TT1P_scAP_WinderDancerCtrl[Base/State/High]</a>	Struktur der Angriffspunkte Der Datentyp ist abhängig von der verwendeten Variante (Base/State/High).		●	●	●
MaterialCounterAxis AXIS_REF	Hier kann eine Modulo-Achse eines Messrades auf dem Material angeschlossen werden. Wenn eine Achse am Eingang angeschlossen ist, so erfolgt die Erhöhung 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 Intergration der Materialgeschwindigkeit (Eingang IrSetLineVel oder IrSetLineVelDiamCalc). <a href="#">▶ Materiallängenzähler (47)</a> <a href="#">▶ Quellen für die Materiallängenzählung (48)</a>		●	●	●

Bezeichner	Datentyp	Beschreibung		Verfügbar in Variante		
				Base	State	High
xMaterialFeeding	BOOL	Materialführung von oben oder unten an den Wickelballen • Initialwert: FALSE		●	●	●
		TRUE	Materialführung von oben			
		FALSE	Materialführung von unten			
xWindingDirection	BOOL	Funktion des Wicklers bei positiver Liniengeschwindigkeit (Eingang IrSetLineVel > 0) • Initialwert: FALSE		●	●	●
		TRUE	Abwickler			
		FALSE	Aufwickler			
xLoadDiam	BOOL	TRUE	Den (Start-)Durchmesser aus dem Eingang IrSetDiam laden. • Initialwert: FALSE	●	●	●
IrSetDiam	LREAL	Vorgabe eines (Start-)Durchmessers Der Durchmesser wird zyklisch geladen wenn der Eingang xLoadDiam = TRUE gesetzt ist. • Einheit: mm • Initialwert: 0		●	●	●
xHoldDiam	BOOL	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.			
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.			
xWebBreakMonit	BOOL	Initialwert: FALSE		●	●	●
		TRUE	Bahnrißüberwachung aktivieren.			
		FALSE	Bahnrißüberwachung deaktivieren.			
xJogLinePos	BOOL	TRUE	Achse in positive Materialflussrichtung fahren (Handfahren). Ist xJogLineNeg auch TRUE, wird die Fahrrichtung beibehalten, die zuerst gewählt wurde.	●	●	●
xJogLineNeg	BOOL	TRUE	Achse in negative Materialflussrichtung fahren (Handfahren). Ist xJogLinePos auch TRUE, wird die Fahrrichtung beibehalten, die zuerst gewählt wurde.	●	●	●
xTrimLinePos	BOOL	TRUE	Den Geschwindigkeits-Offset in positive Materialflussrichtung freigeben, wenn die Wicklerachse auf die Linie synchronisiert ist (xSynclineVel = TRUE).	●	●	●
xTrimLineNeg	BOOL	TRUE	Den Geschwindigkeits-Offset in negative Materialflussrichtung freigeben, wenn die Wicklerachse auf die Linie synchronisiert ist (xSynclineVel = TRUE).	●	●	●
xSynclineVel	BOOL	TRUE	Wicklerachse auf die Linie synchronisieren.	●	●	●

Bezeichner	Datentyp	Beschreibung		Verfügbar in Variante		
				Base	State	High
xTensCurve	BOOL	Initialwert: FALSE		●	●	●
		TRUE	Zugkraftkennlinie freigeben.			
		FALSE	Zugkraftkennlinie sperren.			
IrSetTens	LREAL	Zugkraftsollwert • Einheit: N • Initialwert: 0		●	●	●
IrSetLineVel	LREAL	Aktuelle Liniengeschwindigkeit • Einheit: mm/s		●	●	●
IrSetLineVelDiamCalc	LREAL	Aktuelle Liniengeschwindigkeit für die Durchmesserberechnung • Einheit: mm/s		●	●	●
xTeachUpperPos	BOOL	TRUE	Die aktuelle Tänzerposition wird als oberer Grenzwert gespeichert.	●	●	●
xTeachLowerPos	BOOL	TRUE	Die aktuelle Tänzerposition wird als unterer Grenzwert gespeichert.	●	●	●
xDancerCtrl	BOOL	TRUE	Tänzerlageregelung aktivieren.	●	●	●
IrActDancerPosIn	LREAL	Aktuelle Tänzerposition Die Istposition des Tänzers wird in Form eines analogen Signals (0 ... 10 V) an den Controller zurückgeführt.		●	●	●
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: 0.1 • Wertebereich: 0 ... 1 (0 ... 100 %)		●	●	●
xResetICtrl	BOOL	Der I-Anteil des PI-Reglers kann ausgeschaltet werden und die Stellgröße (Ausgang des Reglers) aus dem I-Anteil kann über die Rampenfunktion auf '0' geführt werden. Die Stellgröße aus dem P-Anteil wird nicht beeinflusst. • Initialwert: FALSE		●	●	●
		TRUE	Funktionalität aktivieren			
		FALSE	Funktionalität deaktivieren			
xAccCmps	BOOL	Beschleunigungskompensation während der Tänzerlageregelung aktivieren/deaktivieren • Initialwert: FALSE			●	●
		TRUE	Beschleunigungskompensation während der Tänzerlageregelung aktivieren			
		FALSE	Beschleunigungskompensation während der Tänzerlageregelung deaktivieren			
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 Wicklerwelle wird ermittelt. Am Ausgang IrIdentMinertia wird das ermittelte Massenträgheitsmoment in kgcm <sup>2</sup> angezeigt.			

Bezeichner	Datentyp	Beschreibung		Verfügbar in Variante		
				Base	State	High
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 einschalten.			
		FALSE	Adaption der Drehzahlreglerverstärkung ausschalten.			
lrAdaptSpdCtrlGainFactor	LREAL	Der resultierende Wert der Adaption Drehzahlreglerverstärkung kann über diesen Eingang multiplikativ beeinflusst werden. • Wertebereich: 0 ... 1 • Initialwert: 1				●

## 3.3.3 Ausgänge

Bezeichner Datentyp	Beschreibung	Verfügbar in Variante		
		Base	State	High
xInternalControlActive BOOL	Die interne Steuerung der Achse ist über die Visualisierung aktiviert. (Eingang xEnableInternalControl = TRUE)	●	●	●
eTMState <a href="#">L_TT1P_States</a>	Aktueller Zustand des Technologiemoduls ► <a href="#">State machine</a> (30)	●	●	●
scStatusABC scStatus_ABC	Struktur der Zustandsdaten des Funktionsbausteins <b>L_MC1P_AxisBasicControl</b>	●	●	●
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 xError = TRUE oder xWarning = TRUE ist.  <b>Referenzhandbuch "FAST Technologiemodule":</b> Hier finden Sie Informationen zu Fehler- oder Warnungsmeldungen.	●	●	●
scErrorInfo L_TT1P_scErrorInfo	Fehlerinformationsstruktur für eine genauere Analyse der Fehlerursache	●	●	●
xAxisEnabled BOOL	TRUE Die Achse ist freigegeben.	●	●	●
scSignalFlow <a href="#">L_TT1P_scSF_WinderDancerCtrl</a> <a href="#">[Base/State/High]</a>	Struktur des Signalflusses Der Datentyp ist abhängig von der verwendeten Variante (Base/State/High). ► <a href="#">Signalflusspläne</a> (32)	●	●	●
xDone BOOL	TRUE Die Anforderung/Aktion wurde erfolgreich abgeschlossen.	●	●	●
xBusy BOOL	TRUE Die Anforderung/Aktion wird zur Zeit ausgeführt.	●	●	●
xSynchronised BOOL	TRUE Der Wickler ist auf die Liniengeschwindigkeit synchronisiert.	●	●	●
xAccDecSync BOOL	TRUE Die Synchronisierungsfunktion ist aktiv. Der Wickler wird auf- oder asynchronisiert.	●	●	●
xUnwind BOOL	Statusbit für Auf- und Abwickler		●	●
	TRUE	Abwickler		
	FALSE	Aufwickler		
xWebBreak BOOL	TRUE Ein Bahnriß liegt vor.	●	●	●
xHoldDiamActive BOOL	TRUE Der aktuelle Durchmesser wird gehalten.	●	●	●
xDiamMax BOOL	TRUE Der maximale Durchmesser wurde erreicht.	●	●	●
xDiamMin BOOL	TRUE Der minimale Durchmesser wurde erreicht.	●	●	●
lrSetDiamOut LREAL	Aktueller berechneter Durchmesser • Einheit: mm	●	●	●
lrSetDiamScaledOut LREAL	Aktueller berechneter skaliert Durchmesser • Einheit: x 100 % • 1 = 100 % = Parameter lrMaxDiam	●	●	●

Bezeichner	Datentyp	Beschreibung		Verfügbar in Variante		
				Base	State	High
IrMaterialCounter	LREAL	Anzeige des Materiallängenzählerstandes auf dem Wickler Je nach <a href="#">Festlegung der Wickelrichtung (Aufwickeln/Abwickeln)</a> (38) wird der Materiallängenzähler hoch- oder runtergezählt. • Einheit: mm		●	●	●
IrSetLineVelScaledOut	LREAL	Aktuelle skalierte Liniengeschwindigkeit • Einheit: x 100 % • 1 = 100 % = Parameter IrLineVelRef		●	●	●
IrActTotalTrqScaled	LREAL	Aktuelles skaliertes Drehmoment der Wicklerwelle • Bezugsgröße: Nenn-/Bezugsdrehmoment des Motors. • Einheit: x 100 % (1 = 100 %)		●	●	●
IrWndSpdRef	LREAL	Referenz der Wicklerdrehzahl bei minimalem Durchmesser und maximaler Liniengeschwindigkeit.		●	●	●
xDancerReachedSetPos	BOOL	TRUE Der Tänzer hat die Sollposition erreicht.		●	●	●
xDancerMaxPos	BOOL	TRUE Der Tänzer hat die obere Grenzposition erreicht.		●	●	●
xDancerMinPos	BOOL	TRUE Der Tänzer hat die untere Grenzposition erreicht.		●	●	●
IrActDancerPosScaled	LREAL	Aktuelle skalierte Tänzerposition • Einheit: x 100 % • Wertebereich: -1 ... 1 (-100 ... 100 %)		●	●	●
IrSetDancerCtrlScaledOut	LREAL	Aktuelle Stellgröße des Tänzerlagereglers • 1 = 100 % = Parameter IrLineVelRef		●	●	●
IrSetTensCurveOut	LREAL	Aktueller Zugkraftsollwert aus der Kurvenfunktion		●	●	●
IrSetMIInertiaOut	LREAL	Aktuelles Massenträgheitsmoment an der Wicklerwelle • Einheit: kgcm <sup>2</sup>			●	●
IrSetAccTrqScaledOut	LREAL	Drehmomentanteil der Beschleunigungskompensation skaliert auf das Nenn-/Bezugsdrehmoment des Motors.			●	●
IrIdentMIInertia	LREAL	Identifiziertes Massenträgheitsmoment an der Wicklerwelle • Einheit: kgcm <sup>2</sup>				●
IrSetSpdCtrlGainAdaptOut	LREAL	Adaption der Drehzahlreglerverstärkung • Einheit: x 100 % (1 = 100 %)				●
IrLimitLineVel	LREAL	Die maximal erlaubte Liniengeschwindigkeit wird aus dem berechnetem Durchmesser und dem Parameter scPar.IrMaxWndSpd bestimmt. • Einheit [mm/s]				●

### 3.3.4 Parameter

#### L\_TT1P\_scPar\_WinderDancerCtrl [Base/State/High]

Die Struktur L\_TT1P\_scPar\_WinderDancerCtrl[Base/State/High] enthält die Parameter des Technologiemoduls.

Bezeichner	Datentyp	Beschreibung	Verfügbar in Variante		
			Base	State	High
IrStopDec	LREAL	Verzögerung für die Stopp-Funktion und bei Auslösung der Hardware-Endschalter, Software-Endlagen und Schleppfehlerüberwachung • Einheit: revs/s • Initialwert: 10000	●	●	●
IrStopJerk	LREAL	Ruck für die Stopp-Funktion und bei Auslösung der Hardware-Endschalter, Software-Endlagen und Schleppfehlerüberwachung • Einheit: revs/s <sup>3</sup> • Initialwert: 100000	●	●	●
IrHaltDec	LREAL	Verzögerung für die Halt-Funktion Vorgabe, mit welcher Geschwindigkeitsänderung maximal bis zum Stillstand verzögert werden soll. • Einheit: revs/s <sup>2</sup> • Initialwert: 3600 • Nur positive Werte sind zulässig.	●	●	●
IrJerk	LREAL	Ruck zum Ausgleich bei einer Haltfunktion • Einheit: revs/s <sup>3</sup> • Initialwert: 100000	●	●	●
IrLineJerk	LREAL	Ruck für das Handfahren und zum Ausgleich bei einer Trimm- oder Kupplungsfunktion • Einheit: mm/s <sup>3</sup> • Initialwert: 10000	●	●	●
IrJogLineAcc	LREAL	Beschleunigung für das Handfahren Vorgabe, mit welcher Geschwindigkeitsänderung maximal beschleunigt werden soll. • Einheit: mm/s <sup>2</sup> • Initialwert: 100	●	●	●
IrJogLineDec	LREAL	Verzögerung für das Handfahren Vorgabe, mit welcher Geschwindigkeitsänderung maximal bis zum Stillstand verzögert werden soll. • Einheit: mm/s <sup>2</sup> • Initialwert: 100	●	●	●
IrJogLineVel	LREAL	Maximale Geschwindigkeit, mit der das Handfahren durchgeführt werden soll. • Einheit: mm/s • Initialwert: 10	●	●	●
IrTrimLineAcc	LREAL	Beschleunigung für die Trimmung Vorgabe, mit welcher Geschwindigkeitsänderung relativ zur Liniengeschwindigkeit beschleunigt werden soll. Die auf den Antrieb wirkende Beschleunigung ist die Summe aus der Linien- und Trimmbeschleunigung. • Einheit: mm/s <sup>2</sup> • Initialwert: 100	●	●	●

Bezeichner	Datentyp	Beschreibung	Verfügbar in Variante		
			Base	State	High
IrTrimLineDec	LREAL	Verzögerung für die Trimmung Vorgabe, mit welcher Geschwindigkeitsänderung relativ zur Liniengeschwindigkeit verzögert werden soll. Die auf den Antrieb wirkende Verzögerung ist die Summe aus der Linien- und Trimmbeschleunigung. • Einheit: mm/s <sup>2</sup> • Initialwert: 100	●	●	●
IrTrimLineVel	LREAL	Geschwindigkeit für die Trimmung Vorgabe, mit welcher Geschwindigkeit getrimmt werden soll. • Einheit: mm/s • Initialwert: 10	●	●	●
IrSyncLineAcc	LREAL	Beschleunigung zur Synchronisierung auf die Liniengeschwindigkeit • Einheit: mm/s <sup>2</sup> • Initialwert: 100	●	●	●
IrSyncLineDec	LREAL	Verzögerung zur Synchronisierung auf die Liniengeschwindigkeit • Einheit: mm/s <sup>2</sup> • Initialwert: 100	●	●	●
IrWebBreakWindow	LREAL	Bahnrissofenster Der aktuelle Durchmesser wird mit dem vergangenen Durchmesser über das Bahnrissofenster verglichen. • Einheit: x 100 % (1.0 = 100 %) • Initialwert: 0.1 (10 %)	●	●	●
IrMaxDiam	LREAL	Maximaler Durchmesser • Einheit: mm • Initialwert: 180	●	●	●
IrMinDiam	LREAL	Minimale Durchmesser • Einheit: mm • Initialwert: 50	●	●	●
rFiltTimeDiam	REAL	PT1-Filterzeit für den aktuellen Durchmesser (IrSetDiamOut) • 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 IrSetDiam 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	●	●	●



Bezeichner	Datentyp	Beschreibung	Verfügbar in Variante		
			Base	State	High
IrTensCurveCtrlScaled	LREAL	Steigung der Kennlinie für die Zugkraftsteuerung <ul style="list-style-type: none"> <li>Einheit: x 100 % (1 = 100 %)</li> <li>Initialwert: 0</li> <li>Mit dem Wert '1' ergibt sich ein konstanter Zugkraftverlauf und damit ein für den Durchmesser proportional ansteigender Sollwert.</li> </ul>	●	●	●
IrTensCurveStartDiamScaled	LREAL	Anfangspunkt der Kennlinie für die Zugkraftsteuerung <ul style="list-style-type: none"> <li>Einheit: x 100 %</li> <li>1 = 100 % = Parameter IrMaxDiam</li> <li>Initialwert: 0 (0 %)</li> </ul>	●	●	●
IrLineVelRef	LREAL	Maximale Liniengeschwindigkeit <ul style="list-style-type: none"> <li>Einheit: mm/s</li> <li>Initialwert: 1000</li> </ul>	●	●	●
IrMinLineVel	LREAL	Minimale Liniengeschwindigkeit Bis zu dieser Geschwindigkeit wird der Durchmesser gehalten. <ul style="list-style-type: none"> <li>Einheit: mm/s</li> <li>Initialwert: 1</li> </ul>	●	●	●
rFiltTimeMaterialCounter	REAL	Filterzeitkonstante für den Materiallängenzähler (Ausgang IrMaterialCounter) <ul style="list-style-type: none"> <li>Initialwert: 0 (Filter ist deaktiviert.)</li> </ul>	●	●	●
IrSetMaterialPos	LREAL	Position des Materiallängenzählers Mit einer FALSE → TRUE-Flanke am Eingang xSetMaterialCounter wird der Materiallängenzähler (Ausgang IrMaterialCounter) auf den Wert in IrSetMaterialPos gesetzt. <ul style="list-style-type: none"> <li>Einheit: mm</li> </ul>	●	●	●
xLineVelDiamCalc	BOOL	Berechnung des Durchmessers <ul style="list-style-type: none"> <li>Initialwert: FALSE</li> </ul>	●	●	●
		TRUE Für die Berechnung des Durchmessers wird die Geschwindigkeit aus dem Eingang IrSetLineVelDiamCalc verwendet.			
		FALSE Für die Berechnung des Durchmessers wird die Geschwindigkeit aus dem Eingang IrSetLineVel verwendet.			
IrDancerPosRamp	LREAL	Beschleunigungsrampe für die Tänzerlagesollwerte <ul style="list-style-type: none"> <li>Einheit: 1/s</li> <li>Initialwert: 1</li> </ul>	●	●	●
IrDancerPosCtrlGain	LREAL	Reglerverstärkung <ul style="list-style-type: none"> <li>Initialwert: 1</li> </ul>	●	●	●
IrDancerPosCtrlResetTime	LREAL	Reglernachstellzeit <ul style="list-style-type: none"> <li>Einheit: s</li> <li>Initialwert: 0 (Die Regler-Nachstellzeit ist deaktiviert.)</li> </ul>	●	●	●
IrDancerPosCtrlLimPos	LREAL	Begrenzung der Tänzerlageregler-Stellgröße (Ausgang des Reglers) in positive Richtung <ul style="list-style-type: none"> <li>Initialwert: 1</li> </ul>	●	●	●
IrDancerPosCtrlLimNeg	LREAL	Begrenzung der Tänzerlageregler-Stellgröße (Ausgang des Reglers) in negative Richtung <ul style="list-style-type: none"> <li>Initialwert: -1</li> </ul>	●	●	●
IrDancerMaxPosScaled	LREAL	Maximale Tänzerposition für das Statusbit xDancerMaxPos <ul style="list-style-type: none"> <li>Einheit: x 100 % (1 = 100 %)</li> <li>Initialwert: 0.95 (95 %)</li> </ul>	●	●	●
IrDancerMinPosScaled	LREAL	Minimale Tänzerposition für das Statusbit xDancerMinPos <ul style="list-style-type: none"> <li>Einheit: x 100 % (1 = 100 %)</li> <li>Initialwert: -0.95 (-95 %)</li> </ul>	●	●	●

Bezeichner Datentyp	Beschreibung	Verfügbar in Variante		
		Base	State	High
xTeachDancerLimits BOOL	Quelle der Tänzerlagebegrenzungen • Initialwert: FALSE	●	●	●
	TRUE Die Endlagen <b>IrDancerLowerLimit</b> und <b>IrDancerUpperLimit</b> 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 <b>IrDancerLowerLimit</b> und <b>IrDancerUpperLimit</b> werden verwendet.			
IrDancerUpperLimit LREAL	Analoger Wert für die obere Tänzerlagegrenze • Initialwert: 10000000	●	●	●
IrDancerLowerLimit LREAL	Analoger Wert für die untere Tänzerlagegrenze • Initialwert: 0	●	●	●
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 xDancerReachedSetPos anzusteuern. • Einheit: x 100 % (1.0 = 100 %) • Initialwert: 0.2 (20 %)	●	●	●
rFiltTimeActDancerPosIn REAL	PT1-Filterzeit für den Eingang IrActDancerPosIn • 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 Tänzerposition			
	1 Bahnrißüberwachung nur aus der Lage des Tänzers			
	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.		●	●
rFiltTimeAccSpd REAL	PT1-Filterzeit für die Drehzahl der Wicklerwelle zur Beschleunigungskompensation • Einheit: s • Initialwert: 0.005		●	●
IrAccCmpsDeadBandTrq Scaled LREAL	Nacheilbereich (Dead-band) für das aktuelle Beschleunigungsmoment • Einheit: Nm • Initialwert: 0.1		●	●

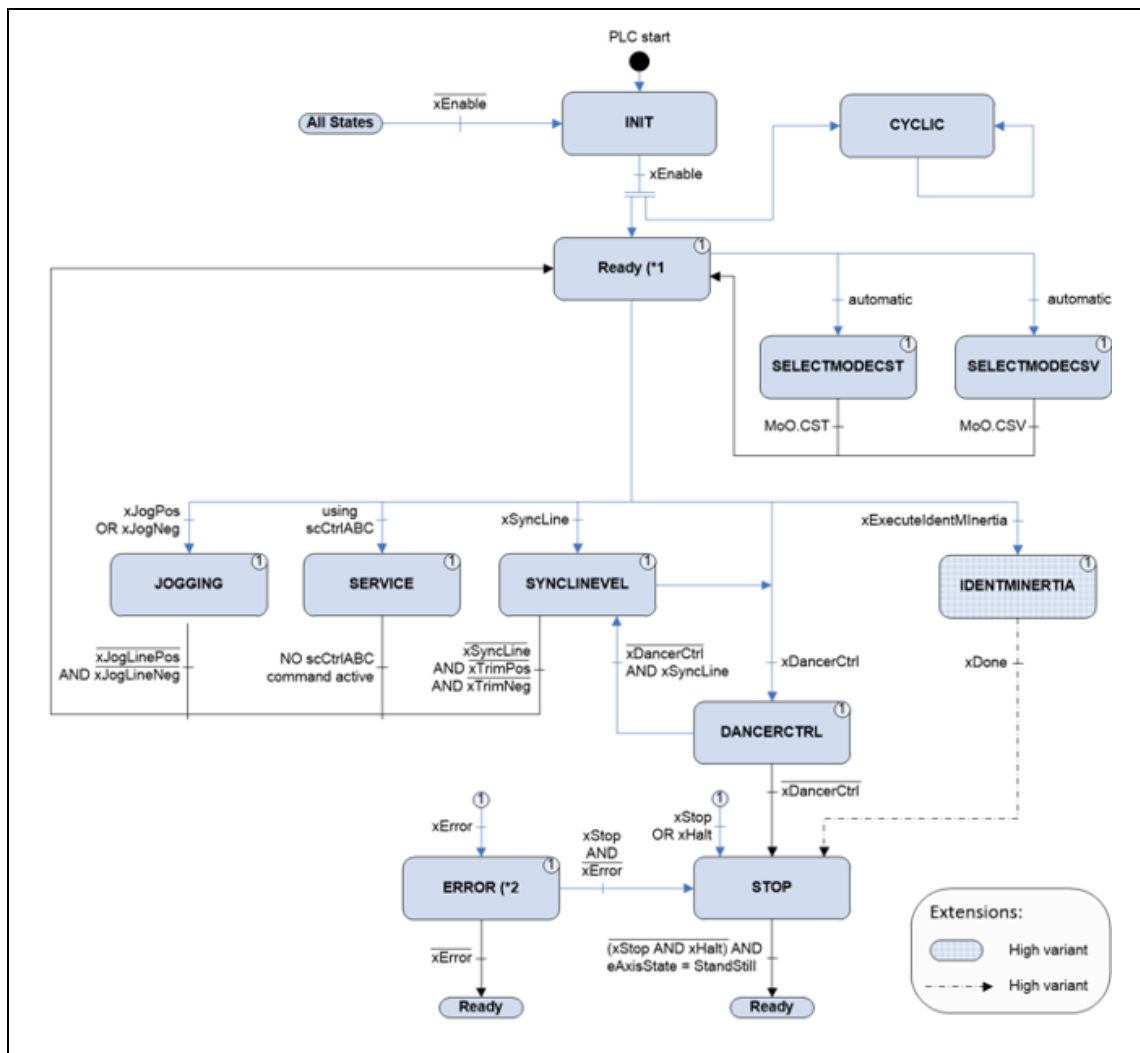
Bezeichner	Datentyp	Beschreibung	Verfügbar in Variante		
			Base	State	High
IrAccCmpsGainAcc	LREAL	Verstärkungsfaktor für das Beschleunigungsmoment in positive Richtung • Einheit: x 100 % (1.00 = 100 %) • Wertebereich: 0 ... 2 (0 ... 200 %) • Initialwert: 1.05 (105 %)		●	●
IrAccCmpsGainDec	LREAL	Verstärkungsfaktor 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 Wicklerwelle • Einheit: kgcm <sup>2</sup> • Initialwert: 9		●	●
IrMaxMIInertia	LREAL	Maximal zulässiges Massenträgheitsmoment an der Wicklerwelle • Einheit: kgcm <sup>2</sup> • Initialwert: 50		●	●
rFiltTimeIdentMIInertiaSpd	REAL	PT1-Filterzeit für die Drehzahl der Wicklerwelle während der Identifikation des Massenträgheitsmoments • Einheit: s • Initialwert: 0.01			●
rFiltTimeIdentMIInertiaTrq	REAL	PT1-Filterzeit für das Drehmoment der Wicklerwelle während der Identifikation des Massenträgheitsmoments • Einheit: s • Initialwert: 0.005			●
IrIdentMIInertiaMaxSpd Scaled	LREAL	Maximale Drehzahl der Wicklerwelle während der Massenträgheitsidentifikation • Einheit: x 100 % (1.0 = 100 % = IrWndSpdRef) • Wertebereich: 0 ... 1 (0 ... 100 %) • Initialwert: 0.2 (20 %)			●
IrIdentMIInertiaMaxTrq Scaled	LREAL	Maximales Drehmoment der Wicklerwelle während der Massenträgheitsidentifikation • Einheit: x 100 % (1.0 = 100 %) • Wertebereich: 0 ... 1 (0 ... 100 %) • Initialwert: 0.2 (20 %)			●
alrSpdCtrlGainAdaptX ARRAY [1...9] OF LREAL		Kennlinienfunktion für die Drehzahlregelungsverstärkung Die X-Achse entspricht dem normierten Massenträgheitsmoment. • Einheit: x 100% (1 = 100% = Parameter IrMaxMIInertia) • 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: • [0.5, 0.6, 0.7, 0.8, 0.9, 1.0, 0.98, 0.95, 0.95] • Untere Begrenzung: 0.5 = 50 % • Obere Begrenzung: 1.0 = 100 % Lineare Erhöhung der Verstärkung bis 100 % des Massenträgheitsmoments			●
IrReducedGainWindow	LREAL	Bereich der Regelabweichung mit reduzierter Verstärkung/Empfindlichkeit • Initialwert: 0.0			●
IrReducedGain	LREAL	Verstärkung der Regelabweichung im Bereich der reduzierten Empfindlichkeit • Initialwert: 0.0			●

Bezeichner	Datentyp	Beschreibung	Verfügbar in Variante								
			Base	State	High						
eDancerCtrlStopMode L_TT1P_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 (IrHaltDec) und den Ruck (IrJerk) angehalten.</td></tr><tr><td>1</td><td>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.</td></tr><tr><td>2</td><td>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.</td></tr></table>	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.			●
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.										
IrPos_Dist	LREAL	Relevant bei Modus: • eDancerCtrlStopMode = 1: Absolute Zielposition in [units] (Bezug auf die absolute Position ist die Nullposition) • eDancerCtrlStopMode = 2: Zu fahrende Wegstrecke in [units] (Bezug auf die Sollposition zum Startzeitpunkt des Kommandos.) • Initialwert: 0			●						
IrVel	LREAL	Geschwindigkeit Relevant nur für die Modi eDancerCtrlStopMode = 1 und 2 (Move ABS, Move Rel). Vorgabe, mit welcher maximalen Geschwindigkeit die Fahrt/Positionierung durchgeführt werden soll. • Einheit: units(Wickler)/s, im Standardfall rev/s • Initialwert: 50			●						
IrAcc	LREAL	Beschleunigung Relevant nur für die Modi eDancerCtrlStopMode = 1 und 2 (Move ABS, Move Rel). Vorgabe, mit welcher Geschwindigkeitsänderung maximal beschleunigt werden soll. • Einheit: units(Wickler)/s <sup>2</sup> , im Standardfall rev/s <sup>2</sup> • Initialwert: 100			●						
IrDec	LREAL	Verzögerung Relevant nur für die Modi eDancerCtrlStopMode = 1 und 2 (Move ABS, Move Rel). Vorgabe, mit welcher Geschwindigkeitsänderung maximal in den Stillstand verzögert werden soll. • Einheit: units(Wickler)/s <sup>2</sup> , im Standardfall rev/s <sup>2</sup> • Initialwert: 100			●						
IrMaxWndSpd	LREAL	Festlegung der maximalen Wicklerwellen-Drehzahl (abtriebsseitig). Aus der maximal erlaubten Drehzahl der Wicklerwelle und dem berechneten Durchmesser wird die maximal erlaubte Liniengeschwindigkeit bestimmt. Einheit: rev/s Initialwert: 100			●						
IrIdentDiamVel	LREAL	Umfangsgeschwindigkeit der Wicklerwelle für die Identifikation des Durchmessers. • Einheit: mm/s • Initialwert: 10			●						

Bezeichner	Datentyp	Beschreibung	Verfügbar in Variante		
			Base	State	High
IrIdentDiamAcc	LREAL	Umfangsbeschleunigung der Wicklerwelle für die Identifikation des Durchmessers. • Einheit: mm/s <sup>2</sup> • Initialwert: 100			●
IrIdentDiamDec	LREAL	Umfangsverzögerung der Wicklerwelle für die Identifikation des Durchmessers. • Einheit: mm/s <sup>2</sup> • Initialwert: 100			●
IrIdentDiamJerk	LREAL	Umfangsruck der Wicklerwelle für die Identifikation des Durchmessers. • Einheit: mm/s <sup>3</sup> • 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. Einstellungsbereich: -1 ... 1 Initialwert: 0.5			●
IrIdentDiamSpdCtrlGain	LREAL	Die Drehzahlregler-Verstärkung wird eingestellt, wenn während der Durchmesser-Identifikation die Drehzahlregler-Verstärkung <i>xAdaptSpdCtrlGain</i> = TRUE ist. Einstellungsbereich: 0 ... 1 Initialwert: 0.5			●
eAdaptSpdCtrlGainMode	ENUM	Modusauswahl zur Adaption der Drehzahlreglerverstärkung. • Initialwert: 2			●
		0      DiamToSquare; VP = f(d <sup>2</sup> )			
		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 <i>scPar. IrAdaptSpdCtrlLowLimit</i> . • Wertebereich: 0 bis 1 • Initialwert: 0			●

## 3.4

## State machine



[3-2] State machine des Technologiemoduls

(\*1 Im Zustand "Ready" muss xRegulatorOn auf TRUE gesetzt werden.

(\*2 Im Zustand "ERROR" muss xResetError zum Quittieren und Zurücksetzen der Fehler auf TRUE gesetzt werden.

## Zustände des Ausgangs eTMState (L\_TT1P\_States)

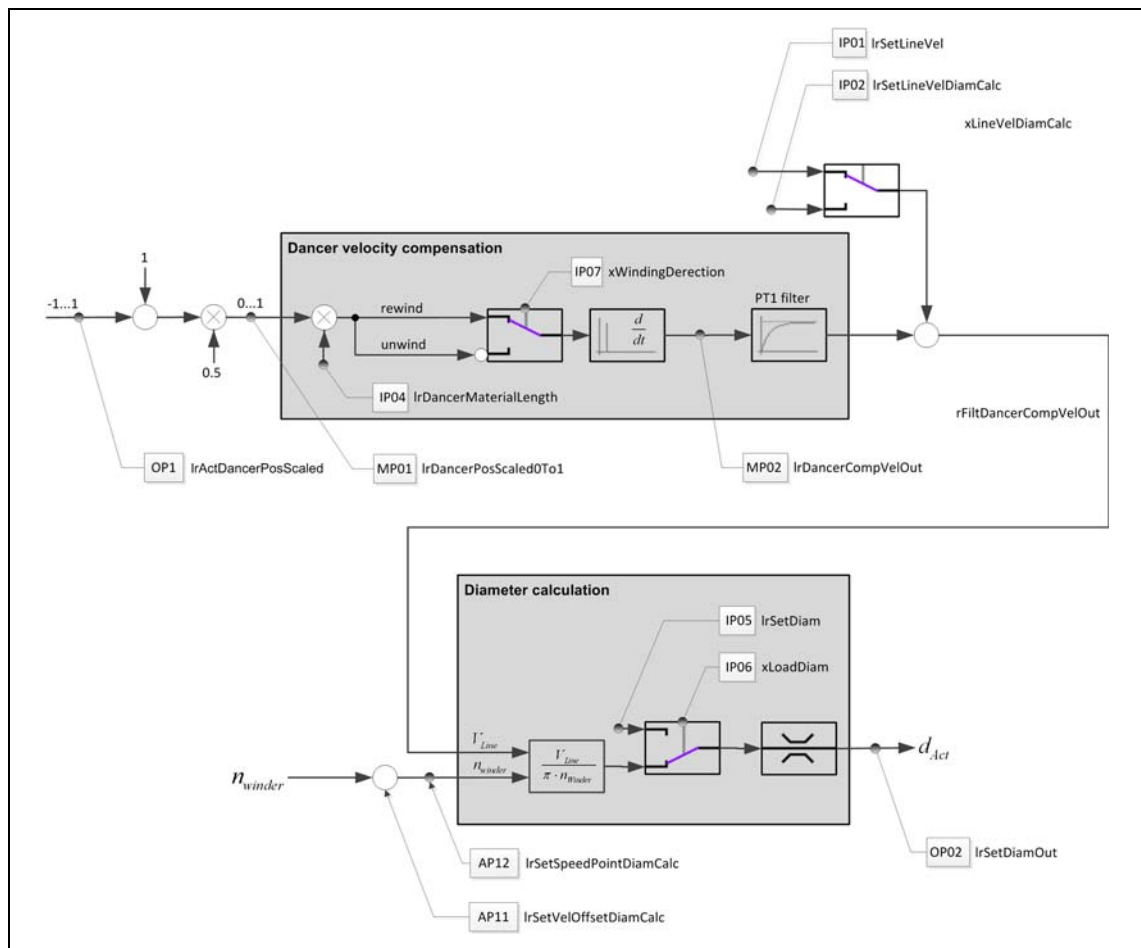
Nr.	L_TT1P_States	Beschreibung
1	INIT	Initialisierung des Technologiemoduls aktiv.
2	READY	Technologiemodul betriebsbereit.
3	HOMING	Referenzierung aktiv.
10	JOGGING	Handfahren aktiv.
11	JOGPOS	Handfahren in positive Richtung aktiv.
12	JOGNEG	Handfahren in negative Richtung aktiv.
70	SYNCLINEVEL	Synchronisation der Wicklerachse auf Linie aktiv.
80	IDENTMINERTIA	Massenträgheitsidentifikation aktiv.
81	IDENTDIAMETER	Durchmesseridentifikation aktiv.
90	IDENTFRICTION	Reibungsidentifikation aktiv.
100	DANCERCTRL	Tänzerlageregelung aktiv.
110	TENSIONCTRL	Zugkraftsteuerung/Zugkraftregelung aktiv.
121	SELECTMODECSV	Die Betriebsart wird auf CSV eingestellt.
122	SELECTMODECST	Die Betriebsart wird auf CST eingestellt.
123	SELECTMODECSP	Die Betriebsart wird auf CSP eingestellt.
996	STOP	Stop/Halt aktiv.
998	SERVICE	Das Technologiemodul befindet sich im Servicemodus. Der interne Funktionsbaustein L_MC1P_AxisBasicControl wird über die Eingangsstruktur <i>scCtrlABC</i> gesteuert. Der Status des Funktionsbausteins ist über die Ausgangsstruktur <i>scStatusABC</i> einsehbar.
999	ERROR	Fehlerzustand
1000	SYSTEMFAULT	Systemfehler

### 3.5 Signalflusspläne

In den Abbildungen [3-3] und [3-4] ist der Haupt-Signalfluss der umgesetzten Funktionen dargestellt.

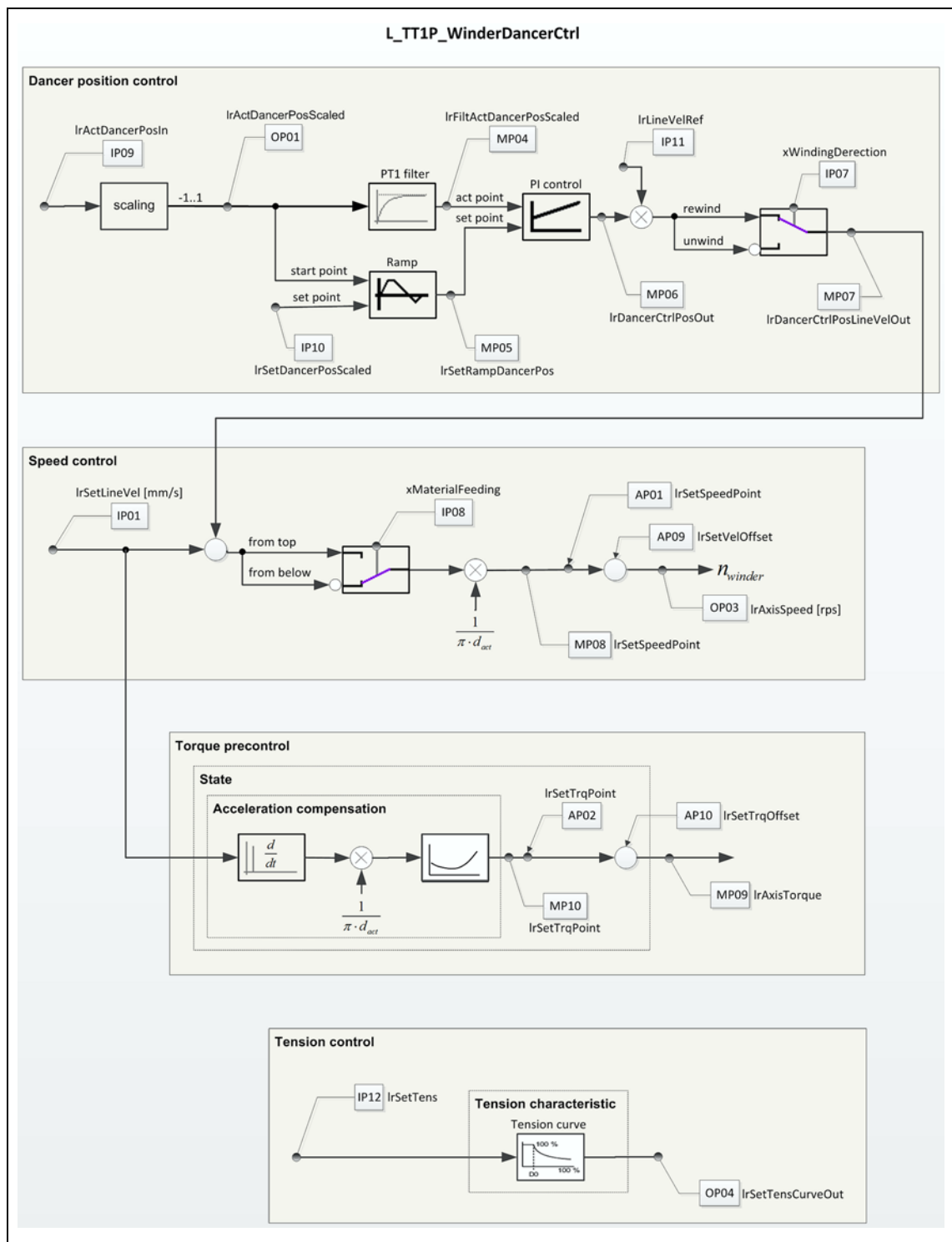
Der Signalfluss der Zusatzfunktionen, wie z. B. "Handfahren", sind hier nicht dargestellt.

#### Durchmesser-Berechnung



[3-3] Signalfluss zur Berechnung des Durchmessers





[3-4] Signalfluss des Technologiemoduls

### 3.5.1 Struktur des Signalflusses

#### L\_TT1P\_scSF\_WinderDancerCtrl [Base/State/High]

Die Inhalte der Struktur **L\_TT1P\_scSF\_WinderDancerCtrl**[Base/State/High] sind nur lesbar und bieten eine praktische Diagnosemöglichkeit innerhalb des Signalflusses ([Signalflusspläne \(32\)](#)).

Bezeichner	Datentyp	Beschreibung	Verfügbar in Variante		
			Base	State	High
IP01_IrSetLineVel	LREAL	Aktuelle Liniengeschwindigkeit • Einheit: mm/s	●	●	●
IP02_IrSetLineVelDiamCalc	LREAL	Aktuelle Liniengeschwindigkeit für die Durchmesserberechnung • Einheit: mm/s	●	●	●
IP03_xLineVelDiamCalc	BOOL	Quelle der Liniengeschwindigkeit für die Durchmesserberechnung • Initialwert: FALSE	●	●	●
		TRUE Für die Berechnung des Durchmessers wird die Geschwindigkeit aus dem Eingang IrSetLineVelDiamCalc verwendet.			
		FALSE Für die Berechnung des Durchmessers wird die Geschwindigkeit aus dem Eingang IrSetLineVel verwendet.			
IP04_IrDancerMaterialLength	LREAL	Länge des Materials im Tänzer • Einheit: mm • Initialwert: 1 • Mit dem Wert '0' wird die Betrachtung der Tänzerbewegung deaktiviert.	●	●	●
IP05_IrSetDiam	LREAL	Vorgabe eines (Start-)Durchmessers Der Durchmesser wird zyklisch geladen wenn der Eingang xLoadDiam = TRUE gesetzt ist. • Einheit: mm	●	●	●
IP06_xLoadDiam	BOOL	TRUE Der Durchmessers wird aus dem Signal IrSetDiam geladen.	●	●	●
IP07_xWindingDirection	BOOL	Funktion des Wicklers bei positiver Liniengeschwindigkeit (Eingang IrSetLineVel > 0)	●	●	●
		TRUE Abwickler			
		FALSE Aufwickler			
IP08_xMaterialFeeding	BOOL	Materialführung von oben oder unten an den Wickelballen	●	●	●
		TRUE Materialführung von oben			
		FALSE Materialführung von unten			
IP09_IrActDancerPosIn	LREAL	Aktuelle Tänzerposition Die Istposition des Tänzers wird in Form eines analogen Signals (0 ... 10 V) an den Controller zurückgeführt. • Einheit: s	●	●	●
IP10_IrSetDancerPosScaled	LREAL	Skalierter Sollwert für die Tänzerposition • Einheit: x 100 % • Wertebereich: -1 ... 1 (-100 ... 100 %)	●	●	●
IP11_IrLineVelRef	LREAL	Maximale Liniengeschwindigkeit • Einheit: mm/s • Initialwert: 1000	●	●	●
IP12_IrSetTens	LREAL	Zugkraftsollwert • Einheit: N	●	●	●

Bezeichner Datentyp	Beschreibung	Verfügbar in Variante		
		Base	State	High
MP01_IrDancerPosScaled 0To1 LREAL	Skalierte Tänzerposition • Einheit: x 100 % • Wertebereich: 0 ... 1 (0 ... 100 %)	●	●	●
MP02_IrDancerCompVelOut LREAL	Resultierende Geschwindigkeit des Tänzers • Einheit: mm/s	●	●	●
MP03_rFiltDancerCompVel Out REAL	Gefilterte resultierende Geschwindigkeit des Tänzers • Einheit: mm/s	●	●	●
MP04_rFiltActDancerPos Scaled REAL	Aktuelle skalierte gefilterte Tänzerposition • Einheit: x 100 % • Wertebereich: -1 ... 1 (-100 ... 100 %)	●	●	●
MP05_IrSetRampDancerPos LREAL	Sollposition des Tänzers nach dem Rampengenerator Diese Position ist die Führungsgröße des Wickelantriebs. • Einheit: x 100 % • Wertebereich: -1 ... 1 (-100 ... 100 %)	●	●	●
MP06_IrDancerCtrlPosOut LREAL	Stellgröße des Tänzerlagereglers • Einheit: x 100 % • Wertebereich: -1 ... 1 (-100 ... 100 %)	●	●	●
MP07_IrDancerCtrlLineVel Out LREAL	Stellgröße des Tänzerlagereglers umgerechnet in die Liniengeschwindigkeit • Einheit: mm/s	●	●	●
MP08_IrSetSpeedPoint LREAL	Resultierender Drehzahlsollwert • Einheit: revs/s	●	●	●
MP09_IrAxisTorque LREAL	Drehmoment für die Vorsteuerung • Einheit: Nm		●	●
MP10_IrSetTrqPoint LREAL	Resultierender Drehmomentsollwert für die Vorsteuerung • Einheit: Nm		●	●
MP11_IrDancerCtrlOutGain LREAL	Die Stellgröße aus dem proportionalen Anteil (P-Anteil) des Tänzerlagereglers (skaliert)	●	●	●
MP12_IrDancerCtrlOutReset Time LREAL	Die Stellgröße aus dem integrierenden Anteil (I-Anteil) des Tänzerlagereglers (skaliert)	●	●	●
MP13_IrDancerCtrlOutRate Time LREAL	Die Stellgröße aus dem differenzierenden Anteil (D-Anteil) des Tänzerlagereglers (skaliert)	●	●	●
OP01_IrActDancerPosScaled LREAL	Aktuelle skalierte Tänzerposition • Einheit: x 100 % • Wertebereich: -1 ... 1 (-100 ... 100 %)	●	●	●
OP02_IrSetDiamOut LREAL	Aktueller berechneter Durchmesser • Einheit: mm	●	●	●
OP03_IrAxisSpeed LREAL	Aktueller Drehzahl des Wickelantriebs • Einheit: revs/s	●	●	●
OP04_IrSetTensCurveOut LREAL	Aktuelle Zugkraft aus der Kurvenfunktion • Einheit: N	●	●	●

### 3.5.2 Struktur der Angriffspunkte

#### L\_TT1P\_scAP\_WinderDancerCtrl [Base/State/High]

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		Verfügbar in Variante		
				Base	State	High
AP01_xSetSpeedPoint	BOOL	Freigabe des Angriffspunktes AP01_lrSetSpeedPoint		●	●	●
		TRUE	Der Angriffspunkt überschreibt die Werte an der Zugriffstelle im Signalfluss.			
AP01_lrSetSpeedPoint	LREAL	Drehzahlsollwert für den Wickelantrieb • Einheit: revs/s				
AP02_xSetTrqPoint	BOOL	Freigabe des Angriffspunktes AP02_lrSetTrqPoint			●	●
		TRUE	Der Angriffspunkt überschreibt die Werte an der Zugriffstelle im Signalfluss.			
AP02_lrSetTrqPoint	LREAL	Drehmomentsollwert für die Vorsteuerung • Einheit: Nm				
AP03_xSetDancerCtrlOutGain	BOOL	Freigabe des Angriffspunktes AP03_lrSetDancerCtrlOutGain			●	●
		TRUE	Der Angriffspunkt überschreibt die Werte an der Zugriffstelle im Signalfluss.			
AP03_lrSetDancerCtrlOutGain	LREAL	Zyklisches Laden der Stellgröße des proportionalen Anteils (P-Anteil) des Tänzerlagereglers (skaliert)				
AP04_xSetDancerCtrlOutResetTime	BOOL	Freigabe des Angriffspunktes AP04_lrSetDancerCtrlOutResetTime			●	●
		TRUE	Der Angriffspunkt überschreibt die Werte an der Zugriffstelle im Signalfluss.			
AP04_lrSetDancerCtrlOutResetTime	LREAL	Zyklisches Laden der Stellgröße des integrierenden Anteils (I-Anteil) des Tänzerlagereglers (skaliert)				
AP05_xSetDancerCtrlOutRateTime	BOOL	Freigabe des Angriffspunktes AP05_lrSetDancerCtrlOutRateTime			●	●
		TRUE	Der Angriffspunkt überschreibt die Werte an der Zugriffstelle im Signalfluss.			
AP05_lrSetDancerCtrlOutRateTime	LREAL	Zyklisches Laden der Stellgröße des differenzierenden Anteils (D-Anteil) des Tänzerlagereglers (skaliert)				
AP09_xSetVelOffset	BOOL	Freigabe des Angriffspunktes AP09_lrSetVelOffset		●	●	●
		TRUE	Der Angriffspunkt überschreibt die Werte an der Zugriffstelle im Signalfluss.			
AP09_lrSetVelOffset	LREAL	Zyklische Vorgabe des Offset für die Geschwindigkeit der Wicklerachse bezogen auf die Wickelwelle (Getriebeausgangsseite) • Einheit: units/s <b>Der Offset-Wert wird ohne Rampengenerator sofort und sprunghaft eingestellt!</b>				

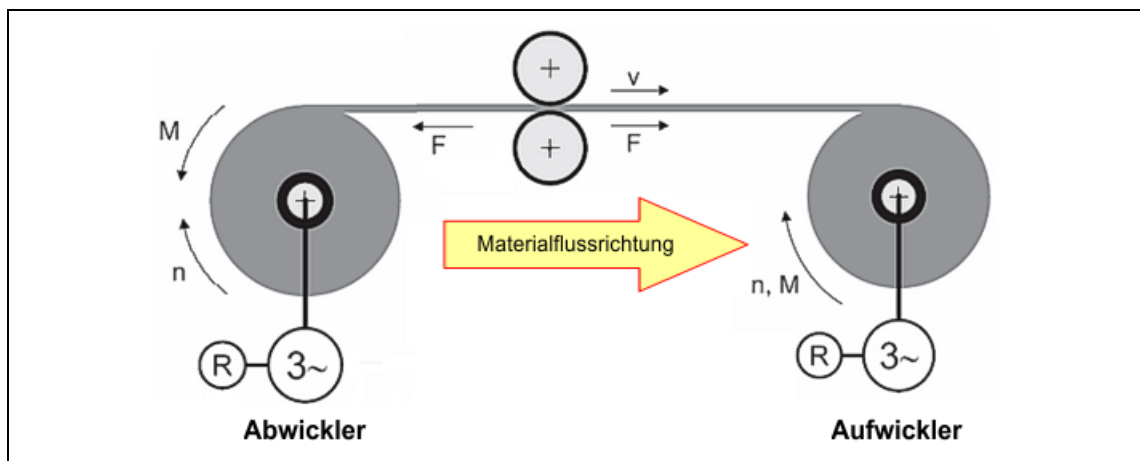
Bezeichner	Datentyp	Beschreibung		Verfügbar in Variante		
				Base	State	High
AP10_xSetTrqOffset	BOOL	Freigabe des Angriffspunktes AP10_IrSetTrqOffset		●	●	●
		TRUE	Der Angriffspunkt überschreibt die Werte an der Zugriffstelle im Signalfluss.			
AP10_IrSetTrqOffset	LREAL	Zyklische Vorgabe des Offset für das Drehmoment der Wicklerachse bezogen auf die Wickelwelle (Getriebeausgangsseite) <ul style="list-style-type: none"> <li>• Einheit: Nm</li> </ul> <b>Der Offset-Wert wird ohne Rampengenerator sofort und sprungartig eingestellt!</b>				
AP11_xSetVelOffsetDiamCalc	BOOL	Freigabe des Angriffspunktes AP11_IrSetVelOffsetDiamCalc:		●	●	●
		TRUE	Der Angriffspunkt überschreibt die Werte an der Zugriffstelle im Signalfluss			
AP11_IrSetVelOffsetDiamCalc	LREAL	Zyklische Vorgabe des Offsets für die Geschwindigkeit der Wicklerachse, die für die Berechnung des Durchmessers verwendet wird. Dieser Angriffspunkt wirkt sich nicht auf den Sollwert der Wicklerachse aus, sondern geht lediglich in die Berechnung des Durchmessers ein.				
AP12_xSetSpeedPointDiamCalc	BOOL	Freigabe des Angriffspunktes AP12_IrSetSpeedPointDiamCalc:		●	●	●
		TRUE	Der Angriffspunkt überschreibt die Werte an der Zugriffstelle im Signalfluss.			
AP12_IrSetSpeedPointDiamCalc	LREAL	Zyklische Vorgabe der Drehzahl der Wicklerachse, die für die Berechnung des Durchmessers verwendet wird. Dieser Angriffspunkt wirkt sich nicht auf den Sollwert der Wicklerachse aus, sondern geht lediglich in die Berechnung des Durchmessers ein.				

### 3.6 Festlegung der Wickelrichtung (Aufwickeln/Abwickeln)

Damit die Vorsteuergrößen, die Störgrößenkompensation und auch 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.)



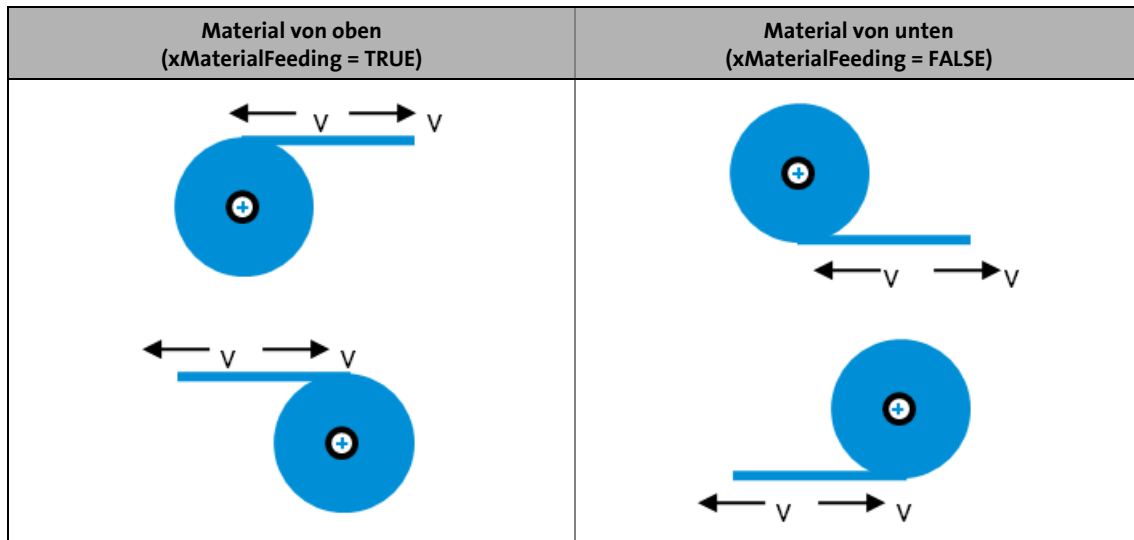
[3-5] Wirkrichtung von Drehzahl und Drehmoment in Abhängigkeit des Materialflusses

### 3.7 Automatische Erkennung der Wickelrichtung

Nach [Festlegung der Wickelrichtung \(Aufwickeln/Abwickeln\)](#) (38) 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.

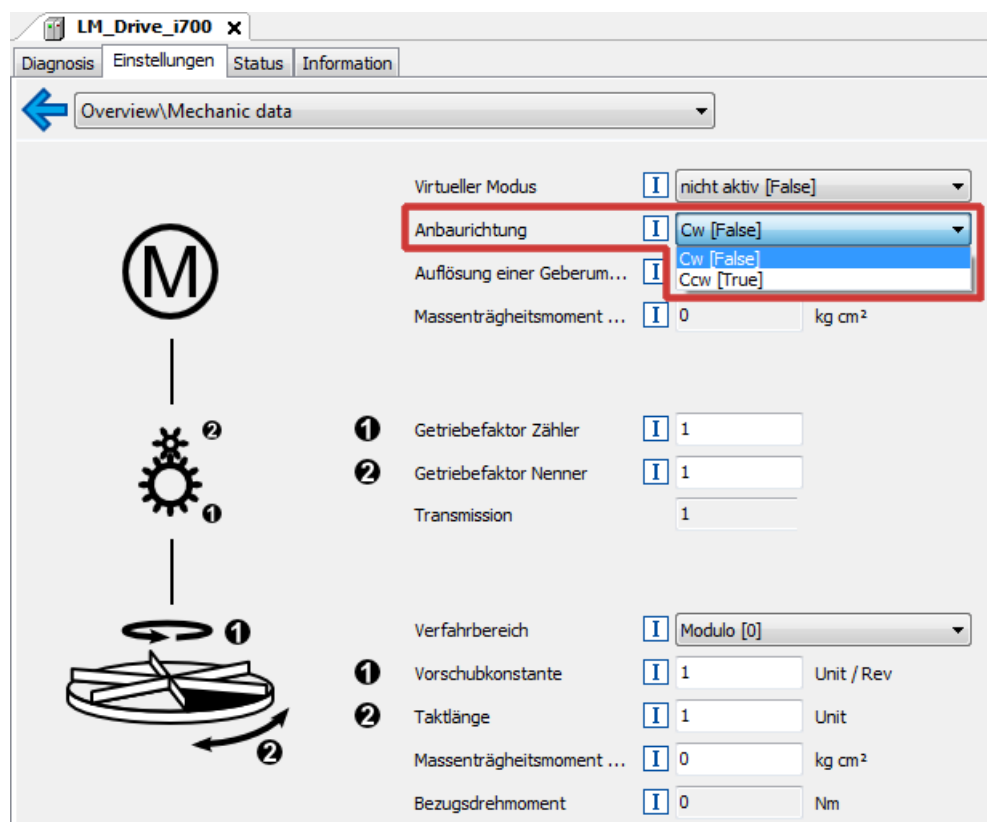
### 3.8 Festlegung der Materialzuführung an den Wickler

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



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

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



---

### 3.9 Leitwert-Quelle für die Durchmesserberechnung

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

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 *lrSetLineVelDiamCalc* verschaltet werden und der Parameter *xLineVelDiamCalc* = TRUE gesetzt werden. Dadurch ist die Berücksichtigung der Tänzerbewegung nicht mehr notwendig – der Parameter *lrDancerMaterialLength* = 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 *lrSetLineVel* verwendet. Um die Berechnung des Durchmessers zu optimieren, kann die Tänzerbewegung berücksichtigt werden (siehe dazu [Durchmesserberechnung mit Korrektur der Tänzerposition](#) (📖 45)).

#### Einzustellende Parameter

Die Parameter für die Durchmesserberechnung mit oder ohne Encoder befinden sich in der Parameterstruktur [L TT1P\\_scPar\\_WinderDancerCtrl \[Base/State/High\]](#) (📖 23).

```
xLineVelDiamCalc : BOOL := FALSE;  
lrDancerMaterialLength : LREAL := 0;
```



### 3.10 Drehzahlvorsteuerung

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

Berechnung der Solldrehzahl für die Drehzahlvorsteuerung		
$nSet = \frac{vLine}{d_{act} \cdot \pi}$		
Formelzeichen	Beschreibung	Maßeinheit
nSet	Solldrehzahl für die Drehzahlvorsteuerung	revs/s
vLine	Liniengeschwindigkeit am Eingang <i>IrSetLineVel</i>	mm/s
dact	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.

#### Drehzahlvorsteuerung prüfen

- Laden Sie den Durchmesser-Rechner mit dem minimalen Durchmesser ( $d_{min}$ ):  
Eingang *IrSetDiam* = 0 (oder  $\leq d_{min}$ )  
Eingang *xLoadDiam* = TRUE
- Bei der [Synchronisierung auf die Liniengeschwindigkeit](#) (52) mit dem Eingang *xSynLineVel* = TRUE folgt die Wicklerachse dem Liniengeschwindigkeitssollwert rein drehzahl geregelt, ohne dass die Tänzerlage korrigiert wird.  
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.

### 3.11 Durchmesserberechnung

Der aktuelle Durchmesser wird durch Division der Liniengeschwindigkeit mit der Wicklerdrehzahl und der Zahl  $\pi$  berechnet:

Berechnung des aktuellen Durchmessers		
$d_{act} = \frac{v_{Line}}{n_{Winder} \cdot \pi}$		
Formelzeichen	Beschreibung	Maßeinheit
$d_{act}$	Aktueller Durchmesser	mm
$v_{Line}$	Liniengeschwindigkeit	mm/s
$n_{Wickler}$	Wicklerdrehzahl	revs/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 *lrDiamCalcRegularDist* bestimmt. Der Initialwert dieses Parameters ist auf 1 Wicklerwellenumdrehung eingestellt.

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

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 Wicklerwelle nur um eine kurze Distanz mit "falschem" Durchmesser. Nach der Durchmesserberechnung ist wieder ein passender Wert vorhanden.

#### Einzustellende Parameter

Die Parameter für die Durchmesserberechnung befinden sich in der Parameterstruktur [L TT1P\\_scPar\\_WinderDancerCtrl \[Base/State/High\]](#) (23).

```
lrDiamCalcRegularDist : LREAL := 1;
lrDiamCalcReducedDist : LREAL := 0.1;
```

### 3.12 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  
(*lrMinLineVel* [mm/s] aus der Parameterstruktur [L TT1P\\_scPar\\_WinderDancerCtrl \[Base/State/High\]](#) (23));
- Wicklerdrehzahl <  $lrMinLineVel \text{ [mm/s]} / (\pi \times d \text{ [mm]})$ ;
- In den Zuständen STOP, ERROR, READY, JOGGING und SYNCLINEVEL.

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

### 3.13 Durchmesser vorgeben / Signal vom Durchmessersensor

Zu Beginn eines Wickelvorgangs kann es erforderlich sein, einen Startdurchmesser vorzugeben oder das Signal eines Durchmessersensors zu verwenden.

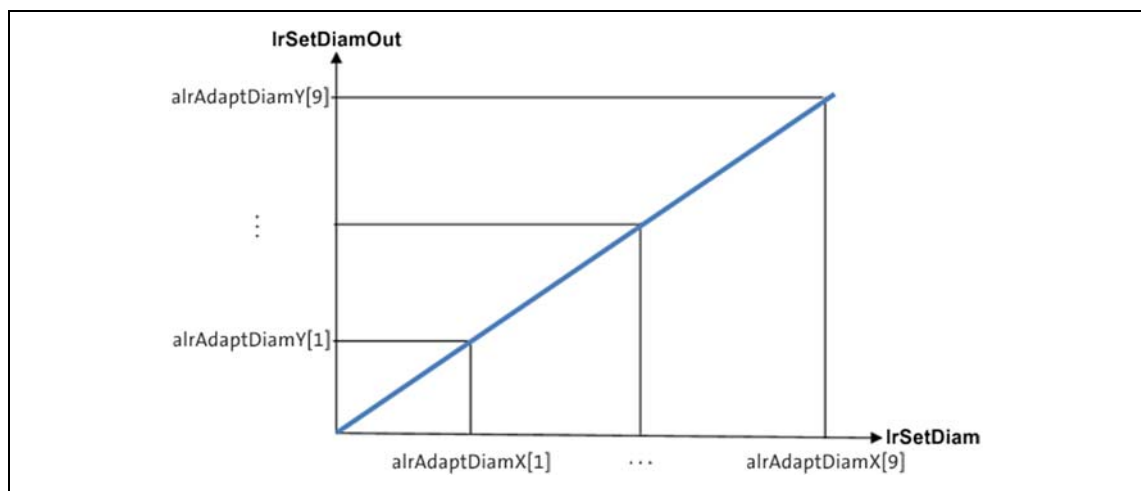
Mit dem Eingang *IrSetDiam* können Sie einen Startdurchmesser festlegen, der mit *xLoadDiam* = TRUE mit höchster Priorität übernommen und zyklisch geladen wird.

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.

#### Einzustellende Parameter

Die Parameter für die Kurvenfunktion befinden sich in der Parameterstruktur [L TT1P\\_scPar\\_WinderDancerCtrl \[Base/State/High\]](#) (23).

```
alrAdaptDiamX : ARRAY[1...9] OF LREAL := [0,100,200,300,400,500,600,700,800];
alrAdaptDiamY : ARRAY[1...9] OF LREAL := [0,100,200,300,400,500,600,700,800]
```

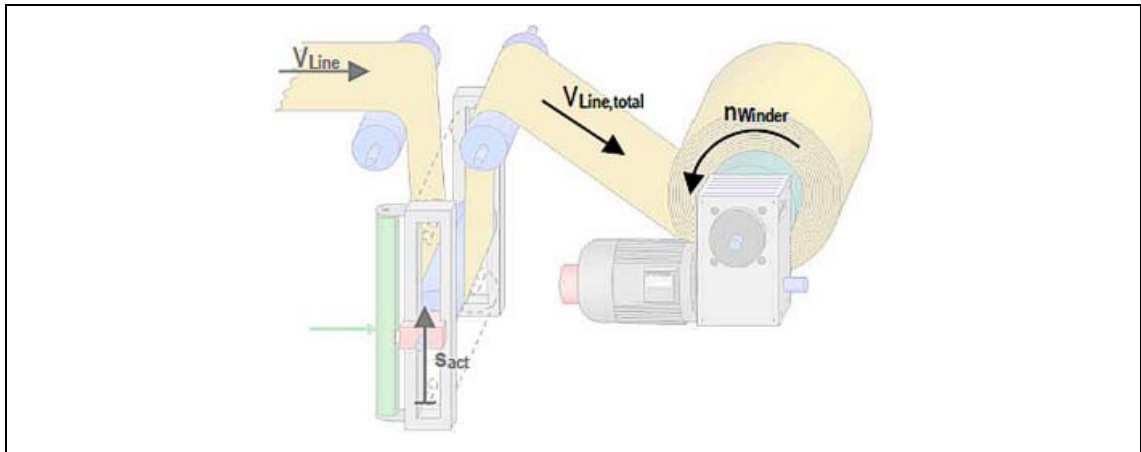


[3-6] Laden eines Durchmessers über eine Kurvenfunktion

### 3.14 Durchmesserberechnung mit Korrektur der Tänzerposition

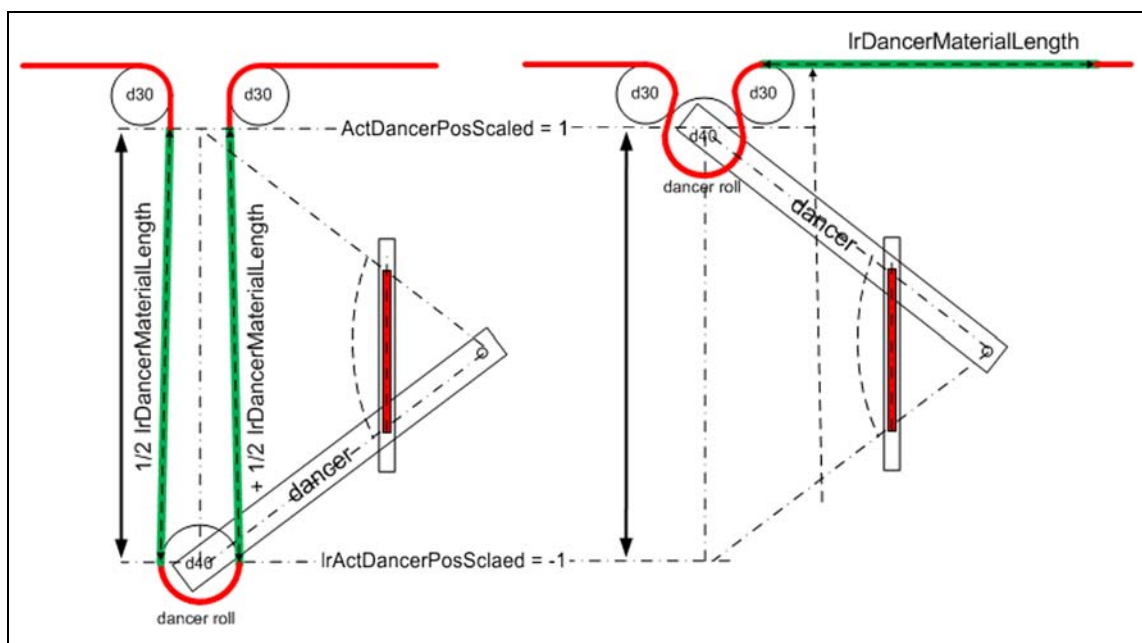
Bewegt sich der Wickler am Umfang schneller oder langsamer als die Linie, muss die Tänzerposition korrigiert werden. Wenn hierbei die Umfangsgeschwindigkeit gegenüber der Liniengeschwindigkeit vor dem Tänzer deutlich zu oder abnimmt, muss die resultierende Umfangsgeschwindigkeit ( $V_{Line, total}$ ) für die Durchmesserberechnung verwendet werden.

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



[3-7] 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 %.



[3-8] Beispiel: Die gespeicherte Materiallänge entspricht dem doppeltem Tänzerweg

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 Zeitkonstante *rFiltTimeActDancerPosIn* gefiltert. Die gefilterte Position wird von -1 bis 1 skaliert und am Ausgang *lrActDancerPosScaled* ausgegeben.

Die Konvertierung der skalierten Tänzerposition zu einer Materiallänge in Millimeter erfolgt mit dem Parameter *lrDancerMaterialLength* durch die Formel:

Gleichung für die Konvertierung der skalierten Tänzerposition zu einer Materiallänge [mm]		
$\text{DancerPos} = \left( \frac{\text{lrActDancerPosScaled} + 1}{2} \right) \cdot \text{lrDancerMaterialLength}$		
Formelzeichen	Beschreibung	Maßeinheit
DancerPos	Tänzerposition	
lrActDancerPosScaled	Aktuelle skalierte Tänzerposition • Wertebereich: -1 ... 1 (-100 ... 100 %)	%
lrDancerMaterialLength	Länge des Materials im Tänzer	mm

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

#### Zusätzliche Geschwindigkeit bei der Durchmesserberechnung

Wenn die folgenden Einstellungen erfolgt sind, wird jede Tänzerbewegung bei der Durchmesserberechnung berücksichtigt:

- Parametrierung der maximalen Materiallänge, die in der Tänzermechanik gespeichert werden kann: *lrDancerMaterialLength* = maximale Materiallänge [mm]
- Mit der Voreinstellung *lrDancerMaterialLength* = 0 [mm] wird keine Zusatzgeschwindigkeit aus der Tänzerbewegung für die Durchmesserberechnung berücksichtigt.

#### Einzustellende Parameter

Die Parameter für die Durchmesserberechnung befinden sich in der Parameterstruktur [L TT1P\\_scPar\\_WinderDancerCtrl \[Base/State/High\]](#) (23).

```
lrDancerMaterialLength : LREAL := 0;
rFiltTimeActDancerPosIn : REAL := 0.005;
```

### 3.15 Materiallängenzähler

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

Die Materiallänge wird durch Integration die Liniengeschwindigkeit am Eingang *lrSetLineVel* berechnet und am Ausgang *lrMaterialCounter* (in Millimeter) angezeigt. Je nach [Festlegung der Wickelrichtung \(Aufwickeln/Abwickeln\)](#) (☞ 38) wird die Materiallänge hoch- oder runtergezählt.

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 (die Voreinstellung ist '0 ms').

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

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

#### Einzustellende Parameter

Die Parameter für den Materiallängenzähler befinden sich in der Parameterstruktur [L\\_TT1P\\_scPar\\_WinderDancerCtrl \[Base/State/High\]](#) (☞ 23).

```
rFiltTimeMaterialCounter : LREAL := 0;  
lrSetMaterialPos : REAL := 0;
```

### 3.16 Quellen für die Materiallängenzählung

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

#### 3.16.1 Quelle: Eingang „IrSetLineVel“

##### Voraussetzungen

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

##### Funktionsweise

In die Materiallängenzählung wird zusätzlich zur Position (Parameter *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.



##### Hinweis!

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.

#### 3.16.2 Quelle: Eingang „IrSetLineVelDiamCalc“

##### Voraussetzungen

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

##### Funktionsweise

In die Materiallängenzählung wird zusätzlich zur Position (Parameter *IrSetMaterialPos*) die Liniengeschwindigkeit für die [Durchmesserberechnung](#) (42) am Eingang *IrSetLineVelDiamCalc* 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.



##### Hinweis!

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.



### 3.16.3 Quelle: Eingang „MaterialCounterAxis“ (Referenzachse)

#### Voraussetzungen

- Eine Referenzachse (Modulo-Achse) ist am Eingang *MaterialCounterAxis* angeschlossen.
- Als Basis wird die Ermittlung der verlustfreien Anzahl der Umdrehungen für die Materiallängenzählung verwendet. – Dieses Verfahren eignet sich für verauschte Signale!

#### Funktionsweise

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.



#### Hinweis!

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

### 3.17 Handfahren (Jogging)

Zum Handfahren des Wicklers wird die Handfahr-Geschwindigkeit *lrJogLineVel* verwendet.

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 *lrJogLineVel*, *lrJogLineAcc* und *lrJogLineDec* für das Handfahren beziehen sich auf die Umfangsgeschwindigkeit oder Liniengeschwindigkeit und nicht auf die Motordrehzahl.

#### Einzustellende Parameter

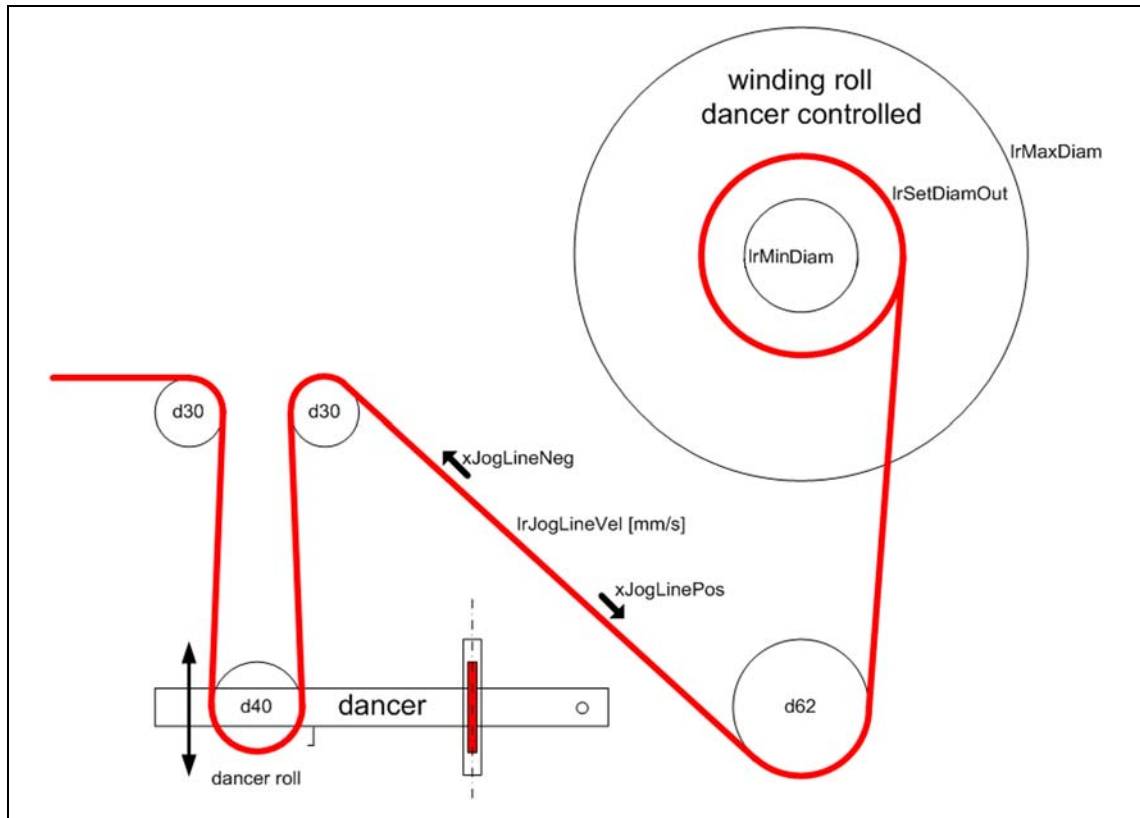
Die Parameter für das Handfahren befinden sich in der Parameterstruktur [L TT1P\\_scPar\\_WinderDancerCtrl \[Base/State/High\]](#) (23).

```
lrLineJerk    : LREAL := 10000; // Jerk [mm/s^3]
lrJogLineVel  : LREAL := 100;   // Velocity [mm/s]
lrJogLineAcc  : LREAL := 100;   // Acceleration [mm/s^2]
lrJogLineDec  : LREAL := 10;    // Deceleration [mm/s^2]
```

Die Parameterwerte können während des Betriebes verändert werden. Sie werden bei erneutem Setzen der Eingänge *xJogLinePos* = TRUE oder *xJogLineNeg* = TRUE übernommen.

**Beispiel**

- *xWindingDirection* = FALSE: Aufwickler (Das Material wird aufgewickelt.)
- *xMaterialFeeding* = FALSE: Das Material wird von unten geführt.



[3-9] Handfahren der Linie

**Hinweis!**

Während des Handfahrens wird der Durchmesserrechner angehalten und der Durchmesser wird gehalten (*xHoldDiamActive* = TRUE).

---

### 3.18 Synchronisierung auf die Liniengeschwindigkeit

Die Synchronisierung der Wicklerachse auf die Liniengeschwindigkeit wird mit dem Eingang *xSyncLineVel* = TRUE ausgeführt.

Die Parameter *lrSyncLineAcc* und *lrSyncLineDec* beziehen sich auf die Umfangsgeschwindigkeit oder Liniengeschwindigkeit und nicht auf die Motordrehzahl.

#### Einzustellende Parameter

Die Parameter für die Synchronisierung auf die Liniengeschwindigkeit befinden sich in der Parameterstruktur [L\\_TT1P\\_scPar\\_WinderDancerCtrl \[Base/State/High\]](#) (23).

```
lrLineJerk      : LREAL := 10000; // Jerk [mm/s^3]
lrSyncLineAcc   : LREAL := 100;  // Acceleration [mm/s^2]
lrSyncLineDec   : LREAL := 100;  // Deceleration [mm/s^2]
```



#### Hinweis!

Während des Handfahrens wird der Durchmesserrechner angehalten und der Durchmesser wird gehalten (*xHoldDiamActive* = TRUE).

## 3.19

## Trimmung

**Hinweis!**

Die Trimmung kann nur verwendet werden, wenn die Wicklerachse auf die Liniengeschwindigkeit synchronisiert ist.

► [Synchronisierung auf die Liniengeschwindigkeit](#) (📖 52)

Mit dem Eingang *xTrimLinePos* = TRUE wird die Linie in positive Richtung und mit dem Eingang *xTrimLineNeg* = TRUE in negative Richtung vertrimmt.

Für die Trimmung wird die Trimm-Geschwindigkeit *lrTrimLineVel* zur Liniengeschwindigkeit *lrSetLineVel* addiert. Bei der Trimmung kann der Gesamtsollwert maximal um den Wert der minimalen Liniengeschwindigkeit größer sein als der Trimm-Sollwert.

Die parametrierbaren Sollwerte *lrTrimLineVel*, *lrTrimLineAcc* und *lrTrimLineDec* für den positiven und negativen Trimm-Betrieb beziehen sich auf die Umfangsgeschwindigkeit oder Liniengeschwindigkeit und nicht auf die Motordrehzahl.

**Einzustellende Parameter**

Die Parameter für die Trimmung befinden sich in der Parameterstruktur [L TT1P\\_scPar\\_WinderDancerCtrl \[Base/State/High\]](#) (📖 23).

```
lrLineJerk      : LREAL := 10000; // Jerk [mm/s^3]
lrTrimLineVel   : LREAL := 100;  // Velocity [mm/s]
lrTrimLineAcc   : LREAL := 100;  // Acceleration [mm/s^2]
lrTrimLineDec   : LREAL := 10;   // Deceleration [mm/s^2]
```

### 3.20 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 *lrActDancerPosIn* anliegen.

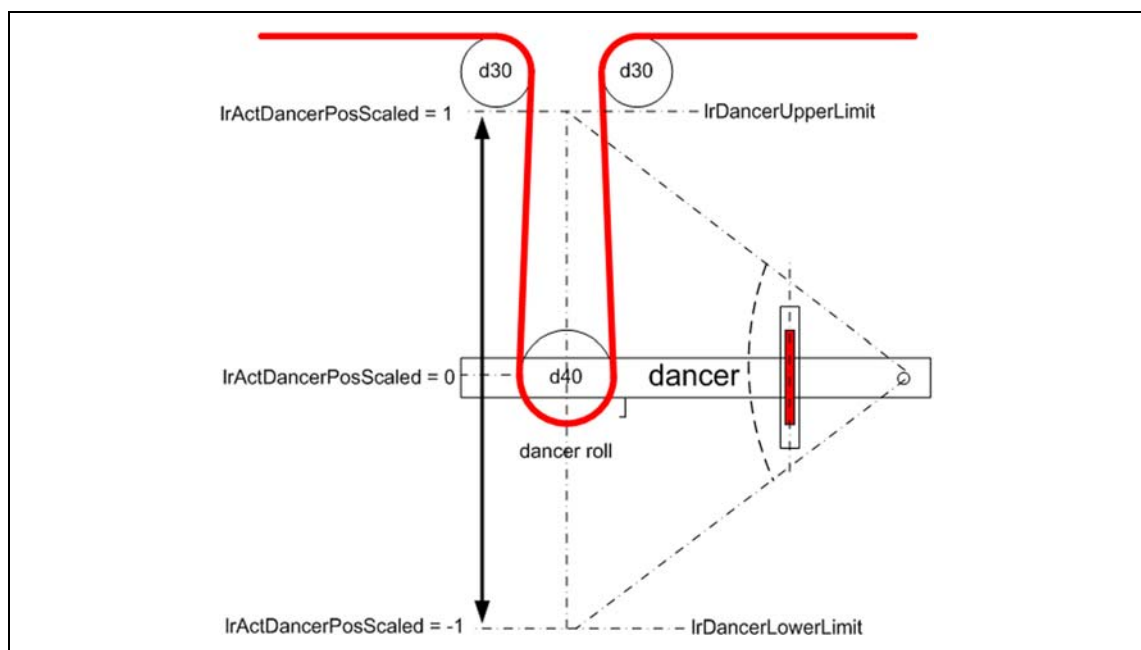
Die Tänzerposition wird über einen PT1-Filter mit der Zeitkonstante *rFiltTimeActDancerPosIn* gefiltert. Die gefilterte Position wird von -1 bis 1 skaliert und am Ausgang *lrActDancerPosScaled* ausgegeben. Dadurch wird die Vorgabe des Sollwertes und die Überwachung der Tänzerposition vereinfacht.

Die Grenzwerte für die obere und untere Tänzerposition (Tänzerendlagen) werden über die Parameter *lrDancerUpperLimit* und *lrDancerLowerLimit* vorgegeben. Die Quelle für die Tänzerlagebegrenzen wird durch den Parameter *xTeachDancerLimits* bestimmt.

#### Einzustellende Parameter

Der Parameter für die Normierung der Tänzerlage befinden sich in der Parameterstruktur [L TT1P\\_scPar\\_WinderDancerCtrl \[Base/State/High\]](#) (23).

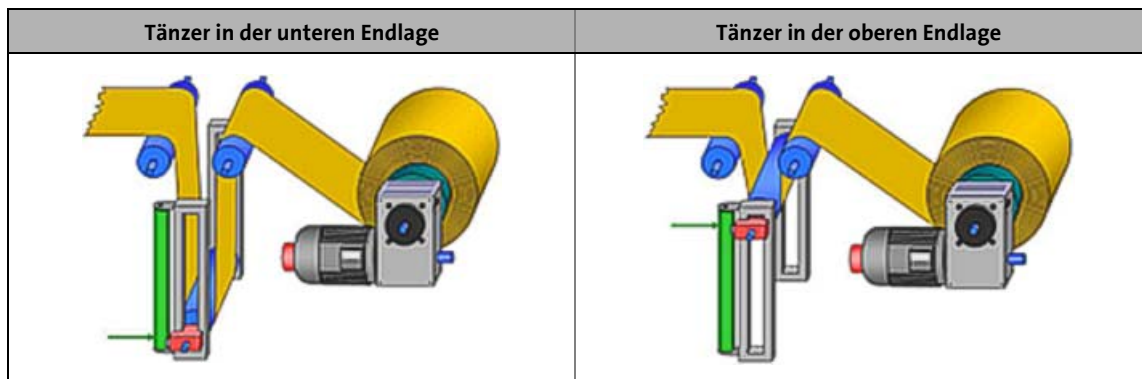
```
rFiltTimeActDancerPosIn : REAL := 0.005; // Filter time for lrActDancerPosIn [s]
lrDancerUpperLimit : LREAL := 10000000; // Mechanical upper limit dancer position
lrDancerLowerLimit : LREAL := 1; // Mechanical lower limit dancer position
xTeachDancerLimits : BOOL := FALSE; // Source for dancer position limiting
```



[3-10] Normierung der Tänzerlage

## 3.21

## Teaching-Funktion für Tänzerendlagen



So führen Sie das Teaching manuell aus:

1. Den Tänzer manuell in die untere Endlage bewegen, so dass maximal Material im Tänzer vorhanden ist.
2. Den Eingang `xTeachLowerPos` = TRUE setzen.  
Der aktuelle Eingangswert `lrActDancerPosIn` wird gespeichert und in den Retain/Persistent-Speicher, falls dieser über den Eingang `PersistentVar` verschaltet ist, geschrieben.
3. Tänzer manuell in die obere Endlage bewegen, so dass minimal Material im Tänzer gespeichert ist.
4. Den Eingang `xTeachUpperPos` = TRUE setzen.  
Der aktuelle Eingangswert `lrActDancerPosIn` wird gespeichert und in den Retain/Persistent-Speicher, falls dieser über den Eingang `PersistentVar` verschaltet ist, geschrieben.

Alternativ zum Teaching können Sie die jeweiligen Eingangswerte manuell in die Parameter `lrDancerUpperLimit` und `lrDancerLowerLimit` eintragen.

Mit dem Parameter `xTeachDancerLimits` wird die Quelle der Tänzerlagebegrenzungen ausgewählt:

Parameterwert	Beschreibung
TRUE	Die Endlagen <code>lrDancerLowerLimit</code> und <code>lrDancerUpperLimit</code> 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 <code>lrDancerLowerLimit</code> und <code>lrDancerUpperLimit</code> werden verwendet.

#### Einzustellende Parameter

Die Parameter für die Teaching-Funktion befinden sich in der Parameterstruktur [L TT1P\\_scPar\\_WinderDancerCtrl \[Base/State/High\]](#) (23).

```
lrDancerUpperLimit : LREAL := 10000000; // Mechanical upper limit dancer position
lrDancerLowerLimit : LREAL := 0;        // Mechanical lower limit dancer position
xTeachDancerLimits : BOOL := FALSE;     // Source for dancer position limiting
```

### 3.22 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 Bahnrisser 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

Die Parameter für die Überwachung der Tänzerposition befinden sich in der Parameterstruktur [L\\_TT1P\\_scPar\\_WinderDancerCtrl \[Base/State/High\]](#) (23).

```
lrDancerMaxPosScaled : LREAL := 0.95; //Upper limit of dancer position 0..1 [x100%]  
lrDancerMinPosScaled : LREAL := -0.95; //Lower limit of dancer position 0..-1 [x100%]  
lrDancerInPosWindowScaled : LREAL := 0.2; //Window of dancer in position [x100%]
```



### 3.23 PI-Regler für die Tänzerlageregelung

Durch Aktivierung der Tänzerlageregelung mit dem Eingang *xDancerCtrl* = TRUE wird auf einen Tänzerpositions-Istwert geregelt.

Mit dem Eingang *lrDancerPosInfluenceScaled* legen Sie fest, welchen Einfluss der PI-Regler auf die Steuerung des Motors haben soll.

Das Signal der aktuellen Tänzerposition kann mit einer PT1-Charakteristik gefiltert werden. Die Filterzeit ist mit dem Parameter *rFiltTimeActDancerPosIn* einstellbar (Standard-Einstellung: 5 ms).

Der I-Anteil des PI-Reglers kann mit dem Parameter *lrDancerPosCtrlResetTime* (Reglernachstellzeit) gesetzt werden. In der Standard-Einstellung ist *lrDancerPosCtrlResetTime* = 0 (deaktiviert) gesetzt.

Die Reglerverstärkung wird mit dem Parameter *lrDancerPosCtrlGain* eingestellt.

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 *lrDancerPosRamp* eingestellt (Standard-Einstellung: 1 = 100 %/s). Eine Einblendung des Tänzerreglereinflusses ist dadurch nicht erforderlich.

Mit dem Eingang *xResetCtrl* = TRUE wird der I-Anteil des PI-Reglers ausgeschaltet und die Stellgröße (Ausgang des Reglers) aus dem I-Anteil wird über die Rampenfunktion auf '0' geführt. Die Stellgröße aus dem P-Anteil wird nicht beeinflusst.

#### Einzustellende Parameter

Die Parameter für den PI-Regler und die Tänzerlageregelung befinden sich in der Parameterstruktur [L TT1P\\_scPar\\_WinderDancerCtrl \[Base/State/High\]](#) (23).

```
rFiltTimeActDancerPosIn : REAL := 0.005 // PT1 filter time for lrActDancerPosIn [s]
lrDancerPosRamp : LREAL := 1; // Ramp of lrSetDancerPosScaled [x100%/s]
lrDancerPosCtrlGain : LREAL := 1; // Gain of dancer position control
lrDancerPosCtrlResetTime : LREAL := 0; // Reset time of dancer position control
```

### 3.24 Zugkraftsteuerung über Kennlinienfunktion (Base-Variante)

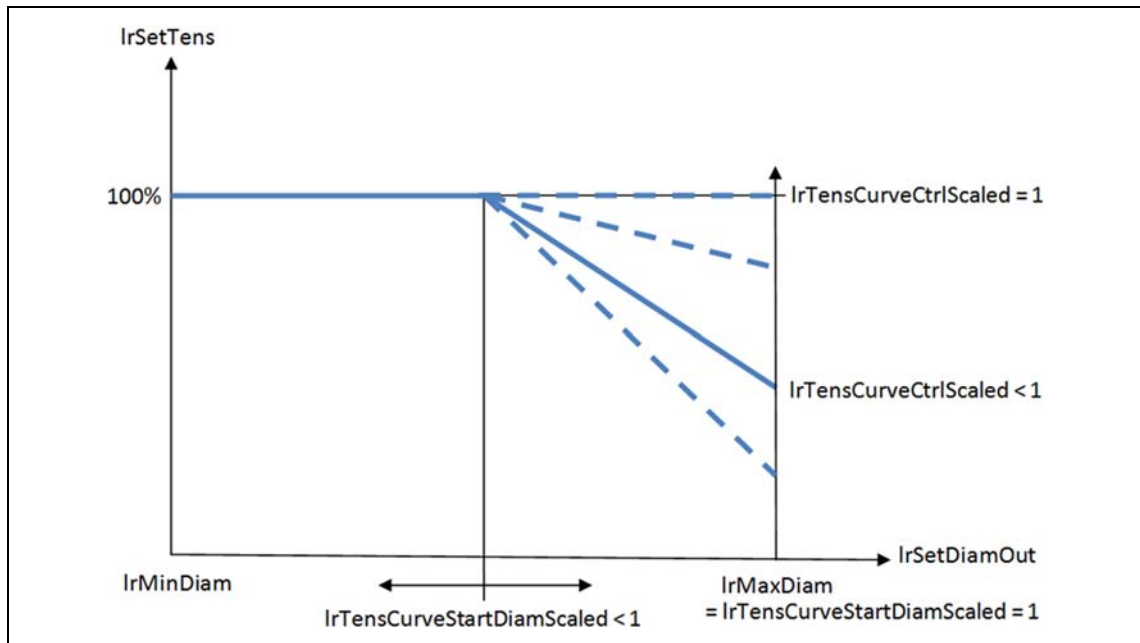
In Abhängigkeit von der Oberfläche und der Art des Wickelmaterials ist es bei vielen Aufwicklern erforderlich, dass die Zugkraft mit zunehmendem Durchmesser reduziert wird, damit der Wickelballen nicht verschoben wird. Man spricht hierbei von der Wickelcharakteristik oder Zugkraftcharakteristik. Der tänzerlagegeregelte Wickler hat keinen direkten Einfluss auf den Bahnzug; dieser wird durch den Druck oder das Gewicht am Tänzer bestimmt.

Dennoch ist es üblich die Zugkraftbeeinflussung in der Wicklersteuerung vorzunehmen, um den adaptierten Sollwert dann, z. B. auf ein pneumatisches Stellglied, wieder zu geben.

Damit die materialabhängige Charakteristik erreicht wird, wird der eigentliche Zugkraftsollwert aus dem Eingang *IrSetTens* über eine lineare Kennlinienfunktion durchmesserabhängig bewertet.

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



[3-11] Kennlinie für einen linearen Zugkraftverlauf

#### Einzustellende Parameter

Die Parameter für die "Zugkraftsteuerung über Kennlinienfunktion" befinden sich in der Parameterstruktur [L TT1P\\_scPar\\_WinderDancerCtrl \[Base/State/High\]](#) (23).

```
IrTensCurveStartDiamScaled : LREAL := 0; // Initial point of the characteristic line
                                   [x100%]
IrTensCurveCtrlScaled : LREAL := 0; // Gapping of the characteristic line [x100%]
IrMaxDiam : LREAL := 0; // Maximal diameter [mm]
IrMinDiam : LREAL := 0; // Minimal diameter [mm]
```

### 3.25 Bahnrißüberwachung

Das Technologiemodul bietet zwei Möglichkeiten einer Bahnrißüberwachung:

- A. Bei einem Bahnriß entwickelt sich der berechnete Durchmesser entgegen der Wickelrichtung (Abwickeln oder Aufwickeln).

Die Überwachung wird mit dem Eingang *xWebBreakMonit* = TRUE und dem Parameter *xWebBreakMode* = 2 aktiviert. Damit ist eine Durchmesseränderung entgegen der Wickelrichtung nur noch innerhalb des im Parameter *lrWebBreakWindow* eingestellten Fensters zulässig.

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



#### Hinweis!

**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.

- B. Auswertung der Tänzerposition:

Ein Bahnriß wird festgestellt, wenn die Tänzerposition die untere Endlage erreicht.

Die Überwachung wird mit dem Eingang *xWebBreakMonit* = TRUE und dem Parameter *xWebBreakMode* = 1 aktiviert.

Reaktionen auf einen Bahnriß:

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

#### Einzustellende Parameter

Die Parameter für die Bahnrißüberwachung befinden sich in der Parameterstruktur [L TT1P\\_scPar\\_WinderDancerCtrl \[Base/State/High\]](#) (23).

```
lrWebBreakWindow : LREAL := 0.1; // Window for web break 0..1 [x100%]
wWebBreakMode : WORD := 0; // Select web break mode:
    0: Use Dancer Position and lrWebBreakWindow for detecting the Web break.
    1: Use only dancer position for detecting the Web Break.
    2: Use only lrWebBreakWindow for detecting the Web Break.
```

### 3.26 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 Einstellung für das Technologiemodul ausgeführt werden.

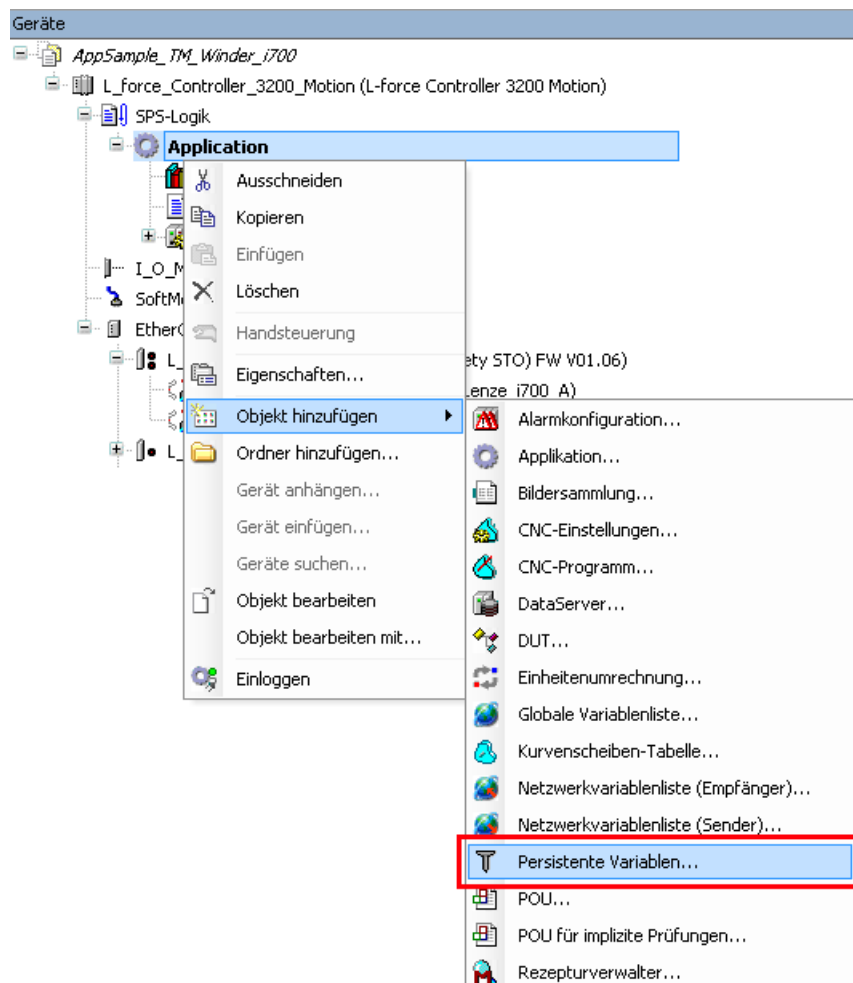


So legen Sie im »PLC Designer« persistente Variablen an:

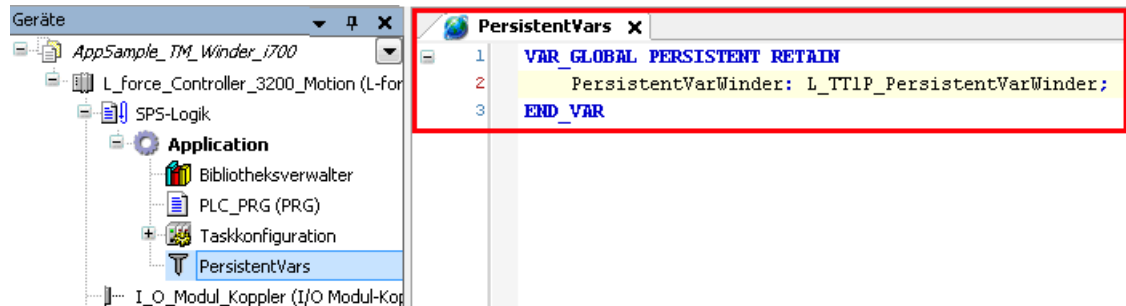
#### Hinweis!

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

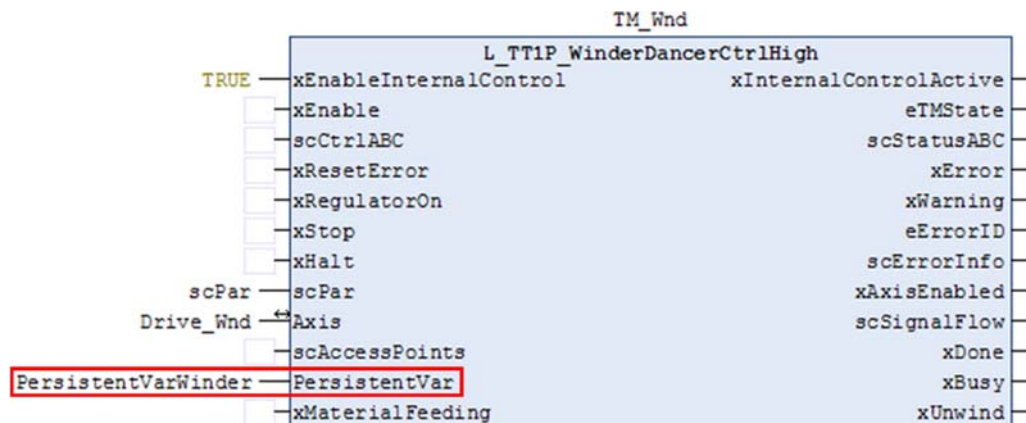
1. Im Kontextmenü zu **Application** mit dem Befehl **Objekt hinzufügen** → **Persistente Variablen...** die globale Liste für die Verwaltung von persistenten Variablen hinzufügen.



- Die Referenz der persistenten Variablen "L\_TT1P\_PersistentVarWinder" in der globalen Struktur der persistenten Variablen instanziiieren.



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



### 3.27 Beschleunigungskompensation

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

Das Beschleunigungsdrehmoment muss also berechnet und als Zusatzdrehmoment vorgesteuert werden.

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

Die Änderung des Drehzahlwertes (neuer Wert - alter Wert) entspricht dabei der Beschleunigung des Wicklers. Die Wicklerdrehzahl wird aus der Liniengeschwindigkeit berechnet.

In der Praxis ist mit einem nicht ideal, stetig verlaufenden Liniengeschwindigkeitssignal zu rechnen. Über die Parameter *IrAccCmpsGainAcc* und *IrAccCmpsGainDec* kann die Auflösung des Signals, welches differenziert wird, eingestellt werden. Zudem kann das Signal vorher über eine PT1-Funktionalität geglättet werden. Die PT1-Zeitkonstante wird über den Parameter *rFiltTimeAccSpd* eingestellt. Zur Rauschunterdrückung kann ein Nacheilbereich über das berechnete Beschleunigungsmoment verschaltet werden. Der Nacheilbereich wird über den Parameter *IrAccCmpsDeadBandTrqScaled* in der Einheit [x 100%] eingestellt.

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

Zur Bildung einer Beschleunigung ist eine Differenzierung der Liniengeschwindigkeit erforderlich. Je nach Auflösung und Stabilität dieses Signals kann es erforderlich sein, die Empfindlichkeit der Differenzierung herabzusetzen. So führen Leitwertschwankungen nicht zu Sprüngen in der Beschleunigung. Unterschiedliche Materialbreiten oder Materialdichten können prozentual über den Eingang *IrMIInertiaAdapt* berücksichtigt werden.

### Massenträgheitsmomente vorgeben



#### Hinweis!

Die Vorgabe der Massenträgheit muss auf die Wicklerwelle und nicht auf die Motorwelle bezogen werden.

Die Trägheit (J) von Motorwelle auf die Wicklerwelle kann mit folgender Gleichung umgerechnet werden:

Berechnung der Trägheit (J) von Motorwelle auf die Wicklerwelle		
$J_{Winder} = i^2 \cdot J_{Motor}$ <p>mit</p> $i = \frac{n_{Motor}}{n_{Winder}}$		
Formelzeichen	Beschreibung	Maßeinheit
$J_{Winder}$	Massenträgheitsmoment der Wicklerwelle	kgcm <sup>2</sup>
$J_{Motor}$	Massenträgheitsmoment der Motorwelle	kgcm <sup>2</sup>
i	Getriebefaktor	
$n_{Motor}$	Motordrehzahl	revs/s
$n_{Winder}$	Wicklerdrehzahl	revs/s

Das Massenträgheitsmoment setzt sich aus einem konstanten und einem durchmesserabhängigen Anteil zusammen. Der konstante Anteil wird durch das Massenträgheitsmoment des Motors bestimmt (Codestelle C00273/1 bei Lenze-Motoren). Der variable Anteil wird aus dem Durchmesser sowie der maximalen und konstanten Massenträgheit im Technologiemodul ermittelt.

Die Einstellung des konstanten Massenträgheitsmoments erfolgt mit dem Parameter *lrConstMinertia*.

Die Einstellung des maximalen Massenträgheitsmoments (voller Wickelballen) erfolgt über den Parameter *lrMaxMinertia*.

#### Einzustellende Parameter

Die Parameter für die Beschleunigungskompensation befinden sich in der Parameterstruktur [L TT1P\\_scPar\\_WinderDancerCtrl \[Base/State/High\]](#) (23).

```

rFiltTimeAccSpd : REAL := 0.005; // Filtertime ActReelSpeed during AccComp [s]
lrAccCmpsDeadBandTrqScaled : LREAL := 0.10; // Dead-band of winder torque [Nm]
lrAccCmpsGainAcc : LREAL := 1.05; // [x100%]
lrAccCmpsGainDec : LREAL := 0.95; // [x100%]
lrConstMinertia : LREAL := 9; // Constant Minertia J_min [kgcm^2]
lrMaxMinertia : LREAL := 50; // Maximal Minertia J_max [kgcm^2]

```

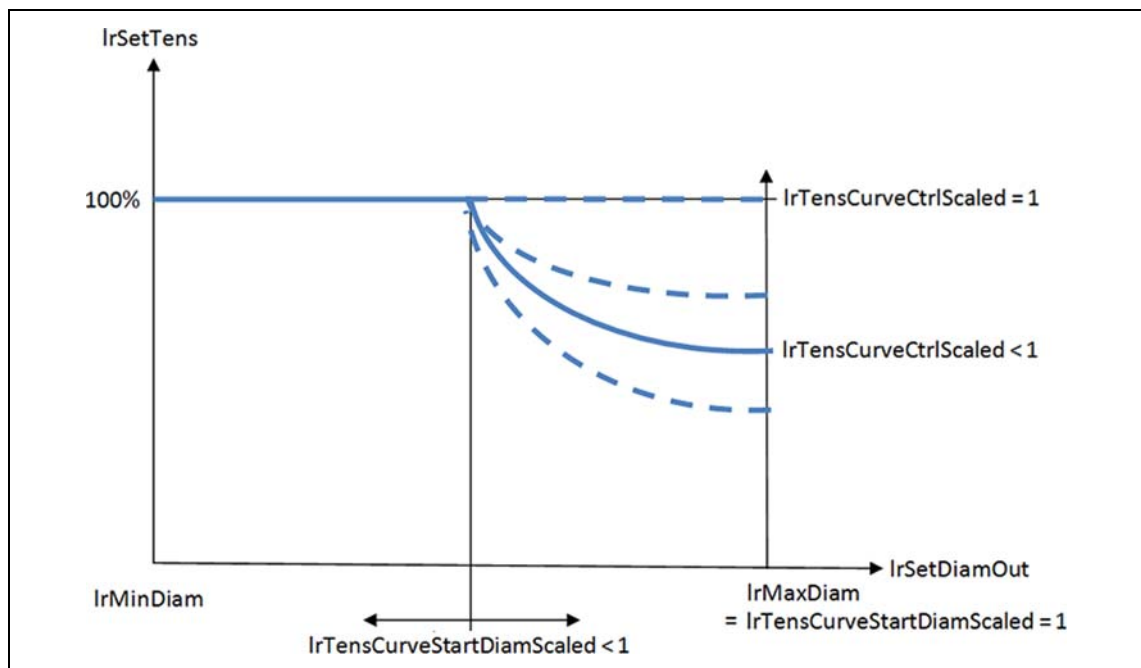
### 3.28 Zugkraftsteuerung über Kennlinienfunktion/Wickelcharakteristik

Die Kennlinienfunktion zur Zugkraftsteuerung ist in der State-Variante erweitert. Damit die materialabhängige Charakteristik erreicht wird, wird der eigentliche Zugkraftsollwert aus dem Eingang *IrSetTens* über eine Kennlinienfunktion durchmesserabhängig bewertet.

Die Adaption kann entsprechend verschiedener Prinzipien erfolgen:

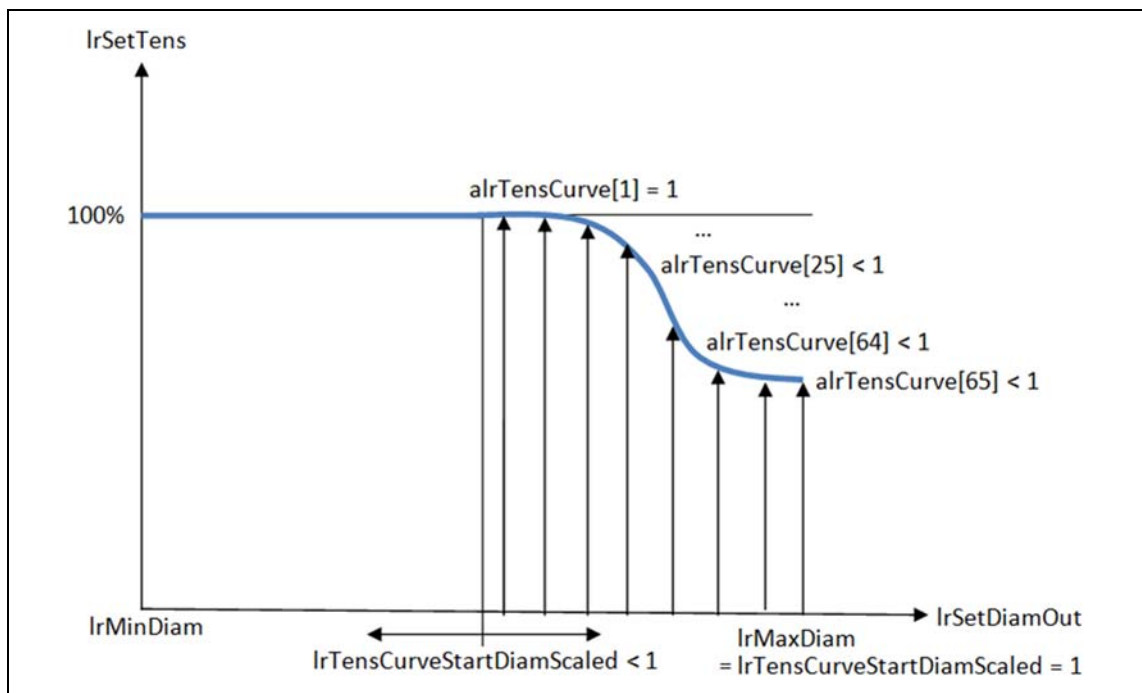
- Kennlinie für einen linearen Zugkraftverlauf (*dwSelectTensCurve* = 0 )
- Kennlinie für einen linearen Drehmomentverlauf (*dwSelectTensCurve* = 1 )
- Frei definierbare Kennlinie mit 64 Stützpunkten (*dwSelectTensCurve* = 2 )

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



[3-12] Kennlinie für einen linearen Drehmomentverlauf





[3-13] Kennlinie mit freidefinierbaren Stützstellen

### Einzustellende Parameter

Die Parameter für die Kennlinienfunktion befinden sich in der Parameterstruktur [L TT1P\\_scPar\\_WinderDancerCtrl \[Base/State/High\]](#) (23).

```
dwSelectTensCurve : DWORD := 0;           // 0..3
alrTensCurve : ARRAY[1..65] OF LREAL;     // Tension curve
lrTensCurveCtrlScaled : LREAL := 0;        // 0 .. 1 [x100%]
lrTensCurveStartDiamScaled : LREAL := 0;   // [x100%]
lrMaxDiam : LREAL := 0;                   // Maximal diameter [mm]
lrMinDiam : LREAL := 0;                   // Minimal diameter [mm]
```

### 3.29 Identifikation der Massenträgheitsmomente

Zur Kompensation des Beschleunigungsdrehmoments ist die Parametrierung oder Identifikation des konstanten Massenträgheitsmoments (Motor + Getriebe + Wicklerwelle) und des maximalen Massenträgheitsmoments (mit vollem Wickelballen) erforderlich.

#### Identifikation des konstanten Massenträgheitsmoments

Die Wicklerwelle ist leer (ohne Material).

Mit dem Parameter *IrIdentMInertiaMaxSpdScaled* wird die maximale Motordrehzahl in [x 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  $\rightarrow$  TRUE) am Eingang *xExecutIdentMInertia* erfolgt die Ermittlung des Massenträgheitsmoments. Am Ausgang *IrIdentMInertia* wird das ermittelte Massenträgheitsmoment angezeigt.

#### 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 Wicklerwelle ( $J_{\text{Winder}}$ ) 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.



#### Hinweis!

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

Übernehmen Sie die Werte der identifizierten Massenträgheitsmomente in die Parameterstruktur [L TT1P\\_scPar\\_WinderDancerCtrl \[Base/State/High\]](#) (23).

### Einzustellende Parameter

Die Parameter für die Identifikation der Massenträgheitsmomente befinden sich in der Parameterstruktur [L TT1P\\_scPar\\_WinderDancerCtrl \[Base/State/High\]](#) (23).

```
rFiltTimeIdentMINertiaSpd : REAL := 0.01; // Filter time ActReelSpeed during Ident
                                   MINertia 1 = 1[s]
rFiltTimeIdentMINertiaTrq : REAL := 0.005; // Filtertime ActTorque during Ident
                                   MINertia 1 = 1[s]
lrIdentMINertiaMaxSpdScaled : LREAL := 0.2; // Max Ident Speed [x 100%]
lrIdentMINertiaMaxTrqScaled : LREAL := 0.2; // Max Ident Torque [x 100%]
```

### 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:		
$J_{\text{MaxWinder}} = i^2 \cdot J_{\text{Motor}} + \frac{m \cdot d_{\text{max}}^2}{800}$		
Formelzeichen	Beschreibung	Maßeinheit
$J_{\text{MaxWinder}}$	Maximales Massenträgheitsmoment der Wicklerwelle	kgcm <sup>2</sup>
$J_{\text{Motor}}$	Massenträgheitsmoment der Motorwelle	kgcm <sup>2</sup>
$i$	Getriebefaktor	
$B$	Materialbreite	mm
$\rho$	Materialdichte	kg/dm <sup>3</sup>
$d_{\text{max}}$	Maximaler Durchmesser	mm
$d_{\text{min}}$	Minimaler Durchmesser (Hülsendurchmesser)	mm
$m$	Masse	kg

### 3.30 Adaption der Drehzahlregler-Verstärkung

#### Voraussetzungen

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

#### Adaption der Drehzahlregler-Verstärkung aktivieren/deaktivieren

Die Adaption der Drehzahlregler-Verstärkung ist vom Zustand des TMs sowie von einer aktuell ausgeführten Funktion unabhängig und kann daher zu einem beliebigen Zeitpunkt aktiviert oder deaktiviert werden.

Eingang *scPar.xAdaptSpdCtrlGain* = TRUE: Adaption Drehzahlregler-Verstärkung aktiviert.

Eingang *scPar.xAdaptSpdCtrlGain* = FALSE: Adaption Drehzahlregler-Verstärkung deaktiviert.

#### Funktionsweise

Der Wert für die Adaption wird im TM berechnet, wobei die Berechnungsvorschrift über den Adaptionsmodus *scPar.eAdaptSpdCtrlGainMode* (siehe unten) vorgegeben wird.

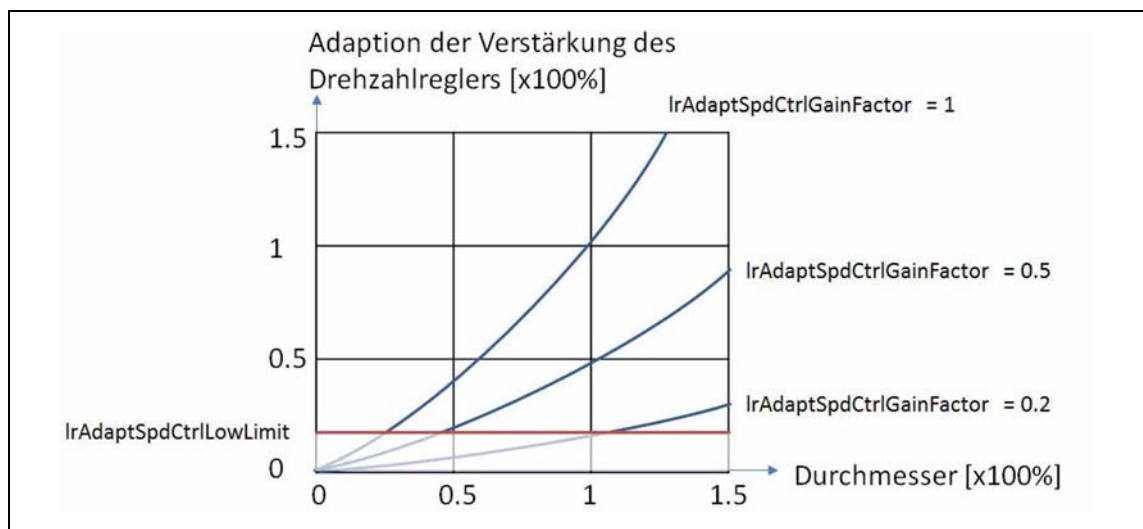
Bereich für den Wert der Adaption: 0 ... 1 (1 = 100 % der Drehzahlverstärkung aus der Einstellung des Drehzahlreglers)

Der im Drehzahlregler eingestellte resultierende Adaptionswert kann mit einem Faktor aus dem Eingang *scPar.lfAdaptSpdCtrlGainFactor* multiplikativ beeinflusst werden.

Über den Parameter *scPar.lfAdaptSpdCtrlLowLimit* wird der kleinste zulässige Wert für die Adaption der Drehzahlregler-Verstärkung festgelegt.

#### 3.30.1 Adaptionsmodus *eAdaptSpdCtrlGainMode* = 0 (DiamToSquare)

Im Modus *eAdaptSpdCtrlGainMode* = 0 wird die Adaption aus dem skalierten Durchmesser (Ausgang *lfSetDiamScaledOut*) zum Quadrat berechnet.

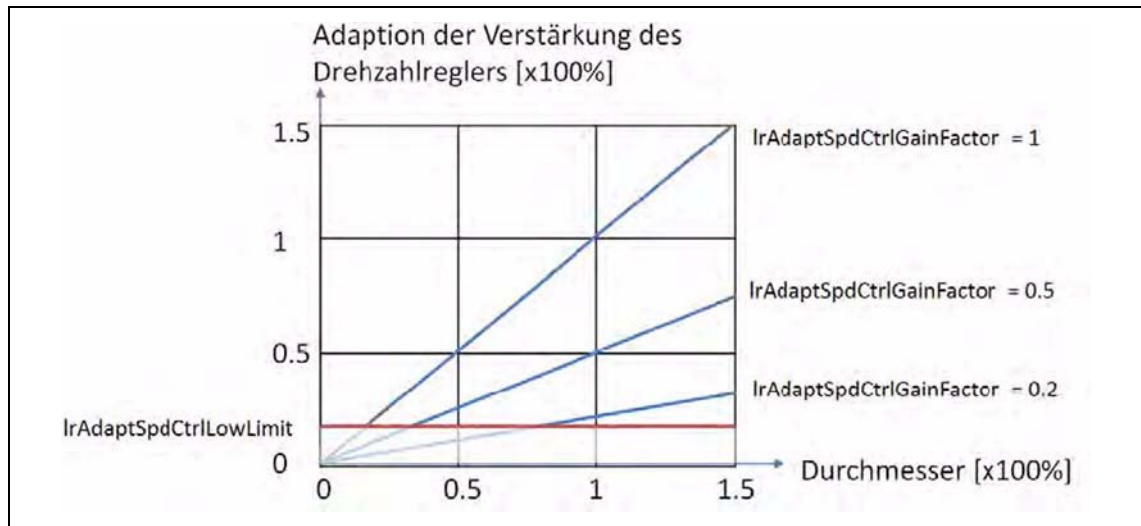


[3-14] Adaption des Drehzahlreglers in Abhängigkeit des Durchmessers zum Quadrat unter Einfluss von *lfAdaptSpdCtrlGainFactor*

Beim maximalen Durchmesser wird der Adaptionswert = 1 gesetzt. Über den Parameter *scPar.lfAdaptSpdCtrlLowLimit* wird die Adaption der Drehzahlregler-Verstärkung nach unten begrenzt.

### 3.30.2 Adaptionsmodus `eAdaptSpdCtrlGainMode:= 1 (Diam)`

Im Modus `eAdaptSpdCtrlGainMode:= 1` wird die Drehzahlregler-Adaption proportional zum skalierten Durchmesser (Ausgang `IrSetDiamScaledOut`) berechnet.



[3-15] Adaption des Drehzahlreglers in Abhängigkeit des Durchmessers unter Einfluss von `IrAdaptSpdCtrlGainFactor`

Beim maximalen Durchmesser wird der Adaptionswert = 1 gesetzt. Über den Parameter `IrAdaptSpdCtrlLowLimit` wird die Adaption der Drehzahlregler-Verstärkung nach unten begrenzt.

### 3.30.3 Adaptionsmodus `eAdaptSpdCtrlGainMode:= 2 (Inertia)`

In einem idealen Modell des Wicklerantriebs betrachtet man Motor und Wickelballen als ein starres Ein-Masse-System. Damit verhält sich die optimale Verstärkung des Drehzahlreglers direkt proportional zum Massenträgheitsmoment  $J$  mit einer  $d^4$ -Funktion.

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

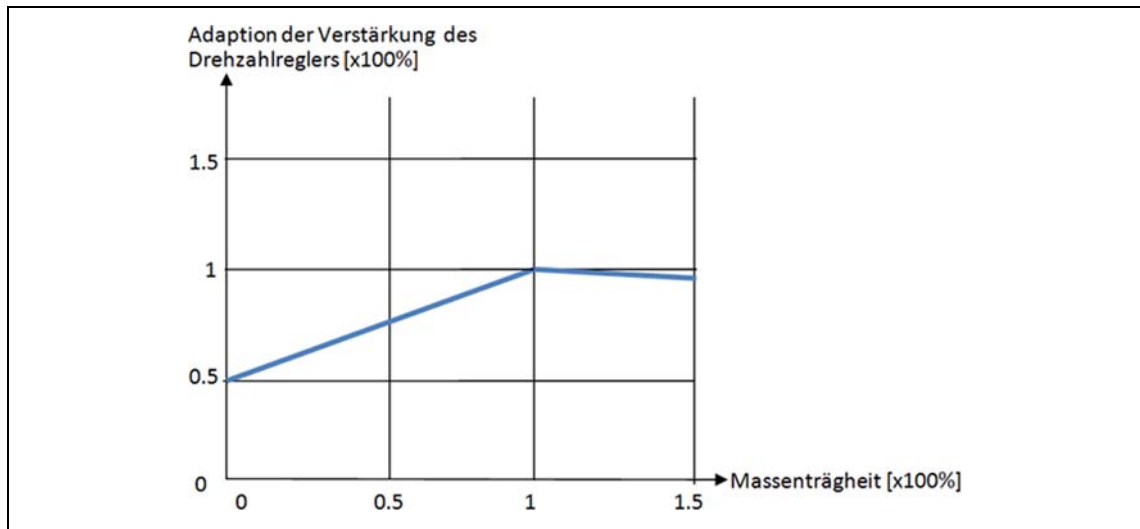
Für den Modus `eAdaptSpdCtrlGainMode:= 2 (Inertia)` ist die Angabe der Massenträgheiten erforderlich:

- Massenträgheit des leeren Wickelballens `scPar.IrConstMinertia` beim minimalen Durchmesser `scPar.IrMinDiam`
- Massenträgheit des Wickelballens mit Material `scPar.IrMaxMinertia` beim maximalen Durchmesser `scPar.IrMaxDiam`

Die Massenträgheit kann entweder berechnet oder über das TM identifiziert werden.

► [Identifikation der Massenträgheitsmomente](#) (66)

Wenn die beiden Massenträgheiten `scPar.IrConstMinertia` und `scPar.IrMaxMinertia` bekannt sind, wird die Adaption anhand der folgenden Kennlinie festgelegt:



[3-16] Voreingestellte Kennlinienfunktion für die Adaption des Drehzahlreglers in Abhängigkeit der Massenträgheit

Diese Kennlinienfunktion beinhaltet in der Standard-Einstellung folgende Werte:

- Untere Begrenzung der Adaption: 50 %
- Obere Begrenzung der Adaption: 100 %
- Lineare Erhöhung der Verstärkung bis 100 % des Massenträgheitsmoments

#### Einzustellende Parameter

Die Kennlinie kann über die Parameterierung verändert oder komplett neu bestimmt werden.

Die einzustellenden Parameter befinden sich in der Struktur *scPar*:

```
alrSpdCtrlGainAdaptX : ARRAY[1..9] OF LREAL :=
    [0.0, 0.2, 0.4, 0.6, 0.8, 1.0, 1.2, 1.4, 1.6];
alrSpdCtrlGainAdaptY : ARRAY[1..9] OF LREAL :=
    [0.5, 0.6, 0.7, 0.8, 0.9, 1.0, 0.98, 0.95, 0.95];
```

---

### 3.31 Regelabweichung im Bereich reduzierter Empfindlichkeit

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

Die Regelabweichung ergibt sich aus der Differenz der Werte aus Eingang *lrSetDancerPosScaled* und Ausgang *lrActDancerPosScaled*.

Mit dem Parameter *lrReducedGainWindow* lässt sich ein Toleranzbereich einstellen, in dem die Regelabweichung mit einer geringeren Verstärkung an den Regler weitergegeben wird. Der Toleranzbereich wird ober- und unterhalb um den Sollwert (Eingang *lrSetDancerPosScaled*) gelegt.

Mit dem Parameter *lrReducedGain* erfolgt die Einstellung, auf welchen Wert die Verstärkung im festgelegten Toleranzbereich reduziert werden soll. Das heißt im Toleranzbereich wirkt die reduzierte Verstärkung (*lrReducedGain*).

#### Einzustellende Parameter

Die Parameter für die Regelabweichung befinden sich in der Parameterstruktur [L TT1P\\_scPar\\_WinderDancerCtrl \[Base/State/High\]](#) (23).

```
lrReducedGain : LREAL := 0;  
lrReducedGainWindow : LREAL := 0;
```

### 3.32 Beendigung des Wickelprozesses

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

- A. Eingänge *xDancerCtrl* = FALSE und *xSyncLineVel* = TRUE setzen.

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

Zustandswechsel: DANCERCTRL ==> SYNCLINEVEL

- B. Eingänge *xDancerCtrl* = FALSE und *xSyncLineVel* = FALSE setzen.

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

- *eDancerCtrlStopMode* = 0: Halt  
Die Achse wird über die Verzögerung (*lrHaltDec*) und den Ruck (*lrJerk*) in den Stillstand geführt.
- *eDancerCtrlStopMode* = 1: Move ABS  
Die Achse wird mit der Geschwindigkeit (*lrVel*), Beschleunigung (*lrAcc*), Verzögerung (*lrDec*) und den Ruck (*lrJerk*) in die Zielposition (*lrPos\_Dist*) gefahren.
- *eDancerCtrlStopMode* = 2: Move Rel  
Die Achse wird mit der Geschwindigkeit (*lrVel*), Beschleunigung (*lrAcc*), Verzögerung (*lrDec*) und den Ruck (*lrJerk*) nach der gefahrenen Wegstrecke (*lrPos\_Dist*) in den Stillstand geführt.

Zustandswechsel: DANCERCTRL ==> STOP

#### Einzustellende Parameter

Die einzustellenden Parameter zur Beendigung des Wickelprozesses befinden sich in der Parameterstruktur [L\\_TT1P\\_scPar\\_WinderDancerCtrl \[Base/State/High\]](#) (23).

```
eDancerCtrlStopMode : L_TT1P_DancerCtrlStopMode := 0;
lrPos_Dist : LREAL := 0;
lrHaltDec : LREAL := 3600;
lrJerk : LREAL := 100000;
lrVel : LREAL := 50;
lrAcc : LREAL := 100;
lrDec : LREAL := 100;
```

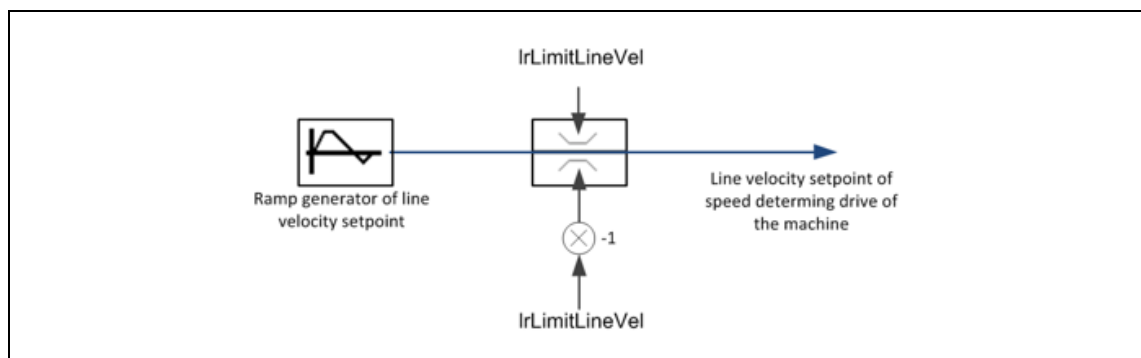


### 3.33 Begrenzung der Master-Liniengeschwindigkeit

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 TM Winder.

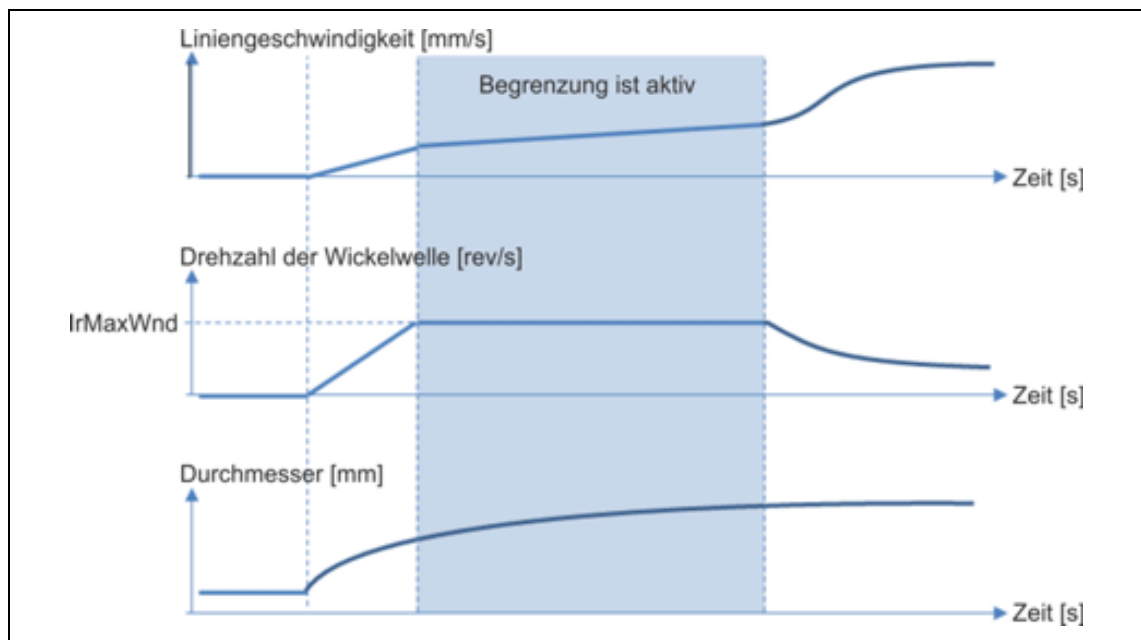
Die maximale Drehzahl der Wicklerwelle (abtriebsseitig) wird über den Parameter *scPar.IrMaxWndSpd* festgelegt. Mit Eingabe dieses Parameters ist die Begrenzungsfunktion direkt freigegeben.

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 *scPar.IrMaxWndSpd*.



[3-17] Signalfluss für die Berechnung der Liniengeschwindigkeit-Sollwert-Begrenzung

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 zugelassen Drehzahl der Winkelwelle *scPar.IrMaxWndSpd* nicht zu überschreiten.



[3-18] Beispiel für die Begrenzung der Liniengeschwindigkeit

### 3.34 Identifikation des Durchmessers durch Anheben des Tänzers

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.

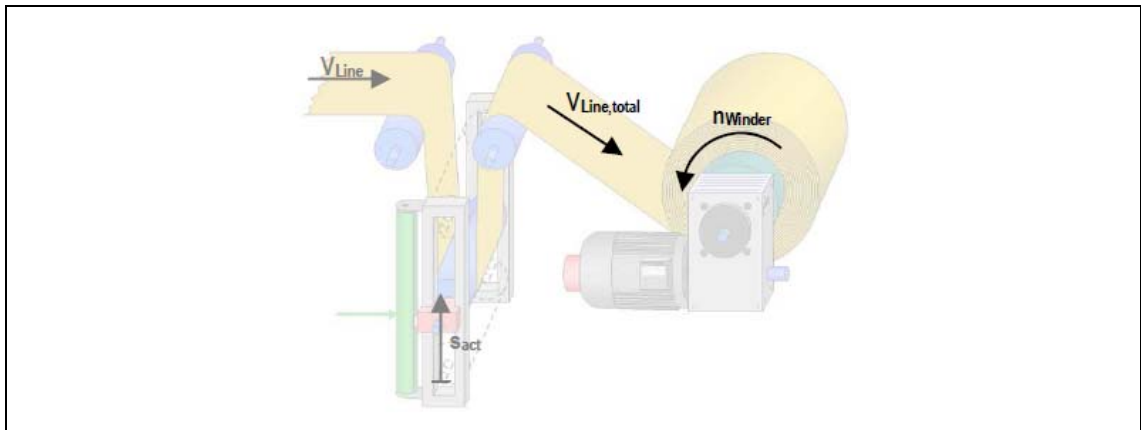


#### Hinweis!

Während der Durchmesser-Identifikation wird bei einer aktiven [Adaption der Drehzahlregler-Verstärkung](#) (`xAdaptSpdCtrlGain = TRUE`) diese auf den definierten Wert `scPar.lrlIdentDiamSpCtrlGain` gesetzt.

#### Voraussetzung

- Die Achse ist freigegeben (Eingang `xRegulatorOn = TRUE`)
- Das TM befindet sich im Zustand "READY"



[3-19]

#### Identifikation starten

Mit einer FALSE→TRUE-Flanke am Eingang `xExecuteIdentDiam` startet die Identifikation. Anschließend wechselt das TM WinderDancer in den Zustand "IDENTDIAMETER".

#### Funtionsablauf

Der Durchmesser wird auf den Startwert von 50 % des maximalen Durchmessers `scPar.lrlMaxDiam` gesetzt. Aufgrund der Unterschreitung der minimalen Liniengeschwindigkeit (`scPar.lrlMinLineVel`), wird das interne Halten des Durchmessers deaktiviert.

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



### Hinweis!

Die Profilparameter beziehen sich auf die Umfangsgeschwindigkeit des Wicklers.

- Geschwindigkeit *scPar.IrIdentDiamVel* in [mm/s]
- Beschleunigung (*scPar.IrIdentDiamAcc*) in [mm/s<sup>2</sup>]
- Verzögerung (*scPar.IrIdentDiamDec*) in [mm/s<sup>2</sup>]
- Ruck *scPar.IrIdentDiamJerk* in [mm/s<sup>3</sup>]

Durch die Materialbewegung wird der Tänzer angehoben.

Anschließend wird geprüft, ob an der Wickelwelle der vorgegebene Durchmesser-Berechnungszyklus mit der erwarteten Häufigkeit aus *scPar.IrDiamIdentCalcCycles* durchlaufen werden konnte.

Der interne Zähler für *IrDiamIdentCalcCycles* wird nur inkrementiert, wenn ...

*xDancerMinPos* = FALSE (untere Endlage)

UND

*xDancerMaxPos* = FALSE (obere Endlage).

Ein Berechnungszyklus ist nach der verkürzten Berechnungsdistanz *scPar.IrDiamCalcReduced* abgeschlossen.

Eine erfolgreiche Identifikation des Durchmessers wird durch *xDone* = TRUE gemeldet.

Nach einer erfolgreichen Identifizierung schaltet das TM in den Zustand ...

- "READY", wenn der Eingang *xDanceCtrl* = FALSE ist.
- "DANCERCTRL", wenn der Eingang *xDanceCtrl* = TRUE ist; der Wickelbetrieb ist damit aktiviert.

### Abbruch Identifikation

Wird die maximale Tänzerposition (*scPar.IrDiamIdentMaxPos*) oder die obere Tänzerendlage (*xDancerMaxPos* = TRUE) erreicht, ohne dass die Identifikation erfolgreich abgeschlossen werden konnte, wird die Identifikation mit einem "Stopp" abgebrochen.

Der Abbruch wird durch *xError* = TRUE und Ausgabe *eErrorId* = 17164 (Fehlertext: *MaxDancerPosDuringDiameterIdent*) gemeldet. Das Technologiemodul wechselt in den Zustand "ERROR".

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

**3.35 CPU-Auslastung (Beispiel Controller 3231 C)**

Die folgende Tabelle zeigt die CPU-Auslastung in Mikrosekunden am Beispiel des Controller 3231 C (ATOM™-Prozessor, 1.6 GHz).

Variante	Beschaltung des Technologiemoduls	CPU-Auslastung	
		Durchschnitt	Maximale Spitze
Base	xEnable := TRUE; xRegulatorOn := TRUE; xSyncLineVel := TRUE;	75 µs	106 µs
State	xEnable := TRUE; xRegulatorOn := TRUE; xSyncLineVel := TRUE;	90 µs	125 µs
High	xEnable := TRUE; xRegulatorOn := TRUE; xSyncLineVel := TRUE;	100 µs	137 µs

## A

Abweichung im Bereich reduzierter Empfindlichkeit [71](#)  
Access points [36](#)  
Adaption der Drehzahlreglerverstärkung [68](#)  
Anlauf der Achsen [14](#)  
Anwendungshinweise [8](#)  
Aufbau der Sicherheitshinweise [8](#)  
Ausgänge [21](#)

## B

Bahnrißüberwachung [59](#)  
Beendigung des Wickelprozesses [72](#)  
Begrenzung der Master- Liniengeschwindigkeit [73](#)  
Beschleunigungskompensation [62](#)  
Betriebsmodus [13](#)

## C

CPU-Auslastung (Beispiel Controller 3231 C) [76](#)

## D

Dokumenthistorie [6](#)  
Drehzahlreglerverstärkung (Adaption) [68](#)  
Drehzahlvorsteuerung [41](#)  
Drehzahlvorsteuerung prüfen [41](#)  
Durchmesser halten [43](#)  
Durchmesser vorgeben [44](#)  
Durchmesserberechnung [42](#)  
Durchmesserberechnung mit Korrektur der Tänzerposition [45](#)  
Durchmessersensor-Signal [44](#)

## E

Eingänge [17](#)  
Eingänge und Ausgänge [16](#)  
E-Mail an Lenze [79](#)  
eTMState [31](#)

## F

Feedback an Lenze [79](#)  
Funktionen des Technologiemoduls (Übersicht) [12](#)  
Funktionsbaustein L\_TT1P\_WinderDancerCtrlBase/State/High [15](#)  
Funktionsbeschreibung "Winder Dancer-controlled" [11](#)

## G

Gestaltung der Sicherheitshinweise [8](#)

## H

Handfahren (Jogging) [50](#)  
Hinweise zum Betrieb des Technologiemoduls [13](#)

## I

Identifikation der Massenträgheitsmomente [66](#)  
Identifikation des Durchmessers [74](#)

## J

Jogging (Handfahren) [50](#)

## K

Kontrollierter Anlauf der Achsen [14](#)  
Korrektur der Tänzerposition [45](#)

## L

L\_TT1P\_scAP\_WinderDancerCtrlBase [36](#)  
L\_TT1P\_scAP\_WinderDancerCtrlHigh [36](#)  
L\_TT1P\_scAP\_WinderDancerCtrlState [36](#)  
L\_TT1P\_scPar\_WinderDancerCtrlBase [23](#)  
L\_TT1P\_scPar\_WinderDancerCtrlHigh [23](#)  
L\_TT1P\_scPar\_WinderDancerCtrlState [23](#)  
L\_TT1P\_scSF\_WinderDancerCtrlBase [34](#)  
L\_TT1P\_scSF\_WinderDancerCtrlHigh [34](#)  
L\_TT1P\_scSF\_WinderDancerCtrlState [34](#)  
L\_TT1P\_WinderDancerCtrlBase [15](#)  
L\_TT1P\_WinderDancerCtrlHigh [15](#)  
L\_TT1P\_WinderDancerCtrlState [15](#)  
Leitwert-Quelle für die Durchmesserberechnung [40](#)

## M

Massenträgheitsmomente identifizieren [66](#)  
Massenträgheitsmomente vorgeben [63](#)  
Master-Liniengeschwindigkeit begrenzen [73](#)  
Materiallängenzähler [47](#)  
Materiallängenzählung (Quellen) [48](#)  
Materialzuführung an den Wickler [39](#)  
Max. Massenträgheitsmoment berechnen [67](#)

## N

Normierung der Tänzerlage [54](#)

## P

Parameterstruktur L\_TT1P\_scPar\_WinderDancerCtrlBase/  
State/High [23](#)  
Persistente Variablen [60](#)  
PI-Regler für die Tänzerlageregelung [57](#)

## Q

Quellen für die Materiallängenzählung [48](#)

## R

Regelabweichung im Bereich reduzierter Empfindlichkeit [71](#)

## S

Sicherheitshinweise [8](#), [9](#)  
Signal vom Durchmessersensor [44](#)  
Signalfluss des Technologiemoduls "Winder Dancer-  
controlled" [33](#)  
Signalfluss zur Berechnung des Durchmessers [32](#)  
Signalflusspläne [32](#)  
Startdurchmesser vorgeben [44](#)

State machine [30](#)

Struktur der Angriffspunkte

L\_TT1P\_scAP\_WinderDancerCtrlBase/State/High [36](#)

Struktur des Signalflusses

L\_TT1P\_scSF\_WinderDancerCtrlBase/State/High [34](#)

Synchronisierung auf die Liniengeschwindigkeit [52](#)

## T

Tänzerlage (Normierung) [54](#)

Tänzerposition korrigieren [45](#)

Tänzerpositions-Überwachung [56](#)

Teaching-Funktion für Tänzerendlagen [55](#)

Trimmung [53](#)

## U

Überwachung der Tänzerposition [56](#)

## V

Variablenbezeichner [7](#)

Verwendete Konventionen [7](#)

## W

Wickelprozess beenden [72](#)

Wickelrichtung (Automatische Erkennung) [38](#)

Wickelrichtung festlegen (Aufwickeln/Abwickeln) [38](#)

Winder Dancer-controlled (Funktionsbeschreibung) [11](#)

## Z

Zielgruppe [5](#)

Zugkraftsteuerung über Kennlinienfunktion (Base-Variante)  
[58](#)

Zugkraftsteuerung über Kennlinienfunktion/  
Wickelcharakteristik [64](#)

Zusätzliche Geschwindigkeit bei der Durchmesserberechnung  
[46](#)

Zustände [30](#)

Zustände des Ausgangs eTMState [31](#)



## 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](mailto:feedback-docu@lenze.com)

Vielen Dank für Ihre Unterstützung.

*Ihr Lenze-Dokumentationsteam*

Lenze Automation GmbH  
Postfach 10 13 52, 31763 Hameln  
Hans-Lenze-Straße 1, 31855 Aerzen  
Germany  
HR Hannover B 205381  
☎ +49 5154 82-0  
📠 +49 5154 82-2800  
✉ [sales.de@lenze.com](mailto:sales.de@lenze.com)  
🌐 [www.lenze.com](http://www.lenze.com)

#### **Service**

Lenze Service GmbH  
Breslauer Straße 3, 32699 Extertal  
Germany  
☎ 008000 24 46877 (24 h helpline)  
📠 +49 5154 82-1112  
✉ [service.de@lenze.com](mailto:service.de@lenze.com)