

SIEMENS

Ausgabe 12.2010

SIMOTION

Kinematiktransformation

Projektierung von
Kinematiktransformationen mit der
SIMOTION Bibliothek *LKTrans*
(Library Kinematic Transformation)

Funktionsbeschreibung

<hr/> <i>Funktionsumfang</i> <hr/>	1
<i>Voraussetzungen</i> <hr/>	2
<i>Allgemeine Funktionsbeschreibung</i> <hr/>	3
<i>Inbetriebnahme</i> <hr/>	4
<i>Konstanten und Datenstrukturen</i> <hr/>	5
<i>Beschreibung der Funktionsbausteine</i> <hr/>	6
<i>Integration eigener Kinematiken</i> <hr/>	7
<i>Fehler und Warnungen</i> <hr/>	8
<i>Anhang</i> <hr/>	9
<i>Stichwortverzeichnis</i>	10

Technische Änderungen des Produktes vorbehalten.

Copyright

Weitergabe sowie Vervielfältigung dieser Unterlage, Verwertung und Mitteilung ihres Inhaltes sind nicht gestattet, soweit nicht ausdrücklich zugestanden. Zuwiderhandlungen verpflichten zu Schadenersatz. Alle Rechte vorbehalten, insbesondere für den Fall der Patenterteilung oder GM-Eintragung

Definitionen und Warnungen

Qualifiziertes Personal

im Sinne der Dokumentation sind Personen, die mit Aufstellung, Montage, Inbetriebsetzung, Betrieb und Instandhaltung der einzusetzenden Produkte der Siemens AG vertraut sind und über die ihrer Tätigkeit entsprechenden Qualifikationen verfügen

z. B.:

- Ausbildung oder Unterweisung bzw. Berechtigung, Stromkreise und Geräte gemäß den Standards der Sicherheitstechnik ein- und auszuschalten, zu erden und zu kennzeichnen.
- Ausbildung oder Unterweisung gemäß den Standards der Sicherheitstechnik in Pflege und Gebrauch angemessener Sicherheitsausrüstung.
- Schulung in Erster Hilfe.

Warnhinweise werden in dieser Dokumentation explizit nicht gegeben. Es wird jedoch ausdrücklich auf die Warnhinweise der Betriebsanleitung für das jeweilige Produkt verwiesen.

Haftungsausschluss

Die Standardapplikationen werden Ihnen unentgeltlich zur Verfügung gestellt. Sie dürfen frei kopiert und benutzt sowie an Dritte weitergegeben werden. Sie dürfen nur unter Beibehaltung aller Schutzrechtsvermerke sowie nur vollständig und unverändert weitergegeben werden. Die kommerzielle Weitergabe an Dritte (z.B. im Rahmen von Share-/Freeware-Distributionen) ist nur mit vorheriger schriftlicher Genehmigung der Siemens Aktiengesellschaft erlaubt. DA DIE APPLIKATIONSBEISPIELE IHNEN UNENTGELTLICH ÜBERLASSEN WERDEN, KÖNNEN DIE AUTOREN UND RECHTSINHABER DAFÜR KEINE GEWÄHRLEISTUNG ÜBERNEHMEN. IHRE BENUTZUNG ERFOLGT AUF EIGENE GEFAHR UND VERANTWORTUNG. DIE AUTOREN UND RECHTSINHABER HAFTEN NUR FÜR VORSATZ UND GROBE FAHRLÄSSIGKEIT. WEITERGEHENDE ANSPRÜCHE SIND AUSGESCHLOSSEN. INSBESONDERE HAFTEN DIE AUTOREN UND RECHTSINHABER NICHT FÜR ETWAIGE MÄNGEL ODER FOLGESCHÄDEN. Falls Sie Fehler in den Applikationsbeispielen bemerken, teilen Sie es uns bitte mit.

Gültige Bedingungen

Es gelten, sofern nicht unten etwas Abweichendes geregelt ist, die „Bedingungen für Lieferungen und Leistungen im Siemens-internen Geschäft“ in der jeweils zum Zeitpunkt der Überlassung gültigen Fassung.

Hinweis auf Warenzeichen

SIMOTION® ist ein Warenzeichen der Siemens AG.
SINAMICS® ist ein Warenzeichen der Siemens AG.
MASTERDRIVES® ist ein Warenzeichen der Siemens AG.
SIMODRIVE® ist ein Warenzeichen der Siemens AG.
SIMATIC® ist ein Warenzeichen der Siemens AG.

Hinweis auf Exportkennzeichen

AL: N
ECCN: N

Vorwort Handbuch

Dieses Handbuch beschreibt die Anwendung der Funktionsbausteine der SIMOTION-Applikationsbibliothek „LKTrans“. Die vorliegende Dokumentation wendet sich ausschliesslich an qualifiziertes Inbetriebnahme- und Service-Personal. Das Handbuch ist als Ergänzung zur SIMOTION Standard-Dokumentation zu verstehen.

Der Umgang mit SIMOTION und der Applikation „LKTrans“ bedingt folgende Voraussetzungen:

- Grundkenntnisse über das System SIMOTION
- Umgang mit dem Engineering System SIMOTION SCOUT
- SIMATIC STEP 7: Kenntnis der HW-Konfig und NetPro
- Kenntnisse zur Parametrierung und Optimierung von Achsregelkreisen.
- Kenntnisse über die eingesetzten Umrichter bezüglich Parametrieren der notwendigen Einstellungen und der Kommunikation über PROFIBUS-DP sowie die Optimierung der Drehzahlregelung

Die Bibliothek „LKTrans“ ist nicht lizenzpflichtig.

Reale Achsen sind lizenzpflichtig. Details hierzu sind der SIMOTION Dokumentation zu entnehmen.

Inhaltsverzeichnis

Vorwort Handbuch	5
Inhaltsverzeichnis	6
Abbildungsverzeichnis	9
Abkürzungsverzeichnis	10
Coding-Standard Festlegungen	10
1 Funktionsumfang	11
1.1 Darstellung des Einsatzfeldes der Applikation.....	11
1.2 Abgrenzungen	12
1.2.1 Verwendbar bei:	12
1.2.2 Nicht verwendbar bei:.....	12
2 Voraussetzungen	13
2.1 Hardware.....	13
2.2 Software	13
2.3 Projekt	13
3 Allgemeine Funktionsbeschreibung	14
3.1 Begriffserklärung	14
3.1.1 Reale Maschinenachsen	14
3.1.2 Bahnobjekt bzw. Bahninterpolator.....	14
3.1.3 Eingangs- und Ausgangsachsen.....	14
3.1.4 Frametransformation	14
3.2 Bibliotheksstruktur	15
4 Inbetriebnahme	16
4.1 Voraussetzungen im Projekt	16
4.2 Anpassen der Bibliothekskonstanten	17
4.3 Beispielaufruf in ST	17
4.3.1 Anlegen der Parameterstruktur	17
4.3.2 Aufruf des Funktionsbausteins	18
4.3.3 Taskeinbindung ins Ablaufsystem.....	19
4.3.4 Wertzuweisung Parameterstruktur	19
4.3.5 Systemfehler deaktivieren	21
5 Konstanten und Datenstrukturen	22
5.1 Globale Konstanten.....	22
5.2 Technologische Parameter	23
5.2.1 Parameterstruktur.....	23
5.2.2 Allgemeine Parameter (sLKTransGeneralType).....	23
5.2.2.1 Parameter der Sollwertabweichung (sLKTransGeneralCmdPathToleranceType).....	23

5.2.2.2	Parameter der Istwertabweichung (sLKTransGeneralActPathToleranceType)	23
5.2.2.3	Parameter derSchleppabweichung (sLKTransGeneralFollowingErrorType)	24
5.2.3	Transformationsparameter (sLKTransTransformationType).....	24
5.2.3.1	Parameter der Frametransformationsparameter (sLKTransType)	24
5.2.4	Parameter der Eingangsachsen (sLKTransInputAxisType).....	24
5.2.5	Parameter der Ausgangsachsen (sLKTransOutputAxisType)	25
5.2.5.1	Parameter der Sollwertüberwachung (sLKTransOutputAxCmdValueToleranceType)	25
5.2.5.2	Dynamikgrenzen des Motion Interface (sLKTransOutputAxMIDynamikType).....	25
5.2.5.3	Dynamikgrenzen bei Stop (sLKTransOutputAxStopDynamikType)	25
5.2.5.4	Dynamikgrenzen beim (Achse-) Heranziehen (sLKTransOutputAxPullDynamikType).....	26
5.2.6	Diagnosestruktur (sLKTransDiagnosticsType).....	27
5.2.6.1	Parameter eines Diagnosepuffereintrags (sLKTransDiagnosticItem)	27
5.2.7	Statuswörter der Achsen	27
5.2.8	Enumeratoren	28
5.2.8.1	Bausteinzustand (eLKTransStateType).....	28
5.2.8.2	Berechnungsart Transformation (eLKTransCalcModeType).....	28
5.2.8.3	Berechnungsart Bewegungsvector (eLKTransMICALCulationType).....	29
6	Beschreibung der Funktionsbausteine	30
6.1	Funktionsbaustein FBLKTransTransformation	30
6.1.1	Aufgabe	30
6.1.2	Taskeinbindung	30
6.1.3	Schematische KOP/FUP-Darstellung.....	31
6.1.4	Parameter des Bausteins	31
6.1.5	Funktionsbeschreibung	32
6.2	Funktionsbaustein FBLKTransKinTransformation	33
6.2.1	Aufgabe	33
6.2.2	Taskeinbindung	33
6.2.3	Schematische KOP/FUP-Darstellung.....	34
6.2.4	Parameter des Bausteins	34
6.2.5	Ablaufdiagramm FBKinTransformation	35
6.2.6	Zustandsmodell FBKinTransformation	35
6.2.7	Aktivierung Motion Interface	36
6.2.7.1	Beispiel	36
6.2.8	Überwachungen	38
6.2.8.1	Bahnabweichung	38
6.2.8.2	Bahnsollwertabweichung	38
6.2.8.3	Bahnschleppabstand	39
6.2.8.4	Sollwertüberwachung Achse	39
6.2.9	Einstellung der Dynamikparameter	40

6.2.10	Beispiel Dynamiken und Überwachungen	41
6.2.11	Frametransformation	42
6.2.11.1	Translation	42
6.2.11.2	Rotation.....	42
6.3	Funktionen FCFrameTrans und FCInvFrameTrans.....	44
6.3.1	Aufgabe	44
6.3.2	Schematische KOP/FUP-Darstellung.....	44
6.3.3	Parameter der Funktionen.....	44
7	Integration eigener Kinematiken.....	45
7.1	Anpassung von Konstanten	45
7.2	Erstellung Vorwärts- und Rückwärtstransformation	45
8	Fehler und Warnungen	48
9	Anhang	52
9.1	Überwachungen	52
10	Stichwortverzeichnis	54

Abbildungsverzeichnis

Bild 1-1: Struktur FBKinTransformation und FBTransformation.....	11
Bild 3-1: Ansicht der Bibliothek im SCOUT Projektnavigator	15
Bild 4-1: Ansicht Beispielkinematik (transId := 4000)	16
Bild 4-2: Projektierte Achsen im Projektnavigator	16
Bild 4-3: Ansicht der Parameterstruktur im Symbolbrowser.....	18
Bild 4-4: Aufruf des Programm im Ablaufsystem.....	19
Bild 6-1: Aufgabe des FBTransformation	30
Bild 6-2: KOP-Ansicht des FBLKTransTransformation	31
Bild 6-3: Bahnabweichung.....	32
Bild 6-4: Anwendung des FBKinTransformation	33
Bild 6-5: KOP-Ansicht des FBKinTransformation.....	34
Bild 6-6: Ablaufdiagramm FBKinTransformation	35
Bild 6-7: Zustandsmodell FBKinTransformation.....	35
Bild 6-8 Grenzen beim Aktivieren des Motion Interface	36
Bild 6-9: Typischer Initialzustand nach Hochlauf.....	37
Bild 6-10: Setzen der Eingangssachspositionen.....	37
Bild 6-11: Übersicht FBLKTRansKinTransformation mit Überwachungen.....	38
Bild 6-12: Struktur Sollwertüberwachung	39
Bild 6-13: Dynamische Begrenzungen bei Einsatz des FBKinTransformation	40
Bild 6-14: Frametransformation im FBKin Transformation	42
Bild 6-15: Rotation des Koordinatensystems	43
Bild 6-16: KOP-Ansicht des FCFrameTrans	44
Bild 6-17: KOP-Ansicht des FCInvFrameTrans.....	44
Bild 7-1: Codeabschnitt zur Konstantendefinition FBTransformation in Unit cPublic.....	45
Bild 7-2: Vorwärts- und Rückwärtstransformation	46
Bild 7-3: Einfaches Transformationsbeispiel	46
Bild 7-4: Verfügbare Anwenderkonstanten.....	46
Bild 7-5: Beispiel (Ausschnitt) einer Fehlerabfrage	47

Abkürzungsverzeichnis

FB	Funktionsbaustein
FC	Funktion
TO	Technologisches Objekt
LKTRANS	Konstantenprefix für Libray Kinematic Transformation
LKTRANS_ERR	Konstantenprefix für Fehlermeldungen Libray Kinematic Transformation

Coding-Standard Festlegungen

Innerhalb von Datenstrukturen gelten folgende Präfixe:

- 1.) Gleitkomma-Variablen werden mit dem Präfix r32 oder r64 gekennzeichnet.
- 2.) time-Variablen werden mit dem Präfix t definiert
- 3.) Bit-Variablen BOOL werden mit dem Präfix bo gekennzeichnet.
- 4.) Datenstruktur-Variablen werden mit dem Präfix s gekennzeichnet.
- 5.) Enumerator-Variablen werden mit dem Präfix e gekennzeichnet.
- 6.) Array-Variablen werden mit dem Präfix a gekennzeichnet.
- 7.) BYTE-Variablen werden mit dem Präfix b8, WORD-Variablen mit dem Präfix b16 und DWORD-Variablen mit dem Präfix b32 gekennzeichnet.
- 8.) DINT-Variablen werden mit dem Präfix i32, INT-Variablen mit dem Präfix i16, SINT-Variablen mit dem Präfix i8, UDINT- Variablen mit dem Präfix u32, UINT-Variablen mit dem Präfix u16, USINT-Variablen mit dem Präfix u8 gekennzeichnet.
- 9.) Diese Festlegungen beziehen sich auch auf die **Innensicht der hier beschriebenen FBs**

1 Funktionsumfang

1.1 Darstellung des Einsatzfeldes der Applikation

Mit den Bausteinen der Standard-Bibliothek „Kinematiktransformation mit SIMOTION“ können einfache kinematische Transformationen, wie sie bei Handling oder Robotikanwendungen vorkommen projiziert werden. Zur Nutzung der Funktionalität ist die Verwendung der Standardapplikationen „Toploading mit SIMOTION“ oder „Handling mit SIMOTION“ nicht zwingend notwendig. Diese Standardapplikationen bieten dem Anwender jedoch alle Funktionalitäten die zur Erstellung kompletter Handlingsprojekte nötig sind. Details können der jeweiligen Dokumentation entnommen werden.

Die hier beschriebenen Bausteine ergänzen die technologischen Funktionalitäten von SIMOTION bzw. die erwähnten Standardapplikationen um folgende Funktionalität:

- Funktionsbaustein FBLKTransTransformation mit einheitlicher Schnittstelle zur Aufnahme der vom Anwender erstellten Kinematiktransformation.
- Funktionsbaustein FBLKTransKinTransformation zur Ansteuerung von realen Achsen mit zyklischen Positionssollwerten über das Motion Interface.
- Die Eingangsachsen werden vor Aktivierung des Motion Interface auf die tatsächlichen, transformierten Positionen der realen Ausgangsachsen gesetzt um Sprünge zu verhindern.
- Überwachung einzelner realer Achsen mittels Sollwertüberwachung
- Überwachung kartesischer Ist- und Sollwerte mittels Bahnabweichung

Hierbei enthält der Funktionsbaustein FBLKTransTransformation die mathematischen Transformationsgleichungen, der FBLKTransKinTransformation übernimmt die Ansteuerungen Achsen, seine Funktionalitäten werden weitgehend konfiguriert.

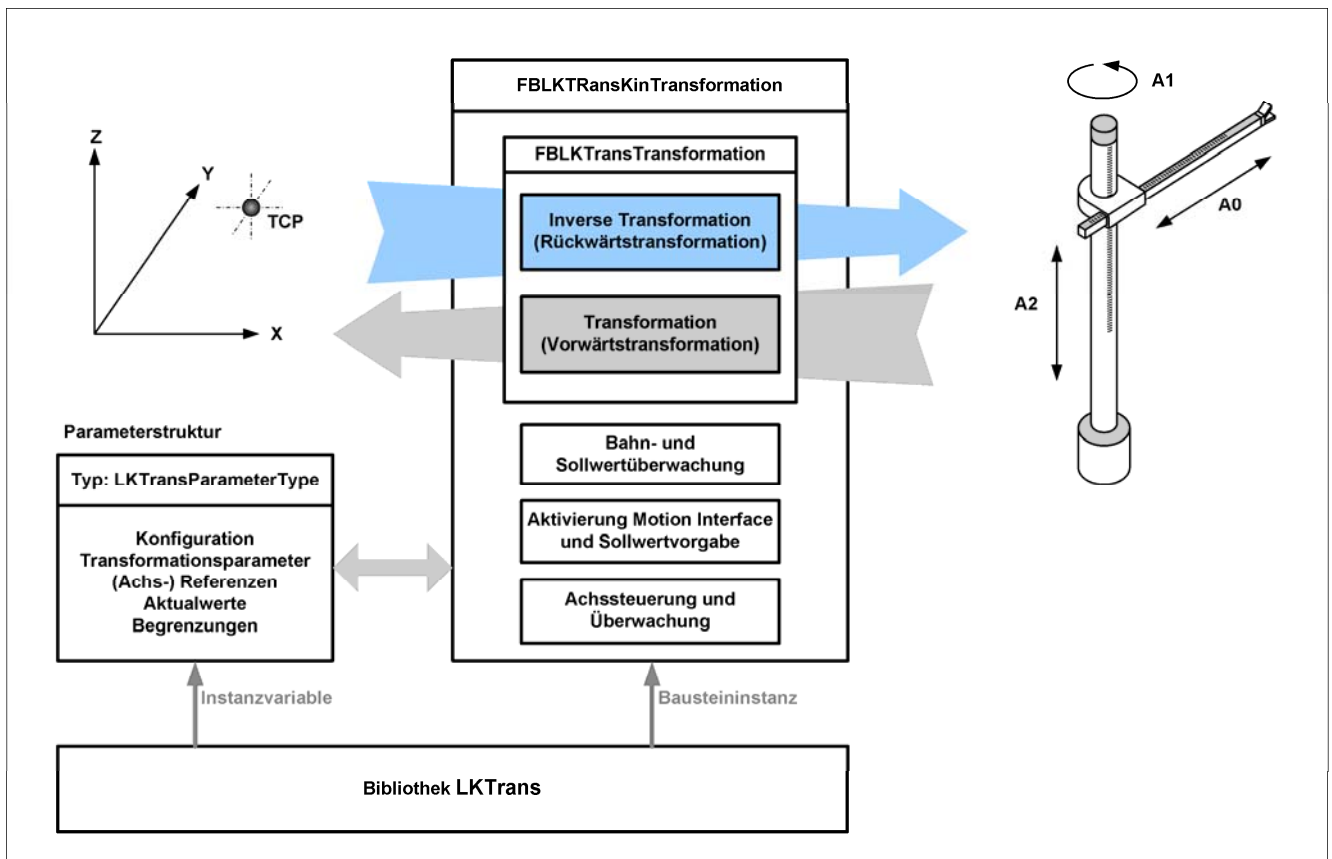


Bild 1-1: Struktur FBLKTransKinTransformation und FBLKTransTransformation

1.2 Abgrenzungen

1.2.1 Verwendbar bei:

- Einfachen, selbst entwickelten Mehrachskinematiken
- Virtuellen Achsen als Eingangsgrößen

1.2.2 Nicht verwendbar bei:

- Ablösender Bewegungsführung an den Ausgangsachsen
- Überlagernder Bewegungsführung an den Ausgangsachsen
- Gleichlaufkopplungen, bei denen die Ausgangsachsen als Master für weitere Achsen verwendet werden

2 Voraussetzungen

2.1 Hardware

Die Bibliothek wurde für SIMOTION entwickelt und kann auf allen Steuerungsausprägungen (SIMOTION C, D oder P) verwendet werden.

2.2 Software

Die Bibliothek wurde mit SIMOTION SCOUT / SIMOTION Runtime V 4.1 SP 2 erstellt und kann ab dieser Version eingesetzt werden. Es sind aktuell jedoch keine Einschränkungen bekannt, die einen Einsatz auf einer Firmwareversion ab V4.0 ausschließen würde (sprachabhängige Erweiterung und ggf. Bahnobjektfunktionalität).

2.3 Projekt

- Die Achsen müssen vollständig parametrisiert und optimiert sein (Lageregelkreis, Drehzahlregelkreis).
- Die beteiligten Achsen müssen im gleichen Takt wie der Funktionsbaustein FBLKTransKintransformation gerechnet werden.
- Die Referenzfahrt referenzierpflichtiger Achsen muss konfiguriert und getestet sein.
- Die Absolutwertgeberjustage an Achsen mit Absolutwertgeber muss erfolgt sein.
- Die Freigabe der Regler ist nicht Bestandteil der hier beschriebenen Bausteine und muss vom Anwender realisiert werden.

3 Allgemeine Funktionsbeschreibung

3.1 Begriffserklärung

3.1.1 Reale Maschinenachsen

Maschinenachsen sind die realen, wirklich vorhandenen Achsen eines Handlinggerätes. Im Zusammenhang mit dem FBLKTransKinTransformation ist nur die Projektierung der Ausgangsachsen als reale Achsen sinnvoll.

3.1.2 Bahnobjekt bzw. Bahninterpolator

Das Bahnobjekt stellt die Funktionalität für die Bahninterpolation von zwei oder drei Bahnachsen und weitere mit der Bahninterpolation verbundene Aufgaben bereit. Es beinhaltet auch die im System implementierten Kinematiktransformationen. Im Zusammenhang mit dem FBLKTransKinTransformation kann mit einem Bahnobjekt die Erzeugung der kartesischen Sollpositionen erfolgen, hierbei wird das Bahnobjekt als kartesisches Portal (CARTESIAN) konfiguriert. Das Bahnobjekt ist jedoch keine Voraussetzung für den Einsatz der hier beschriebenen Bausteine.

3.1.3 Eingangs- und Ausgangsachsen

Der Begriff Eingangsachsen wird im Zusammenhang mit dem FBLKTransKinTransformation für die virtuellen, meist kartesischen Achsen verwendet, die die Positionen X, Y und Z repräsentieren. Ausgangsachsen sind die realen Achsen, deren Motion Interface mit den berechneten Werten der Rücktransformation zyklisch versorgt werden.

3.1.4 Frametransformation

Der Begriff bezeichnet hier die mathematische Funktion zur Ausrichtung des Koordinatensystems. Es können Drehungen und Verschiebungen realisiert werden. Hiermit kann das Eingangskoordinatensystem angepasst werden, bevor es mittels der inversen Transformation in Achskoordinaten umgerechnet wird. Ebenso wird das Koordinatensystem, welches über die Transformation aus den Achskoordinaten berechnet wird, ebenso angepasst um dem Eingangskoordinatensystem zu entsprechen.

3.2 Bibliotheksstruktur

Die Bibliothek LKTrans (**L**ibrary **K**inematik **T**ransformation) wird im Format XML geliefert. Durch Rechtsklick auf den Bibliotheksordner im Projektkonfigurator, kann diese in das Projekt importiert (Menüpunkt: Objekt importieren) werden.

Sie gliedert sich in folgende (Bibliotheks-) Units:

Tabelle 1: Bibliotheksübersicht Datenstrukturen

Unit	Know-How-Schutz	Bedeutung
aVersion	offen	Information über Änderungen und die aktuelle Versionen
cPublic	offen	Die Unit enthält wichtige Konstantendefinitionen (Anzahl Ein- und Ausgangswerte, Fehlernummern)
dProtected	geschützt	Enumeratoren und Typdefinitionen für interne und externe Verwendung.
fKinTrans	geschützt	Enthält den Anwenderfunktionsbaustein FBLKTransKinTransformation , die Funktionen FCLKTransFrameTrans und FCLKTransInvFrametrans , sowie weitere notwendige Hilfsfunktionen, die jedoch nicht extern genutzt werden können.
fTrans	offen	Enthält den Anwenderfunktionsbaustein FBLKTransTransformation in dem die Kinematiktransformation programmiert wird, ebenso notwendige Hilfsfunktionen.

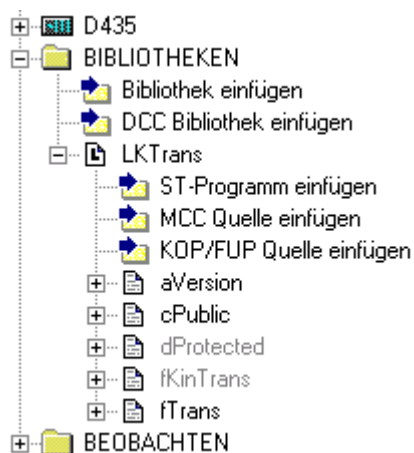


Bild 3-1: Ansicht der Bibliothek im SCOUT Projektnavigator

4 Inbetriebnahme

Die Inbetriebnahme des FBLKTRansKinTransformation wird beispielhaft an der Kinematik 4000, einem Stand-säulenroboter erläutert. Diese Kinematik ist bereits im FBLKTRansTransformation enthalten. Die Integration eigener Kinematiken wird in Kapitel 7 näher erläutert.

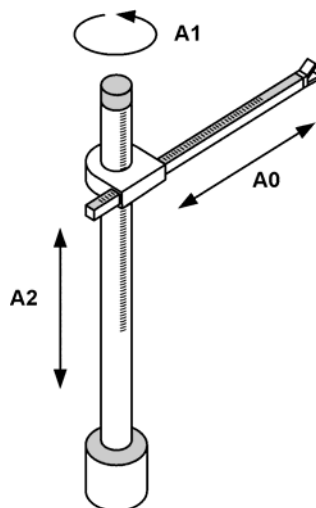


Bild 4-1: Ansicht Beispielkinematik (transld := 4000)

4.1 Voraussetzungen im Projekt

Im Projekt wurden bereits die benötigten Achsen angelegt. Bild 4-2 zeigt beispielhaft sechs projizierte Achsen. Hierbei handelt es sich bei den Achsen *Axis_X*, *Axis_Y* und *Axis_Z* um virtuelle Linearachsen. Die Achsen *AxisOut_0*, *AxisOut_1* und *AxisOut_2* sind als reale Positionierachsen projiziert. Die Achsen *AxisOut_0* und *AxisOut_2* sind als Linearachsen konfiguriert, die Achse *AxisOut_1* als Rundachse mit Modulofunktion 0..360°.

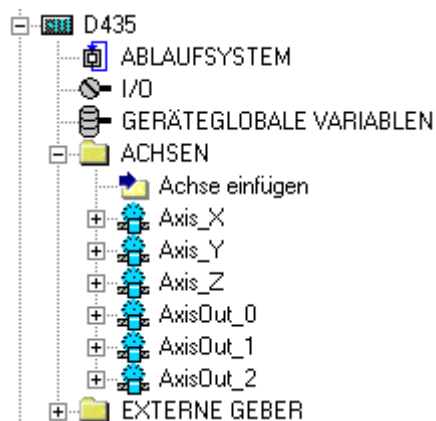


Bild 4-2: Projizierte Achsen im Projektnavigator

Die Bibliothek LKTrans wurde durch Rechtsklick auf den Ordner Bibliotheken im Projektnavigator importiert und auf das jeweilige Zielsystem übersetzt.

4.2 Anpassen der Bibliothekskonstanten

Im ersten Schritt werden die Parameterstruktur und der FBLKTransKinTransformation an die wirklich verwendete Anzahl von Ein- und Ausgangsachsen angepasst. Eine zu groß gewählte Anzahl ist zulässig, aber im Hinblick auf Speicherbedarf und Übersichtlichkeit unzweckmäßig. Die Konstanten können vom Anwender in der Bibliotheksunit *pPublic* angepasst werden. Im dargestellten Beispiel werden 3 Eingangsachsen und 3 Ausgangsachsen verwendet.

Auch der Bearbeitungsstakt des Funktionsbausteins kann hier geändert werden. Dieser muss dem Bearbeitungsstakt der beteiligten Achsen entsprechen. Durch Angabe der Bearbeitungsebene liest der Baustein die eingestellte Zykluszeit aus und berechnet hierraus notwendige Geschwindigkeiten und Beschleunigungen. Der Baustein besitzt einen Ringpuffer in dem Fehlermeldungen gespeichert werden. Die Anzahl Einträge im Puffer kann angepasst werden.

```
//constants for FBLKTransKinTransformation
//-----
//number of used input axes - range: 1..LKTRANS_NUMBER_OF_INPOSITIONS
LKTRANS_NUMBER_OF_USED_INPUT_AXES      : USINT := 3;

//number of used output axes - range: 1..LKTRANS_NUMBER_OF_OUTPOSITIONS
LKTRANS_NUMBER_OF_USED_OUTPUT_AXES     : USINT := 3;

//execution level of fb and technology object: IPO, IPO_2, SERVO - default IPO
LKTRANS_EXECUTIONLEVEL                 : EnumToExecutionLevel := IPO;

//number of items in diagnostic buffer, range: 1..USINT#MAX
LKTRANS_NUMBER_OF_DIAGBUFFER_ITEMS     : USINT := 20;
```

Hinweis:

Die maximale Anzahl entspricht den Konstanten LKTRANS_NUMBER_OF_INPOSITIONS und LKTRANS_NUMBER_OF_OUTPOSITIONS des FBLKTransTransformation. Ggf. sind diese ebenfalls zu erhöhen.

4.3 Beispielaufruf in ST

Im Projekt wird eine neue ST-Programmquelle im Ordner Programme eingefügt.

4.3.1 Anlegen der Parameterstruktur

Die Parameterstruktur wird über einen Ein-/Ausgangsparameter mit dem FBLKTransKinTransformation verbunden und enthält alle notwendigen Einstellungen, Begrenzungen, Soll- und Istwerte.

In folgendem Codebeispiel ist das Anlegen der Parameterstruktur, einer Instanzvariablen vom Typ *sLKTransParameterType* dargestellt. Der Aufbau ist im Kapitel 5.2.1 genauer beschrieben. Der Name kann frei gewählt werden, es wurde hier *gsParameter* gewählt. Mit der Anweisung *USELIB LKTrans* wird die Verbindung zur Bibliothek hergestellt.

```
INTERFACE
//----- Import -----
    USELIB LKTrans; //connection to library
//----- Device Global Variables -----
    VAR_GLOBAL
        //instance of parameter struct
        gsParameter : sLKTransParameterType;
    END_VAR
//-----
END_INTERFACE
```

Nach dem Übersetzen der Unit, stellt sich die Parameterstruktur im SCOUT Symbolbrowser wie in Bild 4-3 dar.

D435.pKinTrans: Symbolbrowser

	Name	Datentyp	Anzeigeformat	Anfangswert
	Alle	Alle	Alle	Alle
1	gsParameter	'sLKTransParameterType'		
2	sGeneral	'sLKTransGeneralType'		
3	sCmdPathTolerance	'sLKTransGeneralCmdPathToleranceType'		
4	sActPathTolerance	'sLKTransGeneralActPathToleranceType'		
5	sActPathFollowingError	'sLKTransGeneralFollowingErrorType'		
6	eMVecCalcMode	'eLKTransMCalculationType'		[0] STANDARD_CALC
7	sTransformation	'sLKTransType'		
8	i32TransID	DINT	DEC	0
9	ar64Parameter	'ARRAY [0..19] OF LREAL'		
10	sCoordTrans	'sLKTransFrameTransType'		
11	asInputAxis	'ARRAY [0..2] OF sLKTransInputAxisType'		
12	asInputAxis[0]	'sLKTransInputAxisType'		
13	asInputAxis[1]	'sLKTransInputAxisType'		
14	asInputAxis[2]	'sLKTransInputAxisType'		
15	asOutputAxis	'ARRAY [0..3] OF sLKTransOutputAxisType'		
16	asOutputAxis[0]	'sLKTransOutputAxisType'		
17	asOutputAxis[1]	'sLKTransOutputAxisType'		
18	asOutputAxis[2]	'sLKTransOutputAxisType'		
19	asOutputAxis[3]	'sLKTransOutputAxisType'		

Bild 4-3: Ansicht der Parameterstruktur im Symbolbrowser

4.3.2 Aufruf des Funktionsbausteins

Im folgenden Codebeispiel ist der Aufruf des Funktionsbausteins in einem Anwenderprogramm (Beispielname pKinTransIpo) dargestellt. Es wird eine Instanz des Bausteins unter dem frei gewählten Namen *myFBKTrans* erstellt. Dem Eingangsparameter *enable* ist die globale Variable *gboKinTransEnable* zugewiesen. Mit diesem kann der Baustein sowie die Berechnung der Transformation aktiviert werden. Dem Eingangsparameter *enableMI* ist die globale Variable *gboKinTransEnableMI* zugewiesen. Mit ihr wird das Motion Interface der Achsen aktiviert und anschließend mit zyklischen Sollwerten versorgt. Der Ein-/ Ausgangsparameter *parameter* ist mit der bereits erstellten Parameterstruktur *gsParameter* verbunden. Die Ausgangsparameter des FB sind im gezeigten Codeabschnitt noch nicht belegt.

```

PROGRAM pKinTransIpo
//-----
VAR
  myFBKTrans : FBLKTransKinTransformation; //instance of function block
END_VAR
//call instance of function block
//-----
myFBKTrans (
  enable := gboKinTransEnable
  ,enableMI := gboKinTransEnableMI
  ,parameter := gsParameter
  // ,valid =>
  // ,state =>
  // ,error =>
  // ,errorId =>
  // ;transErrorId =>
  // ,diagnostics =>
);

```

```

IF myFBKTrans.error THEN
  //error handling
;
END_IF;
END_PROGRAM

```

4.3.3 Taskeinbindung ins Ablaufsystem

Das Programm muss im gleichen Bearbeitungstakt wie die beteiligten Achsen aufgerufen werden. Hierzu ist das Programm im Ablaufsystem der *IPOsynchronousTask* zuzuordnen, wenn an den Achsen als Bearbeitungstakt IPO konfiguriert wurde. Das Programm ist im Ablaufsystem der *IPOsynchronousTask_2* zuzuordnen, wenn an den Achsen als Bearbeitungstakt IPO_2 konfiguriert wurde.

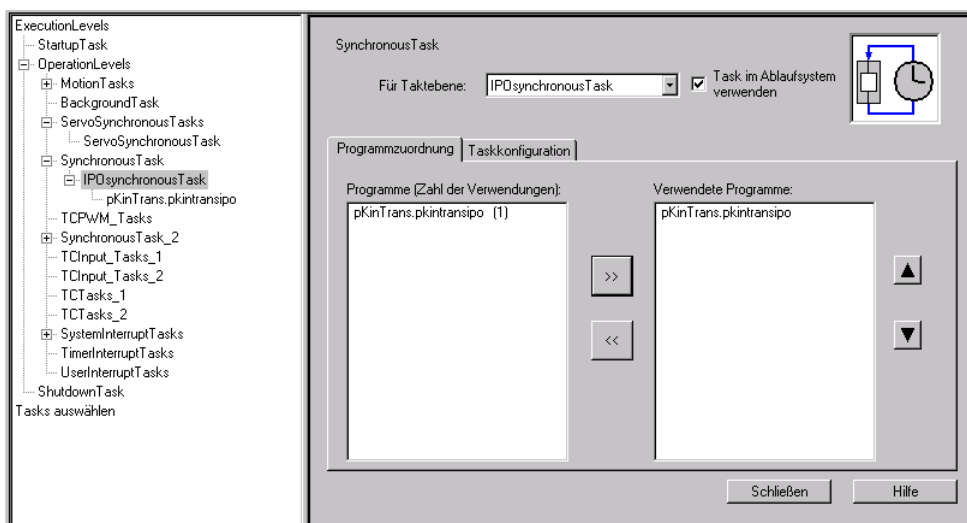


Bild 4-4: Aufruf des Programm im Ablaufsystem

4.3.4 Wertzuweisung Parameterstruktur

Die Parameterstruktur enthält alle relevanten Einstellungen des FBLKTransKinTransformation. Zur Konfiguration müssen hier die notwendigen Werte eingestellt werden. Hierbei kann der Anwender in SIMOTION verschiedene Möglichkeiten nutzen:

- Initialisierungsprogramm in der Startuptask
- Importieren der Werte von CF Karte
- Anlegen der Parameterstruktur als Retainvariable, Onlineänderung von Werten
- Initalisierung im ersten Bearbeitungszyklus einer zyklischen Task

Beispielhaft ist hier die Zuweisung der Werte im ersten Bearbeitungszyklus dargestellt.

```

VAR
  boFirstCycle : BOOL := TRUE; //first cycle in mode run
END_VAR

//initialisation of variables in first cycle
IF boFirstCycle THEN
  //general parameter-----
  //the motionIn will activated direct under this limit , 0.0 = inactive
  gsParameter.sGeneral.sActPathTolerance.r64ActivateMIDirectLimit := 0.1; //mm
  //the (output-) axes will pulled to setpoint under this limit
  //before activating the MI, 0.0 = inactive
  gsParameter.sGeneral.sActPathTolerance.r64ActivateMIPullAxesLimit := 20.0; //mm

```

```

//an active motionIn will deactivated when reach this limit, 0.0 = inactive
gsParameter.sGeneral.sActPathFollowingError.r64Limit := 10.0; //mm
//an active motionIn will deactivated when reach this LIMIT, 0.0 = inactive
gsParameter.sGeneral.sCmdPathTolerance.r64Limit := 1.0 //mm
//calculation mode for motion vector
gsParameter.sGeneral.eMVecCalcMode := STANDARD_CALC;
//transformation parameter-----
//tranformation id - example 4000
gsParameter.sTransformation.i32TransID := 4000;
//mm, minimum length OF actualOutPosition[0]
gsParameter.sTransformation.ar64Parameter[0] := 10.0;
//mm, maximum length of actualOutPosition[0]
gsParameter.sTransformation.ar64Parameter[1] := 2000.0;
//mm, shiftet axis: distance d between axis Z and actualOutPosition[0]
gsParameter.sTransformation.ar64Parameter[2] := 0.0;
//angle in deg for commandOutPosition[1] when X=0.0 and Y=0.0
gsParameter.sTransformation.ar64Parameter[4] := 0.0;
//input axes-----
gsParameter.asInputaxis[0].toAx := Axis_X; //reference
gsParameter.asInputaxis[1].toAx := Axis_Y; //reference
gsParameter.asInputaxis[2].toAx := Axis_Z; //reference
//output axes-----
//axis A0 - liner axis:
gsParameter.asOutputaxis[0].toAx := AxisOut_0; //reference
//limit for commandvalue tolerance
gsParameter.asOutputaxis[0].sCommandValueTolerance.r64Limit := 1.0; //mm
//dynamic limits to open motion in (e.g. axes limits)
gsParameter.asOutputaxis[0].sMIDynamics.r64Velocity := 1000.0; //mm/s
gsParameter.asOutputaxis[0].sMIDynamics.r64Acceleration := 10000.0; //mm/s2
gsParameter.asOutputaxis[0].sMIDynamics.r64Deceleration := 10000.0; //mm/s2
//dynamic limits for Stop, e.g. in case of error
gsParameter.asOutputaxis[0].sStopDynamics.r64Deceleration := 5000.0; //mm/s2
gsParameter.asOutputaxis[0].sStopDynamics.r64Jerk := 10000.0; //mm/s3
//dynamic limits for Stop, e.g. in case of error
gsParameter.asOutputaxis[0].sPullDynamics.r64Velocity := 100.0; //mm/s
gsParameter.asOutputaxis[0].sPullDynamics.r64Acceleration := 1000.0; //mm/s2
gsParameter.asOutputaxis[0].sPullDynamics.r64Deceleration := 1000.0; //mm/s2
gsParameter.asOutputaxis[0].sPullDynamics.r64Jerk := 10000.0; //mm/s3
//axis A1 - rotary axis:
gsParameter.asOutputaxis[1].toAx := AxisOut_1; //reference
gsParameter.asOutputaxis[1].sCommandValueTolerance.r64Limit := 1.0; //°
gsParameter.asOutputaxis[1].sMIDynamics.r64Velocity := 1000.0; //°/s
gsParameter.asOutputaxis[1].sMIDynamics.r64Acceleration := 10000.0; //°/s2
gsParameter.asOutputaxis[1].sMIDynamics.r64Deceleration := 10000.0; //°/s2
gsParameter.asOutputaxis[1].sStopDynamics.r64Deceleration := 5000.0; //°/s2
gsParameter.asOutputaxis[1].sStopDynamics.r64Jerk := 10000.0; //°/s3
gsParameter.asOutputaxis[1].sPullDynamics.r64Velocity := 100.0; //°/s
gsParameter.asOutputaxis[1].sPullDynamics.r64Acceleration := 1000.0; //°/s2
gsParameter.asOutputaxis[1].sPullDynamics.r64Deceleration := 1000.0; //°/s2
gsParameter.asOutputaxis[1].sPullDynamics.r64Jerk := 10000.0; //°/s3
//axis A2 - linear axis:
gsParameter.asOutputaxis[2].toAx := AxisOut_2; //reference
gsParameter.asOutputaxis[2].sCommandValueTolerance.r64Limit := 1.0; //mm
gsParameter.asOutputaxis[2].sMIDynamics.r64Velocity := 1000.0; //mm/s
gsParameter.asOutputaxis[2].sMIDynamics.r64Acceleration := 10000.0; //mm/s2
gsParameter.asOutputaxis[2].sMIDynamics.r64Deceleration := 10000.0; //mm/s2
gsParameter.asOutputaxis[2].sStopDynamics.r64Deceleration := 5000.0; //mm/s2
gsParameter.asOutputaxis[2].sStopDynamics.r64Jerk := 10000.0; //mm/s3
gsParameter.asOutputaxis[2].sPullDynamics.r64Velocity := 100.0; //mm/s
gsParameter.asOutputaxis[2].sPullDynamics.r64Acceleration := 1000.0; //mm/s2
gsParameter.asOutputaxis[2].sPullDynamics.r64Deceleration := 1000.0; //mm/s2
gsParameter.asOutputaxis[2].sPullDynamics.r64Jerk := 10000.0; //mm/s3
//-----
boFirstCycle := FALSE;
END_IF;

```

4.4 Systemfehler deaktivieren

Die externe Transformation nutzt das Motion Interface am Technologieobjekt Achse um die Sollwerte der Transformation an die Achse zu übergeben. Dieses benötigt neben der (Soll-) Position auch die Geschwindigkeit und Beschleunigung. Da die Transformationsgleichung nur die Positionen liefert, muss die Geschwindigkeit und Beschleunigung berechnet werden.

Da diese Berechnung aus performancegründen nur ungenau erfolgt, kann es vorkommen dass das Technologieobjekt Achse folgende Fehler ausgibt:

4002 Die programmierte Geschwindigkeit wird vom System auf die maximal zulaessige Geschwindigkeit begrenzt.

40020 Dynamik der Sollwerte am Motion-Interface (Typ: 2) kann nicht eingehalten werden (Grund: 1) oder (Grund: 2)

Es erfolgt normalerweise keine Reaktionen des Technologieobjektes auf diese Fehler da die Vorbelegung der lokalen Reaktion auf NONE (= keine Reaktion der Achse – nur Information) steht. Es wird empfohlen die Meldungen 40002 und 40020 für alle Ausgangsachsen der Transformation zu deaktivieren und die Überwachungen des Funktionsbausteins (Kapitel 6.2.8) zu nutzen.

Das Deaktivieren von Alarman erfolgt in SIMOTION SCOUT in der Alarmkonfiguration. Diese ist in der TechnologicalFaultTask im Ablaufsystem erreichbar (Siehe Bild 4-5).

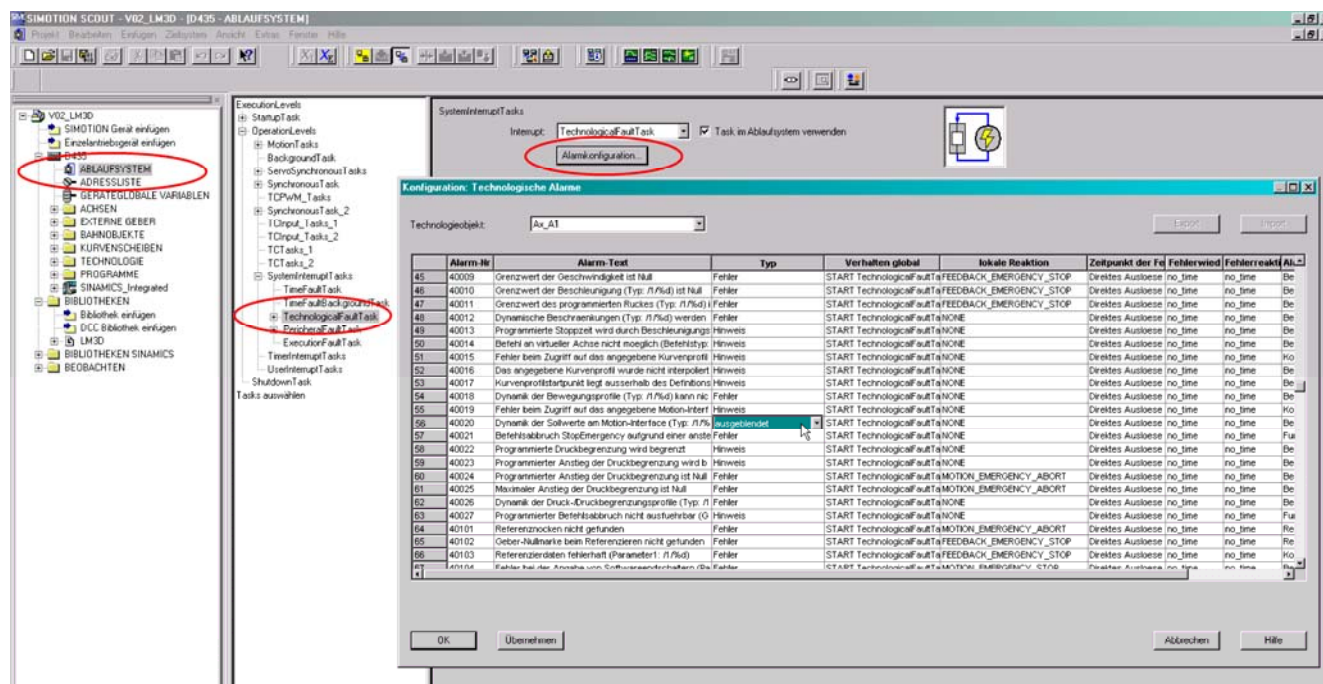


Bild 4-5: Änderung der Alarmreaktion in SIMOTION SCOUT

5 Konstanten und Datenstrukturen

5.1 Globale Konstanten

Mit den folgenden Konstanten kann das Mengengerüst der Funktionsbausteine eingestellt werden (Anzahl Ein- und Ausgänge, Achszahl, usw.). Definiert werden diese Konstanten in der (Bibliotheks-) Unit **cPublic**. In dieser Unit befinden sich auch Konstanten die als Fehlernummern verwendet werden. Diese sind im Kapitel 7 aufgeführt und beschrieben.

Name	Datentyp	Bedeutung
LKTRANS_NUMBER_OF_USED_INPUT_AXES	USINT	Anzahl Eingangsachsen am Baustein FBLKTransKinTransformation. Wertebereich: 1..LKTRANS_NUMBER_OF_INPOSITIONS, Default 3 (z.B. für X,Y,Z)
LKTRANS_NUMBER_OF_USED_OUTPUT_AXES	USINT	Anzahl Ausgangsachsen am Baustein FBLKTransKinTransformation. Wertebereich: 1..LKTRANS_NUMBER_OF_OUTPOSITIONS, Default 4
LKTRANS_EXECUTIONLEVEL	ENUM	Bearbeitungstakt des FB und der beteiligten Achsen. Typ: <i>EnumToExecutionLevel</i> (SIMOTION System-Enum): IPO, IPO_2 oder SERVO
LKTRANS_NUMBER_OF_DIAGBUFFER_ITEMS	USINT	Anzahl von Diagnosepuffereinträgen im Fehlerpuffer. Wertebereich: 1..USINT#MAX, Default 20
LKTRANS_NUMBER_OF_INPOSITIONS	USINT	Anzahl (Eingangs-) Positionen am Baustein FBLKTransTransformation. Wertebereich: 1..USINT#MAX, Default 4
LKTRANS_NUMBER_OF_OUTPOSITIONS	USINT	Anzahl (Ausgangs-) Positionen am Baustein FBLKTransTransformation. Wertebereich: 1..USINT#MAX, Default 10
LKTRANS_NUMBER_OF_TRANS_PARAMETER	USINT	Anzahl Transformationsparameter am Baustein FBLKTransTransformation. Wertebereich: 1..USINT#MAX, Default 20

Erläuterung:

Bei den Konstanten **LKTRANS_NUMBER_OF_INPOSITIONS** und **LKTRANS_NUMBER_OF_OUTPOSITIONS** handelt es sich um die Anzahl der Eingangs- und Ausgangskordinaten des mathematischen Bausteins FBLKTransTransformation. Diese können größer sein, als die Anzahl der wirklich im Projekt verwendeter Achsen, welche durch die Konstanten **LKTRANS_NUMBER_OF_USED_INPUT_AXES** und **LKTRANS_NUMBER_OF_USED_OUTPUT_AXES** bestimmt werden.

5.2 Technologische Parameter

5.2.1 Parameterstruktur

Hierbei handelt es sich um die zentrale Datenschnittstelle des FBLKTransKinTransformation. Dieser besitzt einen Ein- / Ausgangsparameter diesen Typs. Die Struktur gliedert sich in mehrere Bereiche.

sLKTransParameterType - Technologische Parameter

Name	Type	Bedeutung
sGeneral	sLKTransGeneralType	Allgemeine Parameter (Kapitel 5.2.2)
sTransformation	sLKTransType	Transformationsparameter(Kapitel 5.2.3)
asInputAxis	ARRAY[0..n -1] OF sLKTransInputAxisType	Parameter der (kartesischen) Eingangssachsen (Kapitel 5.2.4)
asOutputAxis	ARRAY[0..m -1] OF sLKTransOutputAxisType	Parameter der (realen) Ausgangssachsen (Kapitel 5.2.5)

Hinweis: Die Variable n wird von der Konstante LKTRANS_NUMBER_OF_USED_INPUT_AXES bestimmt. Die Variable m wird von der Konstante LKTRANS_NUMBER_OF_USED_OUTPUT_AXES bestimmt.

5.2.2 Allgemeine Parameter (sLKTransGeneralType)

Dieser Bereich enthält allgemeine Parameter, wie die aktuelle Bahnabweichung, sowie diesbezügliche Begrenzungen.

sLKTransGeneralType – Allgemeine Parameter

Name	Type	Bedeutung
sCmdPathTolerance	sLKTransGeneralCmdPathToleranceType	Parameter der Sollwertabweichung (Kapitel)
sActPathTolerance	sLKTransGeneralActPathToleranceType	Parameter der Istwertabweichung (Kapitel)
sActPathFollowingError	sLKTransGeneralFollowingErrorType	Parameter der kartesischen Schleppabweichung (Kapitel)
eMVecCalcMode	eLKTransMICalculationType	Berechnungsart des Bewegungsvektors (STANDARD_CALC, QUADRATIC_CALC)

5.2.2.1 Parameter der Sollwertabweichung (sLKTransGeneralCmdPathToleranceType)

sLKTransGeneralCmdPathToleranceType – Parameter der Sollwertabweichung

Name	Type	Bedeutung
r64ActualValue	LREAL	Aktuelle Abweichung, wenn Überwachung aktiv.
r64Limit	LREAL	Grenzwert, 0.0 = Überwachung deaktiviert

5.2.2.2 Parameter der Istwertabweichung (sLKTransGeneralActPathToleranceType)

sLKTransGeneralActPathToleranceType – Parameter der Istwertabweichung

Name	Type	Bedeutung
r64ActualValue	LREAL	Aktuelle Abweichung, wenn Überwachung aktiv.
r64ActivateMIDirectLimit	LREAL	Grenzwert, 0.0 = Überwachung deaktiviert Unterhalb der Grenze wird direkt eingekoppelt
r64ActivateMIPullAxesLimit	LREAL	Grenzwert, 0.0 = Überwachung deaktiviert Unterhalb der Grenze werden die (Ausgangs-) Achsen auf den Sollwert positioniert.

5.2.2.3 Parameter der Schleppabweichung (sLKTransGeneralFollowingErrorType)

sLKTransGeneralFollowingErrorType – Parameter der Schleppwertabweichung

Name	Type	Bedeutung
r64ActualValue	LREAL	Aktuelle Abweichung, wenn Überwachung aktiv.
r64Limit	LREAL	Grenzwert, 0.0 = Überwachung deaktiviert

5.2.3 Transformationsparameter (sLKTransTransformationType)

Dieser Bereich enthält sowohl die Transformationsnummer sowie alle Transformationsparameter. Die Parameter werden zur Vorgabe von mechanischen Abmessungen, Offsets oder Gelenkstellvorgaben verwendet. Auch können hier von der Transformation ermittelte Aktualwerte (z.B. Gelenkstellungen) angezeigt werden.

sLKTransTransformationType – Transformationsparameter

Name	Type	Bedeutung
i32TransId	DINT	Transformationsnummer
ar64Parameter	ARRAY[0..n] OF LREAL	Transformationsparameter
sCoordTrans	sLKTransFrameTransType	Frametransformationsparameter (Kapitel 5.2.3.1)

Hinweis: n wird von der Konstanten LKTRANS_NUMBER_OF_TRANS_PARAMETER bestimmt.

5.2.3.1 Parameter der Frametransformationsparameter (sLKTransType)

sLKTransFrameTransType – Frametransformationsparameter

Name	Type	Bedeutung
ar64Translation	ARRAY[0..2] OF LREAL	Koordinaten-Verschiebung
ar64Rotation	ARRAY[0..2] OF LREAL	Koordinaten-Drehung

5.2.4 Parameter der Eingangsachsen (sLKTransInputAxisType)

Dieser Bereich enthält alle Parameter der Eingangsachsen.

sLKTransInputAxisType – Parameter der Istwertabweichung

Name	Type	Bedeutung
r64CommandPosition	LREAL	aktueller Sollwert der Achse
r64ActualPosition	LREAL	Aktueller Istwert aus der Transformation

Name	Type	Bedeutung
toAx	Posaxis	Referenz der Achse, TO#NIL=Inaktiv
i32State	DWORD	Statuswort der Achse
boGetValWithoutAx	BOOL	Auch ohne zugeordneter Achsreferenz wird der aktuelle Sollwert verwendet

5.2.5 Parameter der Ausgangsachsen (sLKTransOutputAxisType)

Dieser Bereich enthält alle Parameter der Ausgangsachsen.

sLKTransOutputAxisType – Parameter der Istwertabweichung

Name	Type	Bedeutung
r64MotionInPosition	LREAL	aktueller Sollwert aus der Transformation
r64CommandPosition	LREAL	aktueller Sollwert der Achse
r64ActualPosition	LREAL	Aktueller Istwert aus der Transformation
toAx	Posaxis	Referenz der Achse
i32State	DWORD	Statuswort der Achse
sCommandValueTolerance	sLKTransOutputAxCmdValueToleranceType	Parameter der Sollwertüberwachung (Kapitel 5.2.5.1)
sMIDynamics	sLKTransOutputAxMIDynamikType	Dynamikgrenzen des Motion Interface MI (Kapitel 5.2.5.2)
sStopDynamics	sLKTransOutputAxStopDynamikType	Dynamikgrenzen bei Stop, z.B. bei Fehler (Kapitel 5.2.5.3)
sPullDynamics	sLKTransOutputAxPullDynamikType	Dynamikgrenzen zum Heranziehen der Achsen bei Aktivierung (Kapitel 5.2.5.4)

5.2.5.1 Parameter der Sollwertüberwachung (sLKTransOutputAxCmdValueToleranceType)

sLKTransOutputAxCmdValueToleranceType – Parameter der Sollwertüberwachung

Name	Type	Bedeutung
r64ActualValue	LREAL	Aktuelle Abweichung, wenn Überwachung aktiv.
r64Limit	LREAL	Grenzwert, 0.0 = Überwachung deaktiviert

5.2.5.2 Dynamikgrenzen des Motion Interface (sLKTransOutputAxMIDynamikType)

sLKTransOutputAxMIDynamikType – Parameter der Sollwertüberwachung

Name	Type	Bedeutung
r64Velocity	LREAL	Maximale Geschwindigkeit
r64Acceleration	LREAL	Maximale Beschleunigung
r64Deceleration	LREAL	Maximale Verzögerung

5.2.5.3 Dynamikgrenzen bei Stop (sLKTransOutputAxStopDynamikType)

sLKTransOutputAxStopDynamikType – Dynamikgrenzen bei Stop

Name	Type	Bedeutung
r64Deceleration	LREAL	Maximale Verzögerung

Name	Type	Bedeutung
r64Jerk	LREAL	Ruckvorgabe

5.2.5.4 Dynamikgrenzen beim (Achs-) Heranziehen (sLKTransOutputAxPullDynamikType)

sLKTransOutputAxPullDynamikType – Dynamikgrenzen beim Heranziehen der Achsen

Name	Type	Bedeutung
r64Velocity	LREAL	Maximale Geschwindigkeit
r64Acceleration	LREAL	Maximale Beschleunigung
r64Deceleration	LREAL	Maximale Verzögerung
r64Jerk	LREAL	Ruckvorgabe

5.2.6 Diagnosestruktur (sLKTransDiagnosticsType)

In der Diagnosestruktur werden Detailinformationen zu aufgetretenen Fehlern gespeichert. Diese unterstützen den Anwender beim Auffinden der Ursache des Fehlers. Siehe hierzu auch Kapitel 8.

sLKTransOutputAxisType – Parameter der Diagnosestruktur

Name	Type	Bedeutung
u8AxNumber	USINT	Betroffener (Achsen-) Index
i32Detail1	DINT	Detailinfo 1
b32Detail2	DWORD	Detailinfo 2
b32Detail3	LREAL	Detailinfo 3
i32BufferIndex	DINT	Index der aktuellen (letzten) Fehlermeldung im Fehlerpuffer, -1=kein eingetragener Fehler
asBuffer	ARRAY[0..n-1] OF sLKTransDiagnosticItem	Fehlerpuffer, realisiert als Ringpuffer (Kapitel 5.2.6.1)

Hinweis: n wird durch die Konstante LKTRANS_NUMBER_OF_DIAGBUFFER_ITEMS bestimmt.

5.2.6.1 Parameter eines Diagnosepuffereintrags (sLKTransDiagnosticItem)

sLKTransDiagnosticItem – Parameter des Diagnosepuffereintrags

Name	Type	Bedeutung
dtPointOfTime	DATE_AND_TIME	Auftrittszeitpunkt des Fehler
b32ErrorId	DWORD	Fehlernummer
u8AxNumber	USINT	Betroffener (Achsen-) Index
i32Detail1	DINT	Detailinfo 1
b32Detail2	DWORD	Detailinfo 2
b32Detail3	LREAL	Detailinfo 3

5.2.7 Statuswörter der Achsen

Am Statuswort **b32state** innerhalb der Achsparameter (Kapitel 5.2.4 und 5.2.5) kann der aktuelle Status der Achse in Form von Einzelbits ausgelesen werden. Zu jeder implementierten Achse ist ein Statuswort vorhanden.

LowWord ->dynamische Stati	HighWord ->statisch
Bit0: Nothalt inaktiv	Bit 16: Gleichlaufinstanz angelegt
Bit1: kein technologischer Achsfehler	Bit 17: Moduloachse angelegt
Bit2: Achse in Regelung	Bit 18 Bahnobjekt angelegt
Bit3: Achse referenziert	Bit 19: Bahnobjekt verschaltet
Bit4: Sollgeschwindigkeit = 0	Bit 20 Bahnbewegung aktiv
Bit5: Synchronisationsstatus	Bit 21 Aufsynchronisiervorgang des Bahnobjekts auf einen Leitwert ist aktiv
Bit6: aktiver Getriebegleichlauf	
Bit7 aktiver Kurvengleichlauf	

Aus Performancegründen werden an den Achsen nur die vom FB benötigten Statusbits aktualisiert. Für die Eingangsachsen ist dies das Bit 4 und für die Ausgangsachsen sind dies die Bits 0,1,2,3 und 4.

5.2.8 Enumeratoren

5.2.8.1 Bausteinzustand (eLKTransStateType)

Am Baustein FBLKTransKinTransformation informiert der Ausgangsparameter *state* über den aktuellen Zustand des Motion Interface. Hierzu wird das Enum vom Typ eLKTransStateType verwendet:

Bezeichner	Wert	Bedeutung
NONE	0	Initialwert nach Laden, Prüfung elementarer Konstanten
INACTIVE	10	Leerlaufzustand, das Motion Interface ist nicht aktiv
REDEFINE_AXES	20	Vorbereitung auf die Aktivierung des Motion Interface: Die Eingangsachsen werden auf die aktuellen Positionswerte der Transformation gesetzt.
PULL_AXES	25	Vorbereitung auf die Aktivierung des Motion Interface: Die Ausgangsachsen werden mittels Positionierung auf den Sollwert der inversen Transformation herangezogen.
ACTIVATING_MI	30	Das Motion Interface wird gerade an allen Ausgangsachsen aktiviert
ACTIVE	50	Das Motion Interface ist an allen Ausgangsachsen aktiviert und wird mit zyklisch Positionswerten der inversen Transformation versorgt
DEACTIVATING_MI	60	Das Motion Interface wird gerade an allen Ausgangsachsen deaktiviert. Dies kann aufgrund der Eingangsparameter oder als Reaktion auf einen Fehler erfolgen.
STOP_PULL_AXES	65	Der Zustand PULL_AXES wurde durch Anwender oder Fehler abgebrochen. Die Positionierung der Achsen wird nun gestoppt.

5.2.8.2 Berechnungsart Transformation (eLKTransCalcModeType)

Dieser Eingangsparameter am FBLKTransTransformation bestimmt was vom Baustein berechnet werden soll: Die Transformation (von ActualOutPosition nach ActualInPosition), die inverse Transformation (von CommandInPosition nach CommandOutPosition) oder beide Richtungen.

Bezeichner	Wert	Bedeutung
BOTH	0	Es werden beide Richtungen berechnet
TRANS	1	Es wird nur die Transformation berechnet
INV_TRANS	2	Es wird nur die inverse Transformation berechnet

5.2.8.3 Berechnungsart Bewegungsvector (eLKTransMICalculationType)

Innerhalb des FBLKTransKinTransformation wird aus der Sollposition der Achse, die aus der inversen Transformation stammt, der komplette Bewegungsvector (Position, Geschwindigkeit, Beschleunigung) berechnet. Die Berechnungsmethode kann in der Parameterstruktur eingestellt werden. Hierzu wird das Enum vom Typ eLKTransStateType verwendet:

Bezeichner	Wert	Bedeutung
STANDARD_CALC	0	Es wird numerisch aus der aktuellen Position und dem letzten Bewegungsvektor abgeleitet.
QUADRATIC_CALC	2	Bestimmung einer Parabel aus dem aktuellen und den letzten zwei Positionen. Hieraus wird der Bewegungsvektor ausgelesen.

6 Beschreibung der Funktionsbausteine

6.1 Funktionsbaustein FBLKTransTransformation

6.1.1 Aufgabe

Der Funktionsbaustein FBLKTRansTransformation stellt einheitliche Schnittstellen zur Umrechnung von Positionswerten zur Verfügung. Im Bereich Handling / Robotik kann hiermit die Umwandlung zwischen den kartesischen Positionswerten X, Y, Z und den (Maschinen-) Achspositionen der verwendeten Kinematik erfolgen. Wird sowohl die Vorwärts- wie auch die Rückwärtstransformation realisiert, berechnet der Baustein den geometrischen Abstand zwischen der vorgegebenen, kartesischen Sollposition und der transformierten, kartesischen Istposition.

Der Anwender kann so spezifische Kinematiken entwickeln, die nicht vom Technologieobjekt Bahninterpolation unterstützt werden. Exemplarisch sind bereits mehrere Kinematiken realisiert. Die Anzahl von Eingangs- und Ausgangspositionen sowie die Anzahl von Transformationsparametern kann mittels Bibliothekskonstanten eingestellt werden. Der FB kann auch im Anwenderprogramm als Auskunftsfunktion genutzt werden.

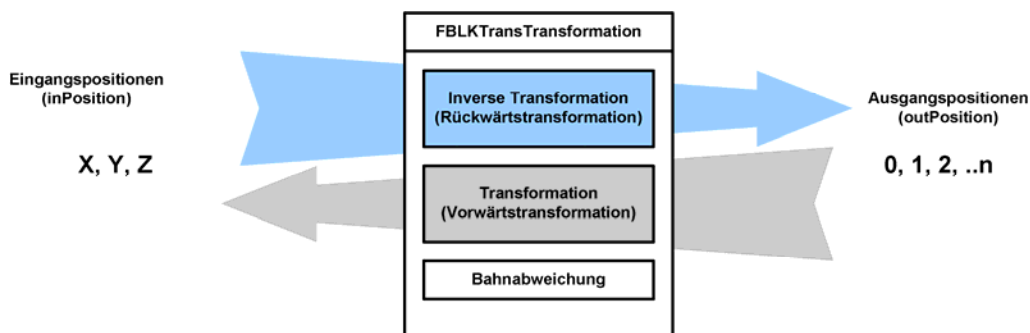


Bild 6-1: Aufgabe des FBLKTransTransformation

6.1.2 Taskeinbindung

Die Einbindung ist prinzipiell in allen Tasks möglich. Werden jedoch Sollwerte mit dem Baustein generiert, erfolgt der Aufruf typischerweise in einer zyklischen, ggf. synchronen Task.

6.1.3 Schematische KOP/FUP-Darstellung

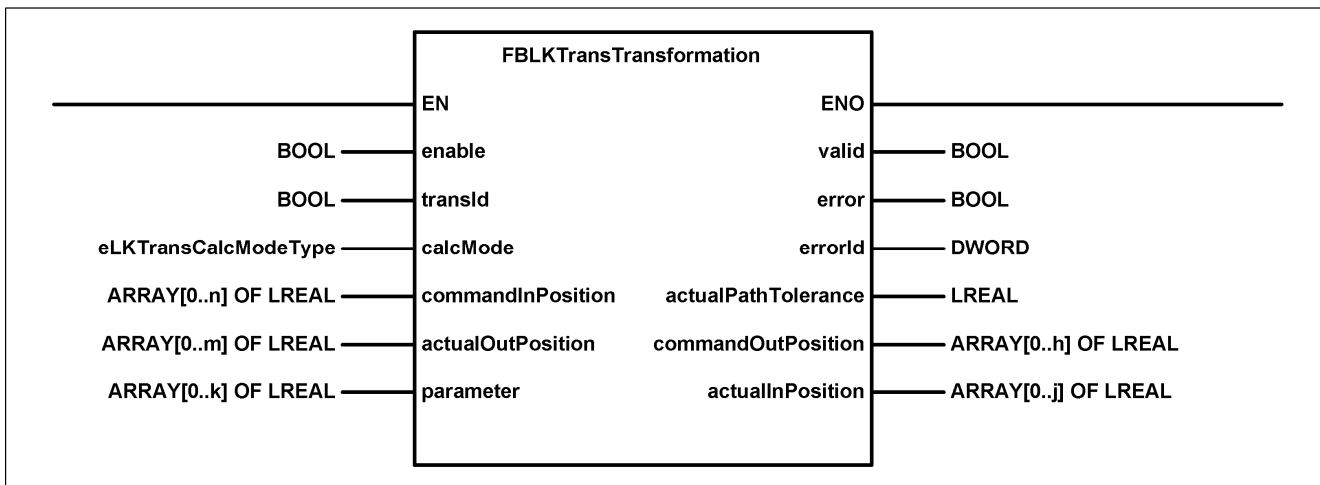


Bild 6-2: KOP-Ansicht des FBLKTransTransformation

6.1.4 Parameter des Bausteins

Name	P-Typ	Datentyp	Bedeutung
enable	IN	BOOL	Entscheidet über Abarbeitung des Bausteins
transId	IN	DINT	Technologiestruktur (siehe Kapitel 5.2.1)
calcMode	IN	eLKTransCalcModeType	Berechnungsart / Richtung: TRANS, INV_TRANS, BOTH
commandInPosition	IN	ARRAY[0..n] OF LREAL	(Soll-) Positionen vor inverser Transformation
actualOutPosition	IN	ARRAY[0..m] OF LREAL	(Ist-) Positionen vor Transformation
parameter	IN_OUT	ARRAY[0..k] OF LREAL	Transformationsparameter
valid	OUT	BOOL	Ausgangswerte sind gültig
error	OUT	BOOL	Anstehender Fehler
errorId	OUT	DWORD	Fehlernummer
actualPathTolerance	OUT	LREAL	Aktuelle geometrische Abweichung (von commandInPosition[0,1,2] und actualInPosition[0,1,2])
commandOutPosition	OUT	ARRAY[0..h] OF LREAL	(Soll-) Positionen nach inverser Transformation
actualInPosition	OUT	ARRAY[0..j] OF LREAL	(Ist-) Positionen nach Transformation

Hinweise:

- n wird bestimmt von der Konstante LKTRANS_NUMBER_OF_INPOSITIONS
- m wird bestimmt von der Konstante LKTRANS_NUMBER_OF_OUTPOSITIONS
- k wird bestimmt von der Konstante LKTRANS_NUMBER_OF_TRANS_PARAMETER
- h wird bestimmt von der Konstante LKTRANS_NUMBER_OF_OUTPOSITIONS
- j wird bestimmt von der Konstante LKTRANS_NUMBER_OF_INPOSITIONS

6.1.5 Funktionsbeschreibung

Durch setzen des Eingangsparameters *enable* wird die Bearbeitung des Funktionsbausteins aktiviert. Dies wird durch den gesetzten Ausgangsparameter *valid* signalisiert. Die Transformationsgleichung wird nun berechnet und kann überprüft werden.

Mit dem Eingangsparameter *transId* kann die Nummer der Transformation gewählt werden. Mit dem Eingangsparameter *calcMode* kann die Berechnungsart vorgegeben werden: Mit der Auswahl TRANS wird nur die Transformation, mit INV_TRANS wird nur die inverse Transformation gerechnet. Bei Anwahl von BOTH werden beide Richtungen, sowie die Bahnabweichung berechnet.

Der Eingangsparameter *commandInPosition* besteht aus einem Array von Positionswerten, welche von der Rücktransformation verarbeitet werden. Die Ergebnisse stehen am Ausgangsparameter *commandOutPosition*, ebenfalls ein Array aus Positionswerten, zur Verfügung

Der Eingangsparameter *actualOutPosition* besteht aus einem Array von Positionswerten, welche von der Vorwärtstransformation verarbeitet. Die Ergebnisse stehen am Ausgangsparameter *actualInPosition*, ebenfalls ein Array aus Positionswerten, zur Verfügung.

Die ersten drei Arrayindizes ([0], [1] und [2]) der Parameter *commandInPosition* und *actualInPosition* werden als kartesische Koordinaten X,Y,Z interpretiert. Zwischen den Parametern wird der Abstand (als Betrag) berechnet und am Ausgangsparameter *actualPathTolerance* zur Verfügung gestellt.

Bei Auftreten eines Fehlers (z.B. ungültige Transformationsnummer) wird der Ausgangsparameter *error* gesetzt und eine Fehlernummer am Parameter *errorId* ausgegeben. Je nach Transformationsgleichung können unterschiedliche Fehler (z.B. undefinierte Wertebereiche) auftreten.

Wichtiger Hinweis:

Es obliegt dem Ersteller einer Transformationsgleichung alle möglichen Fehler abzufangen und zu melden. Werden Rechenfehler, wie Division durch Null nicht abgefangen, kann es zu einem Bearbeitungsfehler in der Steuerung kommen, welche hierdurch in den Betriebszustand STOP wechselt. Je nach konfigurierter Reaktion der Antriebe können hierdurch Maschinenschäden auftreten.

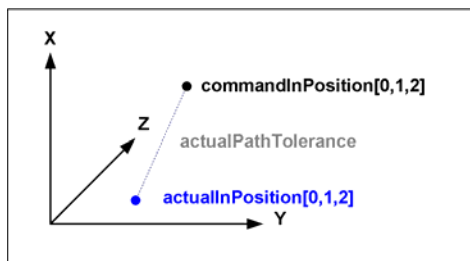


Bild 6-3: Bahnabweichung

6.2 Funktionsbaustein FBLKTransKinTransformation

6.2.1 Aufgabe

Mit dem Funktionsbaustein FBLKTransKinTransformation kann eine Kinematiktransformation zwischen (kartesischen) Eingangsachsen und (realen) Ausgangsachsen projiziert werden. Für die Berechnung wird hierzu der FBLKTransTransformation (Kapitel 6.1) verwendet und zyklisch aufgerufen. Der Funktionsbaustein liest die Sollwerte der konfigurierten Eingangsachsen, die den kartesischen Koordinaten X, Y und Z entsprechen, ein. Aus diesen Sollwerten werden mittels Rückwärtstransformation (inverse Transformation) Sollwerte berechnet und bei aktivem Baustein (enable := TRUE; enableMI := TRUE) in das Motion Interface der realen Maschinenachsen geschrieben. Die Istwerte dieser Ausgangsachsen werden vom Baustein transformiert und stehen zur Information und für Überwachungen zur Verfügung.

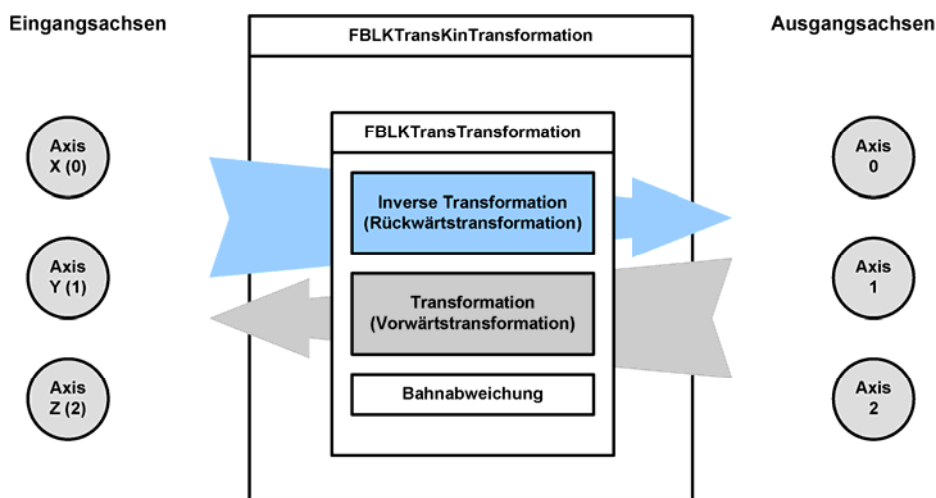


Bild 6-4: Anwendung des FBKintranformation

6.2.2 Taskeinbindung

Der Baustein muss im gleichen Bearbeitungstakt wie die beteiligten Achsen aufgerufen werden. Hierzu ist das Programm im Ablaufsystem der *IPOsynchronousTask* zuzuordnen, wenn an den Achsen als Bearbeitungstakt IPO konfiguriert wurde. Das Programm ist im Ablaufsystem der *IPOsynchronousTask_2* zuzuordnen, wenn an den Achsen als Bearbeitungstakt IPO_2 konfiguriert wurde. Der Aufruf in der *ServosynchronousTask* ist aus Performancegründen nicht sinnvoll und wird nicht empfohlen. Zusätzlich muss mittels der Konstanten *LKTRANS_EXECUTIONLEVEL* (siehe Kapitel 5.1) dem FB mitgeteilt werden, in welcher Task er aufgerufen wird.

6.2.3 Schematische KOP/FUP-Darstellung

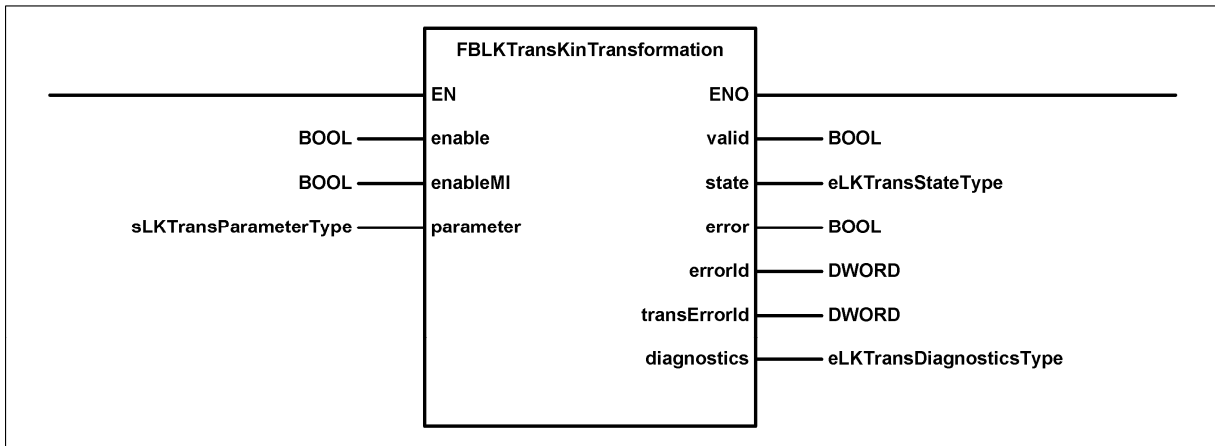


Bild 6-5: KOP-Ansicht des FBLKTransKinTransformation

6.2.4 Parameter des Bausteins

Name	P-Typ	Datentyp	Bedeutung
enable	IN	BOOL	Aktivierung der Bearbeitung und Transformationsberechnung
enableMI	IN	BOOL	Aktiviert das Motion Interface und versorgt dieses mit zyklischen Sollwerten
parameter	IN_OUT	sLKTransParameterType	Parameterstruktur (siehe Kapitel 5.2.1)
valid	OUT	BOOL	Gültiger Bausteinzustand
state	OUT	eLKTransStateType	Zustand des Motion Interface (siehe Kapitel 5.2.8.1)
error	OUT	BOOL	Anstehender Bausteinfehler
errorId	OUT	DWORD	Fehlernummer (siehe Kapitel 8)
transErrorId	OUT	DWORD	Fehlernummer der Transformation
diagnostics	OUT	sLKTransDiagnosticsType	Diagnosestruktur (siehe Kapitel 5.3.3)

Hinweis: Der Ausgangsparameter *transErrorId* liefert die anstehende Fehlernummer des enthaltenen FBLKTransTransformation. Dies wird jedoch nur im aktivierten Zustand des Motion Interface als Fehler betrachtet.

So kann in der Inbetriebsetzungsphase mit inaktivem Motion Interface der fehlerfreie Betrieb der Transformation geprüft werden.

6.2.5 Ablaufdiagramm FBLKTransKinTransformation

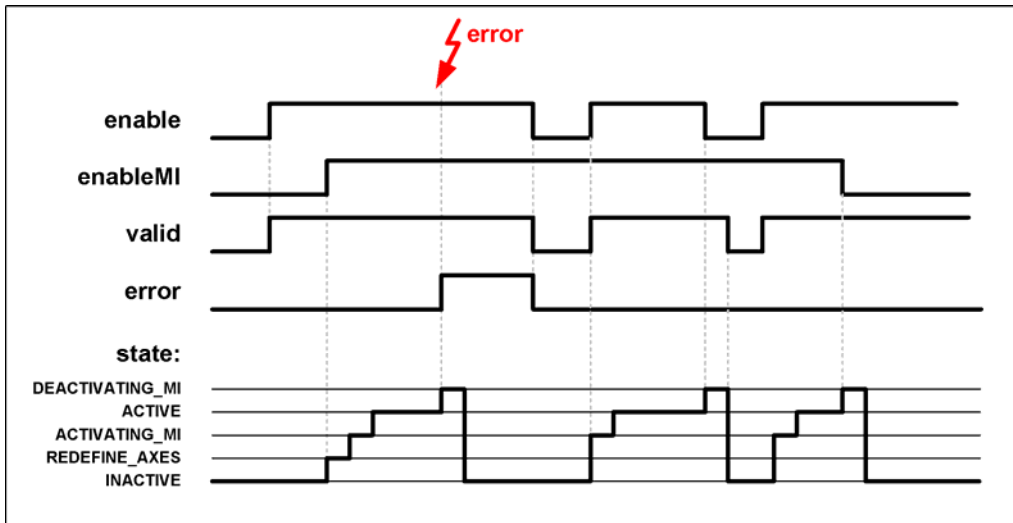


Bild 6-6: Ablaufdiagramm FBLKTransKinTransformation

Tritt bei der Bearbeitung ein Fehler auf, wird das Motion Interface deaktiviert. Um einen anstehenden Fehler zu Quittieren, muss der Eingangsparameter *enable* zurückgesetzt werden.

6.2.6 Zustandsmodell FBLKTransKinTransformation

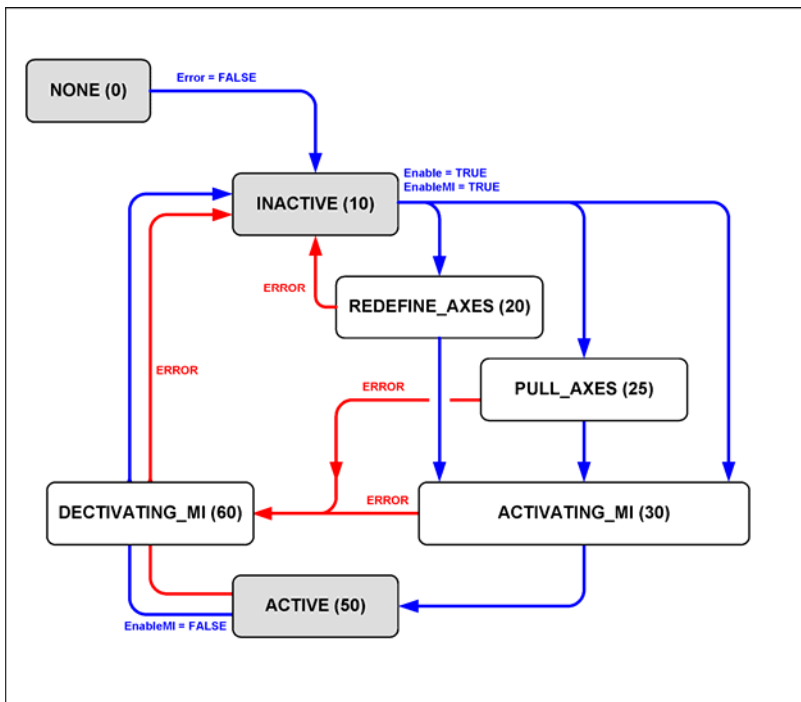


Bild 6-7: Zustandsmodell FBLKTransKinTransformation

Der aktuelle Zustand kann am Ausgangsparameter *state* des FBLKTransKinTransformation ausgelesen werden. Hierbei handelt es sich um ein Enum vom Typ *eLKTransStateType* (Kapitel 5.2.8.1). Der Baustein wechselt nach dem ersten Aufruf vom Initialzustand NONE nach INACTIVE, sofern nicht ein Fehler bei den Konstantenwerten erkannt wird.

6.2.7 Aktivierung Motion Interface

Die Berechnung der Transformation sowie das zyklische Einlesen von Achswerten sind bereits aktiv, wenn der Eingangsparameter *enable* gesetzt ist.

Zur Aktivierung des Motion Interface muss zusätzlich der Eingangsparameter *enableMI* gesetzt werden.

Werden alle Voraussetzungen erfüllt, beginnt der Funktionsbaustein mit der Aktivierung, andernfalls wird ein Fehler ausgegeben. Folgende Voraussetzungen müssen erfüllt sein:

- Die Ausgangsachsen sind referenziert
- An den Ausgangsachsen steht weder ein schwerwiegender Fehler noch ein Not-Stopp an.
- Die Ein- und Ausgangsachsen stehen still
- Die Regler der Ausgangsachsen sind freigegeben

Bei der Aktivierung wird nun vom Baustein überprüft ob das Motion Interface direkt aktiviert werden kann. Dies sollte nur erfolgen wenn die Istposition nur wenig von der Sollposition abweicht. Im Parameter *sGeneral.sActPathTolerance.r64ActivateMIDirectLimit* der Parameterstruktur wird angegeben, wie groß die Bahnabweichung (Abstand von kartesischer Sollposition zur kartesischen Istposition) maximal sein darf. Wird als Limit 0.0 angegeben erfolgt keine direkte Aktivierung.

Wird die Grenze unterschritten, wird das Motion Interface aktiviert, resultierende Ausgleichsbewegungen werden mit den Dynamiken *.asOutputAxis[.].sMIDynamics* der Parameterstruktur ausgeglichen.

Wird die Grenze überschritten oder ist die Grenze deaktiviert erfolgt die Prüfung ob die Ausgangsachsen mittels Positionierung auf den Sollwert herangezogen werden können.

Im Parameter *sGeneral.sActPathTolerance.r64ActivateMIPullAxesLimit* der Parameterstruktur wird angegeben, wie groß die Bahnabweichung beim Aktivieren maximal sein darf. Wird als Limit 0.0 angegeben ist die Grenze deaktiviert. Wird die Grenze unterschritten werden die Achsen mit den Dynamiken *.asOutputAxis[.].sPullDynamics* der Parameterstruktur auf den Sollwert positioniert.

Wurden beide Grenzen überschritten, oder sind diese deaktiviert, werden die Eingangsachsen auf die wirklichen Positionen gesetzt (Systemfunktion *_redefinePosition*).

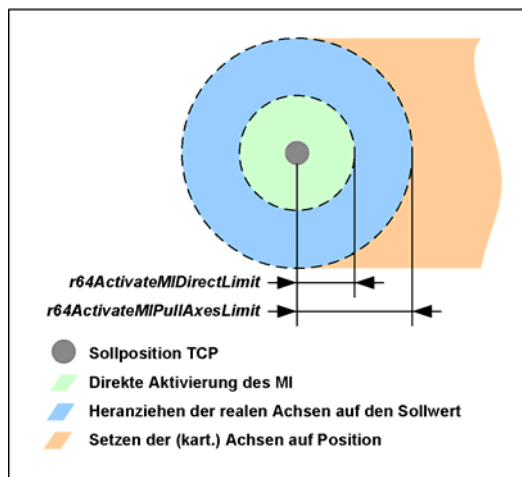


Bild 6-8 Grenzen beim Aktivieren des Motion Interface

6.2.7.1 Beispiel

Wenn es sich bei den Eingangsachsen um virtuelle Achsen handelt, zeigen diese nach dem Hochlauf die Position 0.0 an. Die realen Ausgangsachsen stehen jedoch auf einer realen Position. Bild 6-9 verdeutlicht diesen Zustand (zur Vereinfachung erfolgt keine Umrechnung der Positionen in der Transformation).

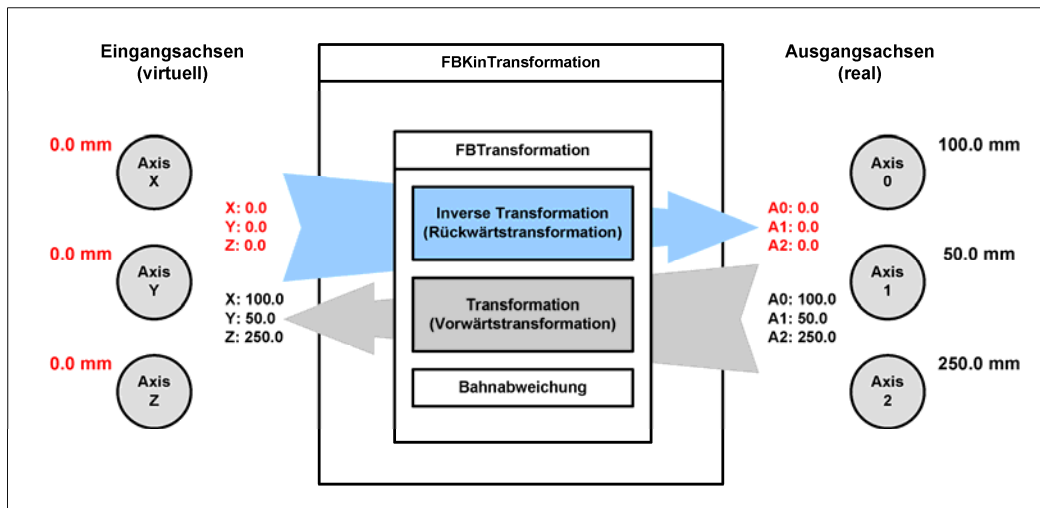


Bild 6-9: Typischer Initialzustand nach Hochlauf

Würde das Motion Interface nun direkt aktiviert, würden die realen Ausgangsachsen diesen Sollwertsprung durch eine unkoordinierte Bewegung mit ihrer maximalen Dynamik ausgleichen. Werden in diesem Zustand die zwei einstellbaren Grenzen überschritten, werden die Positionen der Eingangssachsen auf die (realen) transformierten Positionen gesetzt. Hierdurch liefert nun auch die Rücktransformation die zu den realen Achsen passenden Werte. Dieser Zustand ist in Bild 6-10 dargestellt. Das Motion Interface kann nun ohne Ausgleichsbewegung aktiviert werden.

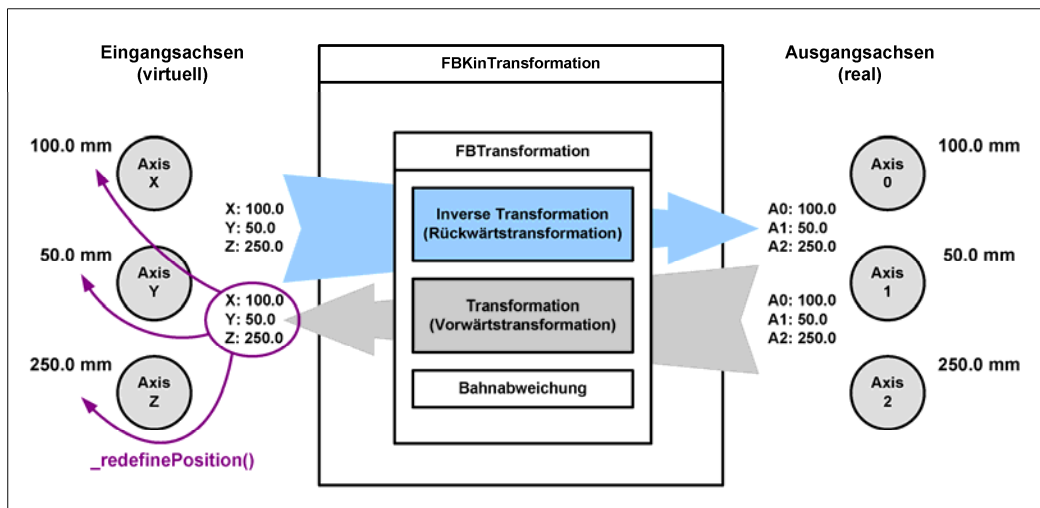


Bild 6-10: Setzen der Eingangssachspositionen

Tritt kein Fehler auf, wechselt der FB in den Zustand ACTIVE und überwacht von nun an das Motion Interface. Die Bahnsollwertüberwachung, Bahnschleppfehlerüberwachung und Achssollwertüberwachung sind nun aktiv. In diesem Zustand kann mit der Bewegung der Eingangssachsen, z.B. mit einem Bahnobjekt begonnen werden.

6.2.8 Überwachungen

Der Funktionsbaustein FBLKTransKinTransformation() überwacht den ordnungsgemäßen Betrieb. Hierzu werden unterschiedliche Kennwerte berechnet und mit Grenzwerten überwacht.

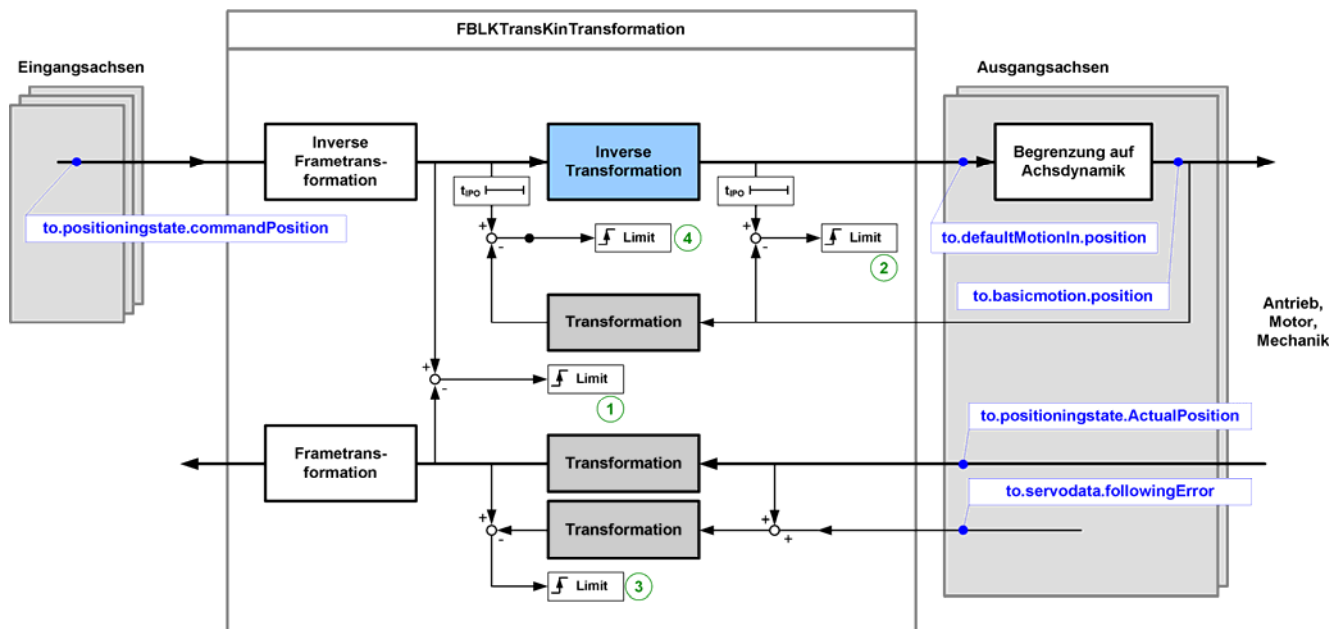


Bild 6-11: Übersicht FBLKTRansKinTransformation mit Überwachungen

Folgende Überwachungen sind im FB realisiert:

- Überwachung der aktuellen Bahnabweichung (im Bild 6-11 mit 1 gekennzeichnet, Details werden in Kapitel 6.2.8.1 beschrieben)
- Überwachung der Bahnsollwertabweichung (im Bild 6-11 mit 4 gekennzeichnet, Details werden in Kapitel 6.2.8.2 beschrieben)
- Überwachung des Bahnschleppabstand (im Bild 6-11 mit 3 gekennzeichnet, Details werden in Kapitel 6.2.8.3 beschrieben)
- Überwachung des Achssollwertes (im Bild 6-11 mit 2 gekennzeichnet, Details werden in Kapitel 6.2.8.4 beschrieben)

6.2.8.1 Bahnabweichung

Die Bahnabweichung ermittelt die Abweichung zwischen aktueller, kartesischer Soll- und Istposition. Die aktuelle Abweichung wird unter *sGeneral.sActPathTolerance.r64ActualValue* in der Parameterstruktur ausgegeben. Da sich beim Technologieobjekt Achse jedoch prinzipbedingt ein Zeitversatz zwischen Sollwert und zurückgelieferter Istposition der Achse ergibt (Bussystem + Interpolationstakt) entspricht dieser Wert nicht der tatsächlichen Abweichung am TCP zum gleichen Zeitpunkt. Es ergibt sich eine geschwindigkeitsabhängige Abweichung.

Daher wird diese Abweichung nur im Stillstand bei der Aktivierung der Transformation verwendet. Details hierzu sind bereits im Kapitel 6.2.7 beschrieben.

6.2.8.2 Bahnsollwertabweichung

Zur Ermittlung der Bahnsollwertabweichung werden die begrenzten Achspositionen transformiert und mit der Sollposition des letzten Aufrufs verglichen. Die aktuelle Abweichung wird unter *sGeneral.sCmdPathTolerance.r64ActualValue* in der Parameterstruktur ausgegeben, wenn die Überwachung aktiv ist. Der Grenzwert wird unter *sGeneral.sCmdPathTolerance.r64Limit* angegeben, ein Wert von 0.0 deaktiviert die Überwachung. Wird die Grenze überschritten, während sich der Baustein im Zustand ACTIVE befindet wird ein

Fehler ausgegeben. Aufgrund des Fehlers wechselt der Baustein in den Zustand DEACTIVATING_MI und die Ausgangsachsen werden mit den vorgegebenen Dynamiken (Details siehe Kapitel 5.2.5) einzeln gestoppt.

Eine Abweichung ergibt sich, wenn Transformation und inverse Transformation nicht zueinander passen (Fehler in der Transformationsgleichung), oder wenn das Technologieobjekt Achse den geforderte Bewegung nicht folgen kann und diese begrenzt.

6.2.8.3 Bahnschleppabstand

Der Bahnschleppabstand wird dadurch ermittelt, dass neben der Transformation der aktuellen Achsposition auch die zeitlich zugehörige Sollposition gebildet und transformiert wird. Die Sollposition wird durch Addition des Schleppabstandes und der aktuellen Position gebildet. Die aktuelle Abweichung wird unter *sGeneral.sActPathFollowingError.r64ActualValue* in der Parameterstruktur ausgegeben, wenn die Überwachung aktiv ist. Der Grenzwert wird unter *sGeneral.sActPathFollowingError.r64Limit* angegeben, ein Wert von 0.0 deaktiviert die Überwachung. Wird die Grenze überschritten, während sich der Baustein im Zustand ACTIVE befindet wird ein Fehler ausgegeben. Aufgrund des Fehlers wechselt der Baustein in den Zustand DEACTIVATING_MI und die Ausgangsachsen werden mit den vorgegebenen Dynamiken (Details siehe Kapitel 5.2.5) einzeln gestoppt.

Eine größere Abweichung ergibt sich wenn die Leistungsgrenze der Antriebe erreicht ist oder wenn die Achs- und Antriebsregler ungenügend eingestellt sind.

6.2.8.4 Sollwertüberwachung Achse

An den realen Ausgangsachsen kann eine Sollwertüberwachung getrennt für jede Achse aktiviert werden. Der Funktionsbaustein FBLKTransKinTransformation schreibt die zyklischen Sollwerte der Rücktransformation in das Motion Interface (MotionIn) der Achse. Überschreiten diese Werte die projektierten, maximalen Dynamikwerte des Motion Interface werden sie auf diese begrenzt. Hierbei wird zwangsläufig von der vorgegebene Bahn abgewichen. Das Technologieobjekt Achse meldet nun den Alarm 40020, was jedoch standardmäßig keine Fehlerreaktion hervorruft. Auch die Schleppabstandsüberwachung der Achse greift hier nicht, da diese ausschließlich den Soll- und Istwert am Lageregler vergleicht.

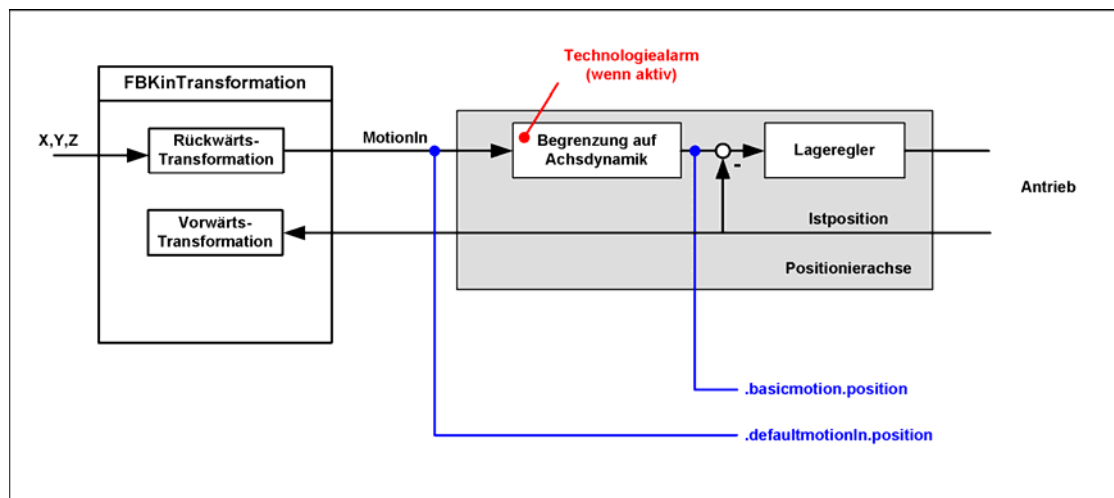


Bild 6-12: Struktur Sollwertüberwachung

Die Sollwertüberwachung des Bausteins vergleicht die Systemvariablen *(axis).defaultmotionIn.position* und *(axis).basicmotion.position* miteinander. Der Versatz von einem IPO-Takt wird berücksichtigt. Übersteigt die Differenz die projektierte Grenze im Zustand ACTIVE, wird ein Fehler ausgegeben, alle Achsen einzeln gestoppt und der Baustein deaktiviert (Details siehe Kapitel 5.2.5).

Die Grenze wird im Parameter *asOutputaxis[n].sCmdValueTolerance.r64Limit* eingestellt. Ein Wert von 0.0 führt zur Deaktivierung der Überwachung an dieser Achse.

6.2.9 Einstellung der Dynamikparameter

Folgende Staffelung der dynamischen Grenzen ist bei der Verwendung des FBLKTransKinTransformation zu beachten.

Die konfigurierte Achsbegrenzungen (der realen Positionierachse) sollten nicht überschritten werden. Die Geschwindigkeitsgrenze wird an der Konfigvariablen *typeOfAxis.maxVelocity.maximum* und der Systemvariablen *plusLimitsOfDynamics.velocity* (ebenso *minusLimitsOfDynamics.velocity*) eingestellt – wirksam ist hier der aktuell geringere Wert. Gleiches gilt für die Beschleunigung mit den Variablen *typeOfAxis.maxAcceleration.maximum* und *plus- / minusLimitsOfDynamics.positiveAccel*, (ebenso *plus- / minusLimitsOfDynamics.negativeveAccel*).

Das Motion Interface kann maximal mit den Achsbegrenzungen aktiviert werden. Hierzu sind gleiche oder kleinere Werte in den Parametern *asOutputAxis[n].sMIDynamics.r64Velocity*, *asOutputAxis[n].sMIDynamics.r64Acceleration* und *asOutputAxis[n].sMIDynamics.r64Deceleration* in der Parameterstruktur anzugeben. Überschreiten die Werte die an der Achse eingestellten Werte wird vom System der Alarm 40020 bei Aktivierung des Motion Interface ausgegeben und alle zyklischen Positionsvorgaben entsprechend begrenzt. Der Fehler steht dauerhaft an, der Betrieb wird jedoch nicht beeinträchtigt.

Die zyklischen Positionswerte und die hieraus resultierende Geschwindigkeit und Beschleunigung sollten weder die Achsbegrenzungen noch die Maximalwerte des Motion Interface überschreiten. Passiert dies trotzdem, wird ebenfalls der Alarm 40020 ausgegeben und die Werte begrenzt.

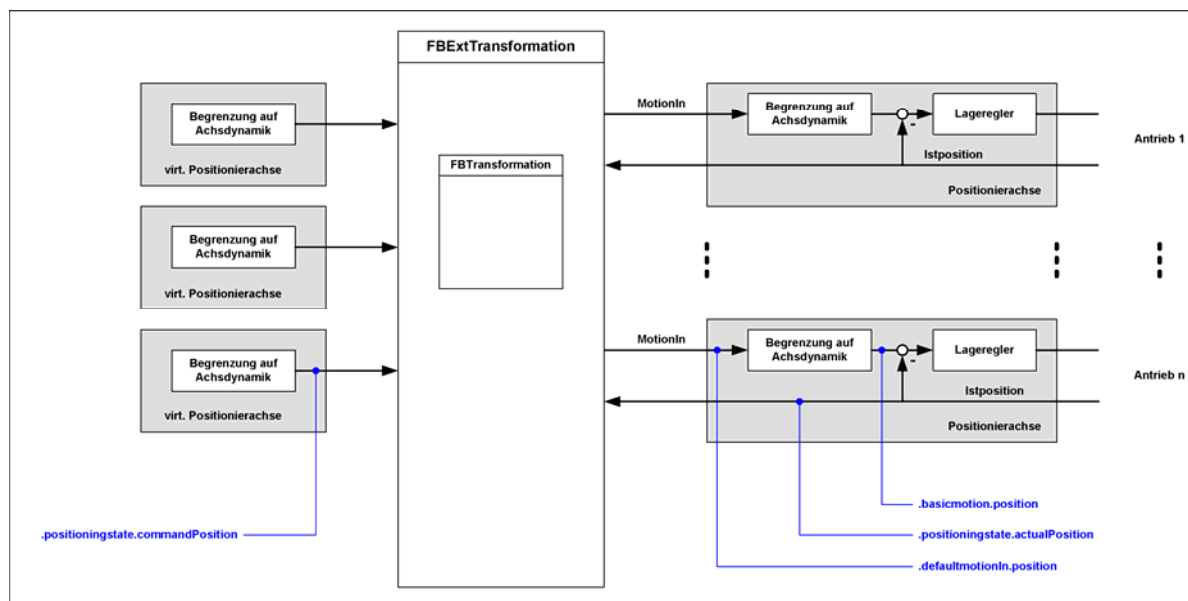


Bild 6-13: Dynamische Begrenzungen bei Einsatz des FBKintranformation

6.2.10 Beispiel Dynamiken und Überwachungen

Dynamik einer Ausgangsachse

Das Motion Interface soll mit den maximalen Dynamiken betrieben werden.

Daher werden die projektierten Grenzen der Achse auch als Dynamikgrenzen für das Motion Interface verwendet. Werte oberhalb der Achsgrenzen sind nicht zulässig.

Begrenzung	Dynamische Grenzen			
	Geschwindigkeit	Beschleunigung	Verzögerung	Ruck
Technologieobjekt Achse (Begrenzung Technologieobjekt)	2000.0 mm/s	20.000 mm/s ²	20.000 mm/s ²	200.000 mm/s ³
sMIDynamics	2000.0 mm/s	20.000 mm/s ²	20.000 mm/s ²	-
sStopDynamics	-	-	10.000 mm/s ²	200.000 mm/s ³
sPullDynamics	100.0 mm/s	1.000 mm/s ²	1.000 mm/s ²	10.000 mm/s ³

Die Dynamiken mit denen im Fehlerfall gestoppt wird müssen unterhalb der Achsgrenzen liegen.

Die Dynamiken, mit denen die Achsen bei Abweichungen herangezogen werden deutlich niedriger gewählt.

Überwachung Achse

Eine Begrenzung durch die Achse im laufenden Betrieb soll überwacht werden. Es sollen nur sehr kleine Abweichungen geduldet werden, daher wird die Grenze klein eingestellt.

gsParameter.asOutputaxis[..].sCommandValueTolerance.r64Limit := 0.5; //mm

Überwachung bei Aktivierung des Motion Interface

Bei Aktivierung des Motion Interface soll nur bei sehr kleinen Abweichungen des TCP eine direkte Aktivierung erfolgen, da resultierende Ausgleichsbewegungen mit den hohen Dynamiken des Motion In erfolgen.

gsParameter.sGeneral.sActPathTolerance.r64ActivateMIDirectLimit := 0.1; //mm

Bis zu einer Abweichung von 50mm sollen die Ausgangsachsen auf die Sollposition positioniert werden.

gsParameter.sGeneral.sActPathTolerance.r64ActivateMIPullAxesLimit := 50.0; //mm

Darüber hinaus erfolgt ein Setzen der Eingangsachsen auf die tatsächliche Position.

Überwachung des Motion Interface im laufenden Betrieb

Die Bahnsollwertüberwachung wird ebenfalls auf einen kleinen Wert eingestellt, da Sollwertabweichungen nicht auftreten sollen.

gsParameter.sGeneral.sCmdPathTolerance.r64Limit := 1.0; //mm

Beim Schleppabstand der Bahn wird ein größerer Wert zugelassen, da hier Systembedingt (Reglereinstellung).

gsParameter.sGeneral.sActPathFollowingError.r64Limit := 10.0; //mm

6.2.11 Frametransformation

Zur räumlichen Anpassung wurde eine Frametransformation in den Funktionsbaustein integriert. Mit dieser kann die Orientierung des geometrische Koordinatensystems gedreht und verschoben werden. Um eine Abgrenzung zum Begriff Frame, wie er in der NC-Sprache definiert ist, vorzunehmen, wird im Folgenden erläutert, welche Bedeutung der Begriff Frame im Bezug aufden FBLKTransKinTransformation hat.

Frame

Über einen Frame läßt sich ein Koordinatensystem in ein anderes überführen.

Hierbei unterscheidet man zwischen Translation und Rotation. In Vorwärtsrichtung wird erst die Rotation dann die Translation gerechnet.

Alle hier beschriebenen Einstellungen befinden sich in der Parameterstruktur im Bereich *sTransformation.sCoordTrans* (siehe Kapitel 5.2.3).

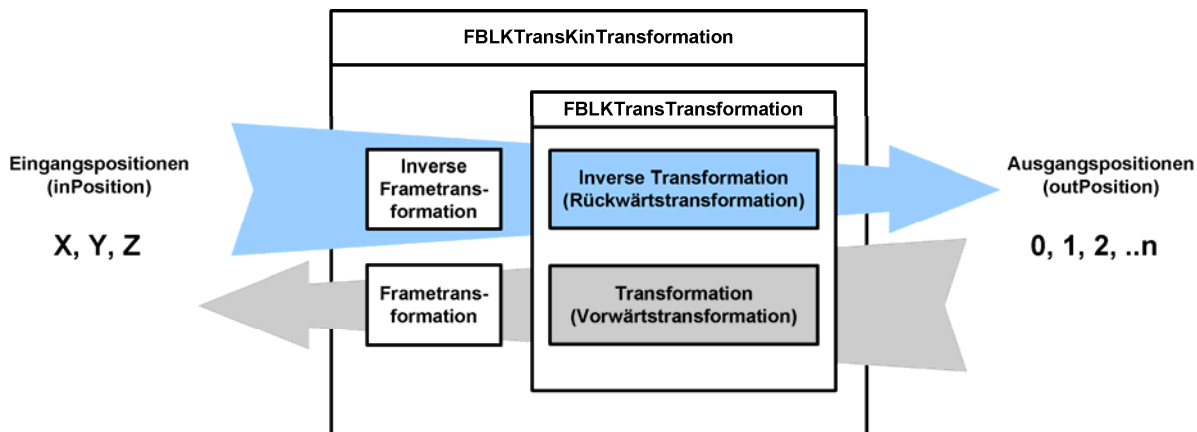


Bild 6-14: Frametransformation im FBKin Transformation

6.2.11.1 Translation

Als Bezeichner für die Translation werden die Koordinaten X, Y und Z verwendet.

Sie sind so festgelegt, dass das Koordinatensystem ein Rechtssystem ergibt.

Über die Variablen *ar64Translation[1 .. 3]* wird die Position des eingebetteten Koordinatensystem im Basiskoordinatensystem definiert (Position des Koordinatenursprungs des eingebetteten Koordinatensystem im Basiskoordinatensystem).

ar64Translation[1]: X - Position
ar64Translation[2]: Y - Position
ar64Translation[3]: Z - Position

6.2.11.2 Rotation

Für die Rotation verwendet man die RPY-Winkel *ar64Rotation[1]*, *ar64Rotation[2]*, *ar64Rotation[3]* (RPY steht für Roll Pitch Yaw). D.h., es erfolgt erst die Drehung um die die X-Achse dann um die (bereits gedrehte) Y-Achse und nachfolgend um die (bereits doppelt gedrehte) Z-Achse. Die positive Drehrichtung wird durch die Rechte-Hand-Regel festgelegt, d.h. zeigt der Daumen der rechten Hand in Richtung der Drehachse, so geben die restlichen Finger die positive Winkelrichtung an. Dabei ist zu beachten, dass *aRot[1]* und *aRot[3]* im Intervall $[-180; +180]$ und *aRot[2]* im Intervall $[-90; +90]$ sinnvoll zu wählen sind, um Mehrdeutigkeiten auszuschliessen.

Die Winkel sind innerhalb der Kinematikstruktur folgendermaßen festgelegt:

- ar64Rotation[1]: **Roll** um die x-Achse
- ar64Rotation[2]: **Pitch** (um die gedrehte) y-Achse
- ar64Rotation[3]: **Yaw** (um die bereits doppelt gedrehte) z-Achse

In Bild 6-15 ist ein Beispiel für die Drehung um die Winkel angegeben. Hierbei wird das Ausgangskordinatensystem X1, Y1, Z1 zuerst um den Winkel ar64Rotation[1] um die X1-Achse gedreht, dann um den Winkel ar64Rotation[2] um die Y2-Achse und zuletzt um den Winkel ar64Rotation[3] um die Z3-Achse.

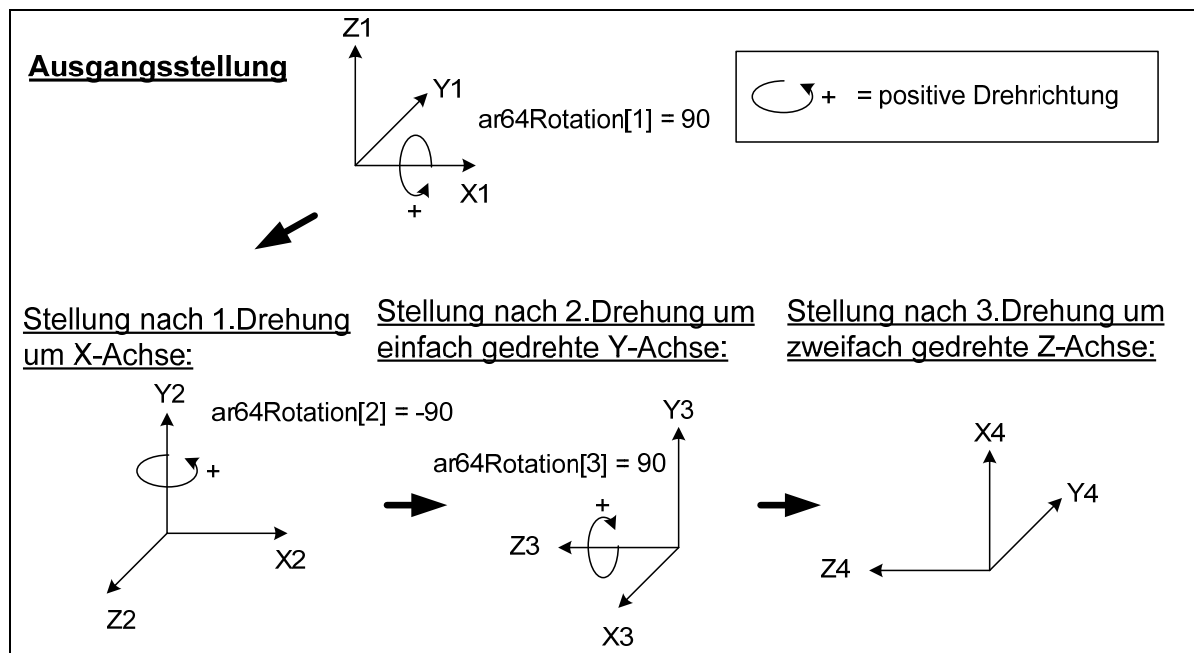


Bild 6-15: Rotation des Koordinatensystems

Intern werden für die Frametransformation die Funktionen FCLKTRansFrameTrans und FCLKTRansInvFrameTrans verwendet. Diese können auch vom Anwender genutzt werden und sind im Kapitel 6.3 beschrieben.

6.3 Funktionen FCLKTransFrameTrans und FCLKTransInvFrameTrans

6.3.1 Aufgabe

Mit den Funktionen FCLKTransFrameTrans und FCLKTransInvFrameTrans können Koordinatensysteme mittels Frametransformation angepasst werden. Hierbei wird zwischen Translation und Rotation unterschieden. Das ursprüngliche Koordinatensystem wird als Array vom Datentyp LREAL übergeben. Hierbei werden die Indizes 0,1 und 2 als X,Y und Z interpretiert. Die Function FCLKTransFrameTrans passt das Koordinatensystem um die angegebenen Werte an und gibt diese als Rückgabewert (Ebenfalls als Array) zurück. Der FCLKTransInvFrameTrans verfährt genauso, nur dass er ein angepasstes Koordinatensystem wieder auf das Ursprungssystem zurück transformiert. Die Parameter des Frame werden als Struktur vom Typ *sLKTransFrameTransType* (siehe 5.2.3) am Eingangsparameter *coordIn* übergeben. Details über die Funktionsweise sind auch in Kapitel 6.2.11 nachlesbar. In der Funktion FCLKTransFrameTrans wird erst die Rotation dann die Translation gerechnet, im FCLKTransInvFrameTrans erst die Translation dann die Rotation.

6.3.2 Schematische KOP/FUP-Darstellung

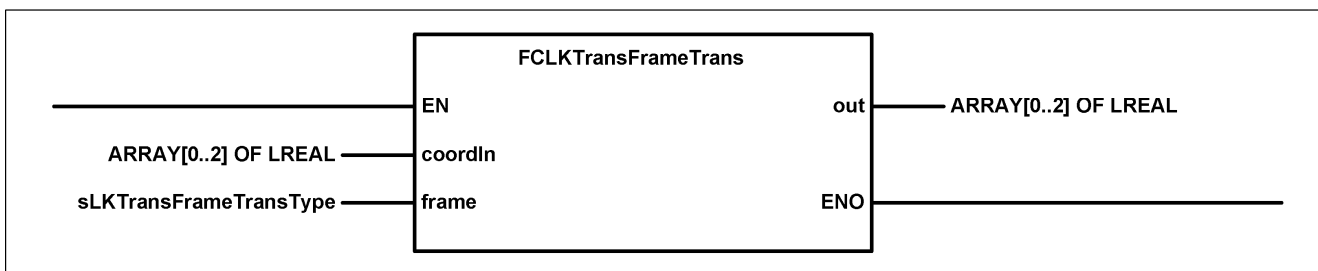


Bild 6-16: KOP-Ansicht des FCLKTransFrameTrans

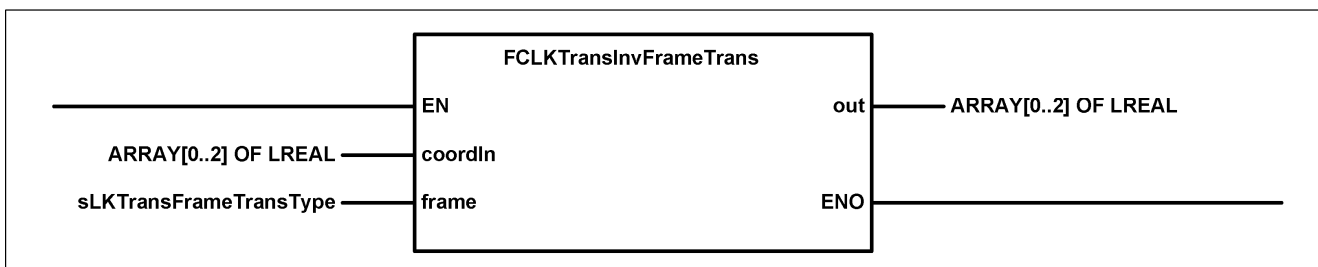


Bild 6-17: KOP-Ansicht des FCLKTransInvFrameTrans

6.3.3 Parameter der Funktionen

Name	P-Typ	Datentyp	Bedeutung
coordIn	IN	ARRAY[0..2] OF LREAL	Koordinatensystem (X,Y,Z)
frame	IN	sLKTransFrameTransType	Framestruktur, enthält Parametern zur Verschiebungen und Rotationen
out	OUT	ARRAY[0..2] OF LREAL	Rückgabewert: Angepasstes Koordinatensystem (X,Y,Z)

7 Integration eigener Kinematiken

Um eine eigene Kinematiktransformation zu erstellen, sollte der Anwender nachfolgende Schritte durchführen.

7.1 Anpassung von Konstanten

Es ist zu prüfen ob genügend Ein- / Ausgangswerte und Transformationsparameter zur Verfügung stehen, oder ob die Anzahl mittels der Konstanten erhöht werden muss. Ein Herabsetzen der bereits definierten Anzahl ist meist nicht sinnvoll, da realisierte Kinematiken diese Listenelemente möglicherweise verwenden und ein Übersetzen des Programmcode dann nicht möglich ist.

Die Definition der Konstanten erfolgt in der Bibliotheksunit *cPublic*.

Im Standard sind bereits vier Eingangspositionen (z.B. für X, Y, Z und W) und zehn Ausgangspositionen für den FBLKTransTransformation festgelegt. Es wurde eine Liste von zwanzig Transformationsparametern (z.B. für mechanische Abmessungen, Gelektstellungsräume und Offsetpositionen) vorgesehen.

```
//constants for FBLKTransTransformation
//-----
//number of input values, range: 3..USINT#MAX
LKTRANS_NUMBER_OF_INPOSITIONS      : USINT := 4; //default: 4

//number of output values, range: 1..USINT#MAX
LKTRANS_NUMBER_OF_OUTPOSITIONS     : USINT := 10; //default: 10

//number of transformation parameter, range: 1..USINT#MAX
LKTRANS_NUMBER_OF_TRANS_PARAMETER  : USINT := 20;
```

Bild 7-1: Codeabschnitt zur Konstantendefinition FBLKTransTransformation in Unit cPublic

Die Anzahl der tatsächlich genutzten Eingangs- und Ausgangsachsen am FBLKTransKinTransformation kann hiervon getrennt eingestellt werden.

7.2 Erstellung Vorwärts- und Rückwärtstransformation

Um die gesamte Funktionalität zu nutzen, muss der Anwender sowohl die Vorwärtstransformation wie auch die Rückwärtstransformation realisieren.

Bei der Rückwärtstransformation werden aus den kartesischen Sollpositionen (commandInPosition[0, 1, 2] = [X,Y,Z]) die Maschinenachspalten (commandOutPosition[0,...,n]) berechnet.

Bei der Vorwärtstransformation werden die aktuellen Maschinenachspalten (actualOutPosition[0,..., n]) in die katesischen Istpositionen (actualInPosition[0,1,2] = [X,Y,Z]) umgerechnet.

Das Einbinden mechanischer Abmessungen, Verschiebungen oder Ähnliches erfolgt über den Ein- / Ausgangsparameter *parameter[n]*, der aus einem Feld vom Typ LREAL besteht. In der Parameterstruktur des FBLKTransKinTransformation ist dieses unter *.sTransformation.ar64Parameter[n]* zu finden. Für jede neue Transformation ist eine eigene Transformationsnummer festzulegen.

Soll die Transformation im Anwenderprogramm auch als Auskunftsfunktion verwendet werden, sollte der Eingangsparameter calcMode, mit seinen drei Zuständen TRANS, INV_TRANS und BOTH berücksichtigt werden.

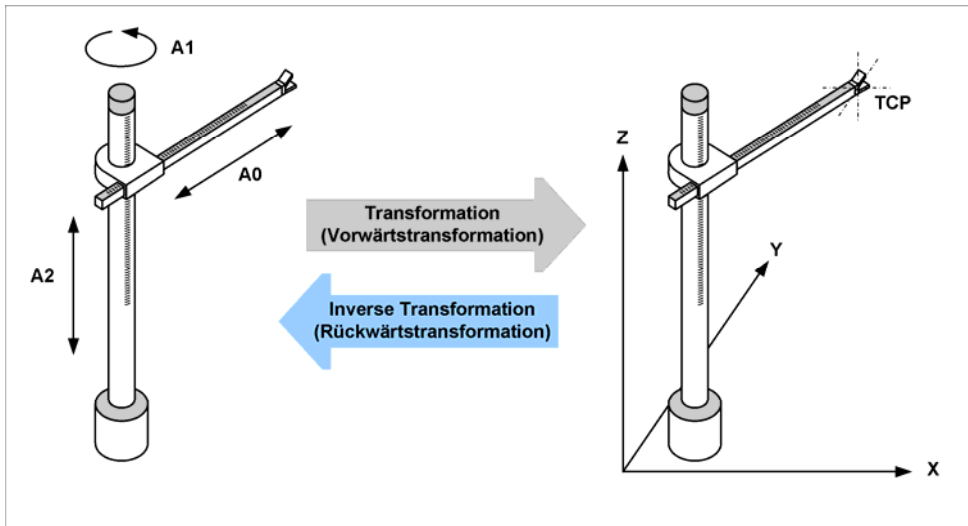


Bild 7-2: Vorwärts- und Rückwärtstransformation

In Bild 7-3 ist ein einfaches Transformationsbeispiel dargestellt.

```

IF calcMode = TRANS OR calcMode = BOTH THEN
  //transformation
  //-----
  actualInPosition[X] := actualOutPosition[0] * COS(DEGRAD * actualOutPosition[1]);
  actualInPosition[Y] := actualOutPosition[0] * SIN(DEGRAD * actualOutPosition[1]);
  actualInPosition[Z] := actualOutPosition[2];
END_IF;
    
```

Bild 7-3: Einfaches Transformationsbeispiel

Zur Erhöhung der Lesbarkeit werden im dargestellten Code für die Indizes der kartesischen Positionen die Konstanten X, Y und Z verwendet. Zur einfachen Wandlung von °(Bogenmaß, $0..2\pi$) nach °(Winkelgrad, $0..360$) stehen entsprechende Konstanten wie DEGRAD zur Verfügung.

```

VAR_GLOBAL CONSTANT
  //Constant PI
  PI      : LREAL := 3.1415926535897932384626433832795;
  //Angle[RAD]=DEGRAD*Angle[DEG]
  DEGRAD  : LREAL := 1.74532925199432957692369076848e-2;
  //Angle[DEG]=RADDEG*Angle[RAD]
  RADDEG  : LREAL := 57.295779513082320876798154814105;
  //use for axes
  X       : USINT := 0;
  Y       : USINT := 1;
  Z       : USINT := 2;
  W       : USINT := 3;
END_VAR
    
```

Bild 7-4: Verfügbare Anwenderkonstanten

Der Anwender muss die Erkennung von Fehlern sicherstellen. In Bild 7-5 ist der Aufruf eines Fehlers durch den Anwender dargestellt. Die Ausgangsvariable error wird gesetzt und die Fehlernummer als Konstante LKTRANS_ERR_XY_UNDEFINED dem Ausgangsparameter errorId zugewiesen. In der Bibliotheksunit cPublic können eigene Fehlerkonstanten angelegt werden. Außerhalb des Bausteins kann nun auf diese Fehlermeldung reagiert werden.

```
//inverse transformation
IF parameter[2] <= SQRT(
  commandInPosition[X] ** 2.0
  + commandInPosition[Y] ** 2.0 )
THEN
  //calculate inverse transformation
  //...

ELSE
  //error
  error := TRUE;
  errorId := LKTRANS_ERR_XY_UNDEFINED; //error number 4002
END_IF;
```

Bild 7-5: Beispiel (Ausschnitt) einer Fehlerabfrage

8 Fehler und Warnungen

In der folgenden Tabelle sind alle Fehler, die bei den Funktionsbausteinen **FBLKTransTransformation** und **FBLKTransKinTransformation** auftreten können, beschrieben. Fehler am FBLKTransTransformation werden in jedem Zyklus ermittelt und müssen nicht quittiert werden.

Tritt am Baustein FBLKTransKinTransformation ein Fehler auf, werden weitere Details in der Diagnosestruktur am Bausteinparameter *diagnostics* angezeigt. Hierzu stehen die Parameter *i8AxNumber*, *i32Detail1*, *b32Detail2*, *r32Detail3* zur Verfügung.

Ebenso befindet sich in der Diagnosestruktur ein Fehlerpuffer (*asBuffer*) in Form eines Ringpuffers, der die vergangenen Fehler zusammen mit dem Zeitpunkt des Auftretens enthält. Der Aktuelle (letzte) Eintrag wird vom Parameter *i32BufferIndex* angezeigt.

Fehler an diesem Baustein müssen durch Rücksetzen des Eingangsparameters *enable* quittiert werden.

Tabelle 2: Fehlerliste

Fehler-Nr. Hex (dez) Kurzbezeichnung	Bedeutung / Details
0000_4000 (16384) LKTRANS_ERR_UNKNOWN_TRANSID	Am Baustein FBLKTransTransformation wurde eine unbekannte Transformationsnummer ausgewählt.
0000_4001 (16385) LKTRANS_ERR_OUTAX_0_POSITION	Die aktuelle Position der Ausgangsachse 0 befindet sich außerhalb des gültigen Bereichs.
0000_4002 (16386) LKTRANS_ERR_XY_UNDEFINED	Die vorgegebene, kartesische Position (X,Y) kann mit der aktuellen Parametrierung nicht erreicht werden.
0000_4003 (16387) LKTRANS_ERR_SINGULARITY	Bei der Berechnung der inversen Transformation ist ein Fehler aufgetreten.
0000_4004 (16388) LKTRANS_ERR_OUT_OF_RANGE	Die Kinematik kann die Position mit den angegebenen geometrischen Maßen nicht erreichen
0000_4003 .. 0000_4999	Freigehalten für weitere Anwenderfehler im FBLKTransTransformation
0000_5000 (20480) LKTRANS_ERR_TRANSFORMATION	FBLKTransKinTransformation: Es ist ein Fehler im Transformationsbaustein (FBLKTransTransformation) aufgetreten. Detailinformationen: b32Detail2: Fehlernummer des Transformationsbausteins
0000_5001 (20481) LKTRANS_ERR_STATE_REDEFINE_COMMAND	Fehler bei Setzen der Eingangsachsen mittels Systemfunktion <i>_redefinePosition()</i> . Detailinformationen: u8AxNumber: Index der betroffenen Eingangsachse i32Detail1: 1 = Error, 2= Abbruch b32Detail2: Fehlernummer gemäß PLC-Open

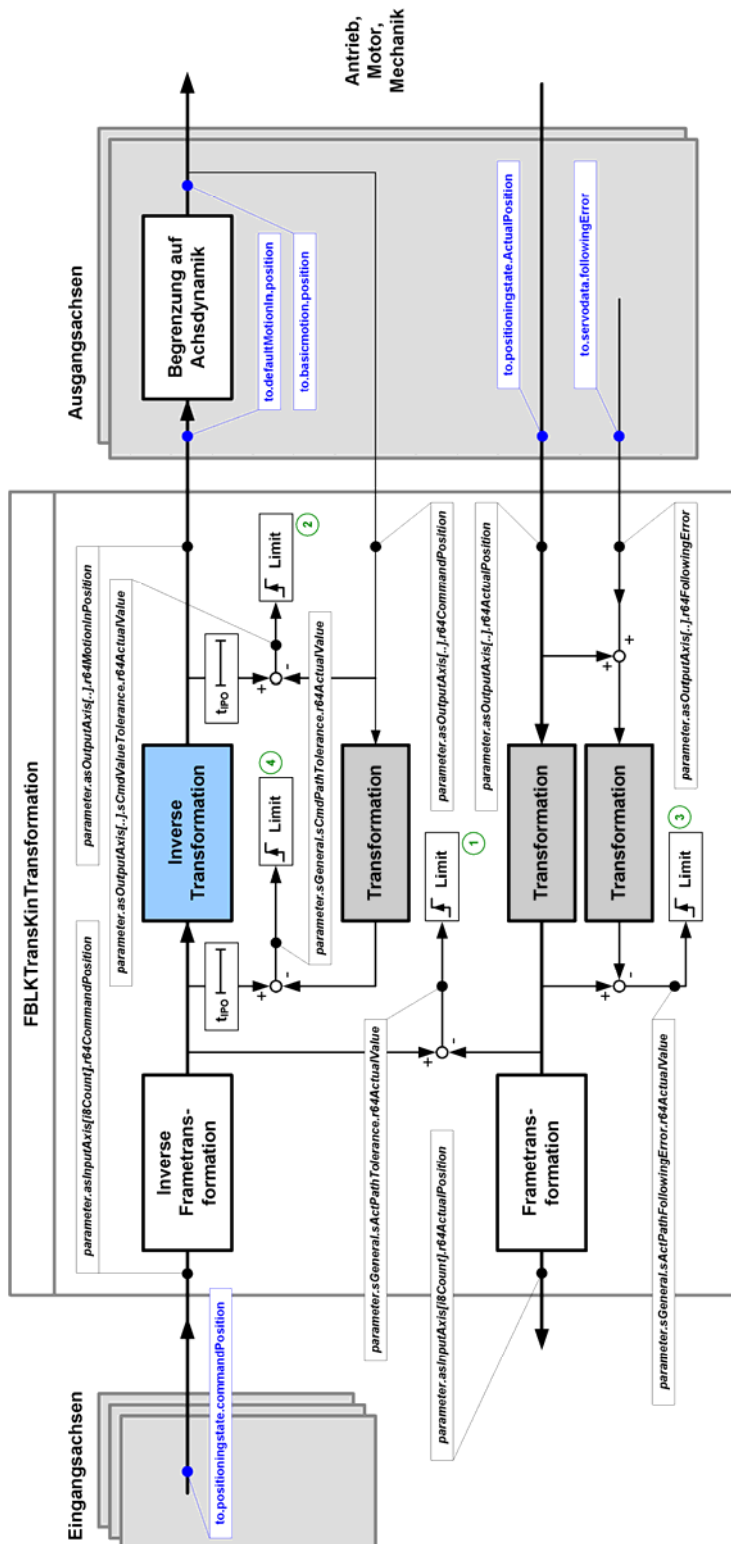
Fehler-Nr. Hex (dez) Kurzbezeichnung	Bedeutung / Details
0000_5002 (20482) LKTRANS_ERR_MOTIONIN	<p>Fehler bei der Aktivierung oder Überwachung des Motion Interface mittels Systemfunktion <code>_runPositionBasedMotionIn()</code>. Detailinformationen:</p> <p>u8AxNumber: Index der betroffenen Ausgangsachse i32Detail1: 1 = Error, 2= Abbruch b32Detail2: Fehlernummer gemäß PLC-Open</p>
0000_5003 (20483) LKTRANS_ERR_CONSTANTS	<p>Die Werte für die Anwenderkonstanten (siehe Kapitel 5.1), der Bibliotheksunit <code>cPublic</code> sind ungültig oder nicht aufeinander abgestimmt. Detailinformationen:</p> <p>i32Detail1: 1 = <code>NUMBER_OF_TRANS_PARAMETER</code> muss größer 0 sein 2 = <code>NUMBER_OF_INPOSITIONS</code> muss größer 0 sein 3 = <code>NUMBER_OF_OUTPOSITIONS</code> muss größer 0 sein 4 = <code>NUMBER_OF_INPOSITIONS</code> muss größer als <code>NUMBER_OF_USED_INPUT_AXES</code> sein. 5 = <code>NUMBER_OF_OUTPOSITIONS</code> muss größer als <code>NUMBER_OF_USED_OUTPUT_AXES</code> sein.</p>
0000_5004 (20484) LKTRANS_ERR_LIMIT_REACHED_AFTER_REDEF	<p>Nach erfolgtem Setzen der Eingangsachsen auf transformierte Positionen wurde eine oder mehrere Begrenzung überschritten und deshalb das Aktivieren des Motion Interfaces abgebrochen. Der Baustein wechselt in den Zustand <code>INACTIVE</code>. Ursache für diesen Fehler können zu klein gewählte Begrenzungen oder Abweichungen von Transformation zu inverser Transformation sein. Detailinformationen:</p> <p>u8AxNumber: Index der betroffenen Achse (wenn <code>Details1 = 4</code>) i32Detail1: Überschrittene Überwachung 1 = Limit zum direkten Aktivieren des MI (<code>sActPathTolerance.r64ActivateMIDirectLimit</code>) 2 = Limit zum heranziehen der Achsen. Parameter (<code>sActPathTolerance.r64ActivateMIPullAxesLimit</code>) 3= Überwachung gemäß 1 und 2 wurden überschritten 4 = Sollwertüberwachung der Achsen. Parameter (<code>asOutputAxis[...].sCommandValueTolerance.r64Limit</code>) r32Detail3: Abweichung zum Auslösezeitpunkt</p>
0000_5005 (20485) LKTRANS_ERR_AX_CMDVAL_TOLERANCE	<p>Bei aktiviertem Motion Interface wurde an mindestens einer Ausgangsachse die Sollwertüberwachung überschritten (Parameter <code>sCmdValueTolerance.r64Limit</code>). Als Reaktion wird das Motion Interface deaktiviert. Detailinformationen:</p> <p>u8AxNumber: Index der betroffenen Ausgangsachse r32Detail3: Eingestellte (überschrittene) Toleranz</p>
0000_5006 (20486) LKTRANS_ERR_PATH_TOLERANCE	<p>Bei aktiviertem Motion Interface wurde die projektierte Bahntoleranz überschritten (Parameter <code>r64ActualPathToleranceLimit</code>). Als Reaktion wird das Motion Interface deaktiviert. Detailinformationen:</p> <p>r32Detail3: Eingestellte (überschrittene) Toleranz</p>

Fehler-Nr. Hex (dez) Kurzbezeichnung	Bedeutung / Details
0000_5007 (20487) LKTRANS_ERR_AXES_ PRECONDITION	<p>Beim Aktivieren des Motion Interface wurden die Voraussetzungen an mindestens einer Achse nicht erfüllt. Detailinformationen:</p> <p>u8AxNumber: Index der betroffenen Achse i32Detail1: 1 = Eingangsachse, 2 = Ausgangsachse b32Detail2: aktuelles Statuswort (siehe Kapitel 5.2.7) der betroffenen Achse (Es müssen folgende Bits gesetzt sein: 0, 1, 2, 3, 4)</p>
0000_5008 (20488) LKTRANS_ERR_OUTPUTAXIS	<p>Beim Betrieb des Motion Interface wurde an mindestens einer Ausgangsachse eine Bedingung nicht erfüllt. Detailinformationen:</p> <p>u8AxNumber: Index der betroffenen Achse i32Detail1: 1 = Eingangsachse, 2 = Ausgangsachse b32Detail2: aktuelles Statuswort (siehe Kapitel 5.2.7) der betroffenen Achse (Es müssen folgende Bits gesetzt sein: 0, 1, 2, 3)</p>
0000_5009 (20489) LKTRANS_ERR_NO_AXES_USED	<p>Es konnten aus der Parameterstruktur keine Achsreferenzen zur Verwendung ausgelesen werden. Es muss mindestens eine Achse verwendet werden. Detailinformationen:</p> <p>i32Detail1: 1 = Eingangsachsen, 2 = Ausgangsachsen</p>
0000_500A (20490) LKTRANS_ERR_STATE_POS_COMMAND	<p>Es ist ein Fehler im Zustand PULL_AXES aufgetreten. Die Positionierung einer oder mehrerer Achsen wurde abgebrochen. Hierzu wird der Systemfunktionsbaustein _MC_MoveAbsolute() verwendet.</p> <p>u8AxNumber: Index der betroffenen Achse i32Detail1: 1 = Fehler (error), 2 = Abbruch (commandAborted) b32Detail2: Fehlernummer gemäß PLCOpen</p>
0000_500B (20491) LKTRANS_ERR_ LIMIT_REACHED_AFTER_PULL	<p>Nach dem Heranziehen der Ausgangsachsen (PULL_AXES) wird die Grenze einer Überwachung noch immer überschritten. Der Baustein wechselt in den Zustand INACTIVE. Die projektierten Grenzwerte sollten geprüft werden.</p> <p>u8AxNumber: Index der betroffenen Achse (wenn Details1 = 4) i32Detail1: Überschrittene Überwachung 1 = Limit zum direkten Aktivieren des MI (sActPathTolerance.r64ActivateMIDirectLimit) 2 = Limit zum heranziehen der Achsen. Parameter (sActPathTolerance.r64ActivateMIPullAxesLimit) 3= Überwachung gemäß 1 und 2 wurden überschritten 4 = Sollwertüberwachung der Achsen. Parameter (asOutputAxis[...].sCommandValueTolerance.r64Limit) r32Detail3: Abweichung zum Auslösezeitpunkt</p>
0000_500C (20492) LKTRANS_ERR_INPUT_PARAMETER	<p>Es ist zu einem internen Fehler in der Berechnung des Bewegungsvektors gekommen. Hierzu wird die Funktion FBLKTransCalcMotionVec() verwendet. Bitte wenden Sie sich an Ihren SIEMENS Ansprechpartner.</p> <p>i32Detail1: Aktuelle Rechenmode (eLKTransMICalculationType)</p>

Fehler-Nr. Hex (dez) Kurzbezeichnung	Bedeutung / Details
0000_500D (20493) LKTRANS_ERR_SPECIFIC_POINT	Es ist zu einem internen Fehler in der Berechnung des Bewegungsvektors gekommen. Hierzu wird die Funktion FBLKTransCalcMotionVec() verwendet. Bitte wenden Sie sich an Ihren SIEMENS Ansprechpartner. i32Detail1: Aktuelle Rechenmode (eLKTransMICalculationType)
0000_500E (20494) LKTRANS_ERR_PATH_CMDVAL_TOLERANCE	Die Überwachung des Bahn- Sollwertes hat eine Überschreitung des Grenzwertes festgestellt. Der Baustein wechselt in den Zustand INACTIVE. i32Detail1: Aktuell eingestellter Grenzwert (aus parameter.sGeneral.sCmdPathTolerance.r64Limit)
0000_500F (20495) LKTRANS_ERR_PATH_FOLLOWING_ERROR	Die Überwachung des Bahn- Schleppfehlers hat eine Überschreitung des Grenzwertes festgestellt. Der Baustein wechselt in den Zustand INACTIVE. i32Detail1: Aktuell eingestellter Grenzwert (aus parameter.sGeneral.sActPathFollowingError.r64Limit)
0000_5010 (20496) LKTRANS_ERR_UNKNOWN_CALCULATION_MODE	Es wurde ein Berechnungs- Mode gewählt der nicht unterstützt wird. Es sollte die Konfiguration (sGeneral.eMVecCalcMode) geprüft werden. i32Detail1: angewählter (nicht unterstützter) Rechenmode (eLKTransMICalculationType)
0000_5011 (20497) LKTRANS_ERR_TASK_CYCLE_TIME	Es ist zu einem internen Fehler in der Berechnung des Bewegungsvektors gekommen, da die Zykluszeit fehlerhaft ist. Hierzu wird die Funktion FBLKTransCalcMotionVec() verwendet. Bitte wenden Sie sich an Ihren SIEMENS Ansprechpartner. i32Detail1: Aktuelle Rechenmode (eLKTransMICalculationType)
0000_5012 (20498) LKTRANS_ERR_STATE_STOP_COMMAND	Es ist ein Fehler im Zustand STOP_PULL_AXES aufgetreten. Der Stop einer oder mehrerer Achsen wurde abgebrochen. Hierzu wird der Systemfunktionsbaustein _MC_Stop() verwendet. u8AxNumber: Index der betroffenen Achse i32Detail1: 1 = Fehler (error), 2 = Abbruch (commandAborted) b32Detail2: Fehlernummer gemäß PLCOpen

9 Anhang

9.1 Überwachungen



9.2 Ansprechpartner

Siemens AG

Industry Sector

I DT MC PM APC

Frauenauracher Straße 80

D - 91056 Erlangen

Fax.: +49 9131 98 1297

mailto: applications.erlf.aud@siemens.com

9.3 Internetadressen

Weitere Informationen zu den verschiedenen Themen finden Sie auf folgenden Internetseiten:

SIMOTION (www.siemens.com/simotion)

SINAMICS (www.siemens.com/sinamics)

Handling (www.automation.siemens.com/mc-app/handling/html_00/00-handling.htm)

TopLoading (<http://support.automation.siemens.com/WW/view/de/37584177>)

MotionControl/Applikationszentrum (www.siemens.com/motioncontrol/apc)

10 Stichwortverzeichnis

A

Ablaufdiagramm FBLKTransKinTransformation.....	35
Aktivierung Motion Interface.....	36
Allgemeine Parameter.....	23
aVersion.....	15

B

b32state.....	27
Bahnabweichung.....	38, 39
Bausteinzustand.....	28
Berechnungsart.....	29
Berechnungsart Transformation.....	28
Bibliotheksstruktur.....	15

C

cPublic.....	15
--------------	----

D

Diagnosestruktur.....	27
dProtected.....	15
Dynamikgrenzen bei Stop.....	26
Dynamikgrenzen beim (Achsen-) Heranziehen.....	26
Dynamikgrenzen des Motion Interface.....	25

E

eLKTransCalcModeType.....	28
eLKTransMICalculationType.....	29
eLKTransStateType.....	28
Enumeratoren.....	28
ErrorId	
16#4000.....	48
16#4001.....	48
16#4002.....	48
16#4003.....	48
16#4004.....	48
16#5000.....	48
16#5001.....	49
16#5003.....	49
16#5004.....	49
16#5005.....	49
16#5006.....	50
16#5007.....	50
16#5008.....	50
16#5009.....	50, 51
16#500A.....	50
16#500B.....	50
16#500C.....	51
16#500D.....	51
16#500E.....	51
16#5010.....	51
16#5011.....	51
16#5012.....	51

F

FBLKTransKinTransformation.....	33
FBLKTransTransformation.....	30
FCLKTransFrameTrans.....	44
FCLKTransInvFrameTrans.....	44
Fehler und Warnungen.....	48
fKinTrans.....	15
Frametransformation.....	42
Frametransformationsparameter.....	24
fTrans.....	15

G

Globale Konstanten.....	22
-------------------------	----

I

Istwertabweichung.....	24
------------------------	----

L

LKTRANS_ERR.....	51
LKTRANS_ERR_INPUT_PARAMETER.....	51
LKTRANS_ERR_LIMIT_REACHED_AFTER_PULL.....	50
LKTRANS_ERR_PATH_CMDVAL_TOLERANCE.....	51
LKTRANS_ERR_SPECIFIC_POINT.....	51
LKTRANS_ERR_STATE_POS_COMMAND.....	50
LKTRANS_ERR_STATE_STOP_COMMAND.....	51
LKTRANS_ERR_TASK_CYCLE_TIME.....	51
LKTRANS_ERR_UNKNOWN_CALCULATION_MODE.....	51
LKTRANS_ERR_AX_CMDVAL_TOLERANCE.....	49
LKTRANS_ERR_AXES_PRECONDITION.....	50
LKTRANS_ERR_CONSTANTS.....	49
LKTRANS_ERR_LIMIT_REACHED_AFTER_REDEF.....	49
LKTRANS_ERR_MOTIONIN.....	49
LKTRANS_ERR_NO_AXES_USED.....	50
LKTRANS_ERR_OUT_OF_RANGE.....	48
LKTRANS_ERR_OUTAX_0_POSITION.....	48
LKTRANS_ERR_OUTPUTAXIS.....	50
LKTRANS_ERR_PATH_TOLERANCE.....	50
LKTRANS_ERR_SINGULARITY.....	48
LKTRANS_ERR_STATE_REDEFINE_COMMAND.....	49
LKTRANS_ERR_TRANSFORMATION.....	48
LKTRANS_ERR_UNKNOWN_TRANSID.....	48
LKTRANS_ERR_XY_UNDEFINED.....	48
LKTRANS_EXECUTIONLEVEL.....	22
LKTRANS_NUMBER_OF_DIAGBUFFER_ITEMS.....	22
LKTRANS_NUMBER_OF_INPOSITIONS.....	22
LKTRANS_NUMBER_OF_OUTPOSITIONS.....	22
LKTRANS_NUMBER_OF_TRANS_PARAMETER.....	22
LKTRANS_NUMBER_OF_USED_INPUT_AXES.....	22
LKTRANS_NUMBER_OF_USED_OUTPUT_AXES.....	22

P

Parameter der Ausgangsachsen.....	25
Parameter der Eingangsachsen.....	25
Parameter der Sollwertüberwachung.....	25

Stichwortverzeichnis

Parameter eines Diagnosepuffereintrags	27
Parameterstruktur	23

R

Rotatorische Verschiebung	42
---------------------------------	----

S

Schleppabweichung	24
sLKTransDiagnosticItemType	27
sLKTransDiagnosticsType	27
sLKTransFrameTransType	24
sLKTransGeneralActPathToleranceType	24
sLKTransGeneralCmdPathToleranceType	23
sLKTransGeneralFollowingErrorType	24
sLKTransGeneralType	23
sLKTransInputAxisType	25
sLKTransOutputAxCmdValueToleranceType	25
sLKTransOutputAxDynamicType	25
sLKTransOutputAxisType	25
sLKTransOutputAxMIDynamikType	25
sLKTransOutputAxPullDynamikType	26

sLKTransOutputAxStopDynamikType	26
sLKTransParameterType	23
sLKTransTransformationType	24
sLKTransType	24
Sollwertabweichung	23
Sollwertüberwachung Achse	39
Statuswörter der Achsen	27

T

Technologische Parameter	23
Transformationsparameter	24
Translatorische Verschiebung	42

U

Überwachungen	38
---------------------	----

Z

Zustandsmodell FBLKTransKinTransformation	35
---	----