



DOKUMENTATION ISG-kernel

McCOM - Anbindung einer kinematischen Transformation

Kurzbezeichnung:
McCOM-Trafo

© Copyright
ISG Industrielle Steuerungstechnik GmbH
STEP, Gropiusplatz 10
D-70563 Stuttgart
Alle Rechte vorbehalten
www.isg-stuttgart.de
support@isg-stuttgart.de

Dokumentation Version: 1.11
13.11.2024

Vorwort

Rechtliche Hinweise

Diese Dokumentation wurde sorgfältig erstellt. Die beschriebenen Produkte und der Funktionsumfang werden jedoch ständig weiterentwickelt. Wir behalten uns das Recht vor, die Dokumentation jederzeit und ohne Ankündigung zu überarbeiten und zu ändern.

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

Qualifikation des Personals

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

Zur Installation und Inbetriebnahme ist die Beachtung der Dokumentation, der nachfolgenden Hinweise und Erklärungen unbedingt notwendig. Das Fachpersonal ist verpflichtet, für jede Installation und Inbetriebnahme die zum betreffenden Zeitpunkt veröffentlichte Dokumentation zu verwenden.

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

Weiterführende Informationen

Unter den Links (DE)

<https://www.isg-stuttgart.de/produkte/softwareprodukte/isg-kernel/dokumente-und-downloads>

bzw. (EN)

<https://www.isg-stuttgart.de/en/products/softwareproducts/isg-kernel/documents-and-downloads>

finden Sie neben der aktuellen Dokumentation weiterführende Informationen zu Meldungen aus dem NC-Kern, Onlinehilfen, SPS-Bibliotheken, Tools usw.

Haftungsausschluss

Änderungen der Software-Konfiguration, die über die dokumentierten Möglichkeiten hinausgehen, sind unzulässig.

Marken und Patente

Der Name ISG®, ISG kernel®, ISG virtuos®, ISG dirigent® und entsprechende Logos sind eingetragene und lizenzierte Marken der ISG Industrielle Steuerungstechnik GmbH.

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

Copyright

© ISG Industrielle Steuerungstechnik GmbH, Stuttgart, Deutschland.

Weitergabe sowie Vervielfältigung dieses Dokuments, Verwertung und Mitteilung seines Inhalts sind verboten, soweit nicht ausdrücklich gestattet. Zuwiderhandlungen verpflichten zu Schadenersatz. Alle Rechte für den Fall der Patent-, Gebrauchsmuster oder Geschmacksmustereintragung vorbehalten.

Allgemeine- und Sicherheitshinweise

Verwendete Symbole und ihre Bedeutung

In der vorliegenden Dokumentation werden die folgenden Symbole mit nebenstehendem Sicherheitshinweis und Text verwendet. Die (Sicherheits-) Hinweise sind aufmerksam zu lesen und unbedingt zu befolgen!

Symbole im Erklärtext

- Gibt eine Aktion an.
- ⇒ Gibt eine Handlungsanweisung an.



GEFAHR

Akute Verletzungsgefahr!

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



VORSICHT

Schädigung von Personen und Maschinen!

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



Achtung

Einschränkung oder Fehler

Dieses Symbol beschreibt Einschränkungen oder warnt vor Fehlern.



Hinweis

Tipps und weitere Hinweise

Dieses Symbol kennzeichnet Informationen, die zum grundsätzlichen Verständnis beitragen oder zusätzliche Hinweise geben.



Beispiel

Allgemeines Beispiel

Beispiel zu einem erklärten Sachverhalt.



Programmierbeispiel

NC-Programmierbeispiel

Programmierbeispiel (komplettes NC-Programm oder Programmsequenz) der beschriebenen Funktionalität bzw. des entsprechenden NC-Befehls.



Versionshinweis

Spezifischer Versionshinweis

Optionale, ggf. auch eingeschränkte Funktionalität. Die Verfügbarkeit dieser Funktionalität ist von der Konfiguration und dem Versionsumfang abhängig.

Inhaltsverzeichnis

Vorwort	2
Allgemeine- und Sicherheitshinweise	3
1 Übersicht	7
2 Beschreibung	8
2.1 Allgemeines zur kinematischen Transformation (TRAFO).....	9
2.1.1 Koordinatensysteme	11
2.1.2 Positionsverschiebungen	13
2.1.3 Modulo-Einstellung der Achsen	15
3 Anbindung Transformation via TcCOM	16
3.1 Methoden der Transformation.....	16
3.2 Arbeitsdaten (Instanzdaten) der Transformation.....	17
3.2.1 Basis Arbeitsdaten: TcNcTrafoParameter	17
3.2.2 Erweiterte Arbeitsdaten: TcNcTrafoParameterExtCnc	18
3.3 Konfigurieren und Anmeldung der Transformation bei der CNC	23
4 Parametrierung	24
4.1 CNC-Parameter: Kanal und Werkzeug	24
4.1.1 Transformationsparameter des Werkzeugs.....	25
4.1.2 Kanalparameter	26
4.2 TcCOM-Parameter.....	30
5 Fehlerbehandlung und Diagnose	32
5.1 Fehlermeldung	32
5.2 Diagnosedaten	36
6 Verkettung von Transformationen, Multistep Transformationen	37
7 Erstellen einer Transformation	39
7.1 Erstellungsablauf.....	39
7.1.1 Neues Projekt anlegen.....	40
7.1.2 Transformation erstellen	43
7.1.3 Transformation einbinden	45
7.1.4 Transformation debuggen.....	47
7.1.5 Quellcodeerweiterung / Codierung	49
7.2 Unterschiede erweiterte Transformation / Standardtransformation	50
8 Zusätzliche Optionen der erweiterten Transformation	51
8.1 Versionskennung Transformationsinterface.....	51
8.2 Drehreihenfolge.....	51
8.3 Modulobehandlung der Achspositionen	53
8.4 Anwendung erweiterter Parameter	54
8.5 Verwendung erweiterter Optionen	56
9 Anwenden und Nutzen der Caller-ID	58
10 Anhang	61
10.1 Anregungen, Korrekturen und neueste Dokumentation.....	61

Stichwortverzeichnis..... 62

Abbildungsverzeichnis

Abb. 1:	Anbindung der kinematischen Transformation über TcCOM unter TwinCAT3	8
Abb. 2:	Funktion der kinematischen Transformation	8
Abb. 3:	Beispiel einer kinematischen Transformation	9
Abb. 4:	Koordinatensysteme im Detail	11
Abb. 5:	Koordinatensysteme im Detail	13
Abb. 6:	Zugriff auf kinematische Parameter	14
Abb. 7:	Modulobehandlung einer Achse	15
Abb. 8:	Dimensionierung Eingangs- und Ausgangskordinaten	16
Abb. 9:	Kinematische Transformation mit aktiver Schnittpunktberechnung	22
Abb. 10:	Kinematische Transformation mit inaktiver Schnittpunktberechnung	22
Abb. 11:	Transformationsparameter des Werkzeugs	25
Abb. 12:	Transformationsparameter des Kanals	27
Abb. 13:	Transformationsparameter über TcCOM	30
Abb. 14:	TMC-Editor	31
Abb. 15:	Verkettung von kinematischen Transformationen	37
Abb. 16:	Erstellung eines neuen Projekts	40
Abb. 17:	Konfiguration des neuen Projekts	40
Abb. 18:	Anlegen einer CNC-Konfiguration	41
Abb. 19:	Anlegen eines Kanals	41
Abb. 20:	Anlegen einer Achse	42
Abb. 21:	TwinCAT Treiber-Projekt anlegen	43
Abb. 22:	Transformations-Klasse anlegen	43
Abb. 23:	Benennung Transformations-Klasse	44
Abb. 24:	Treiber erstellen	45
Abb. 25:	Einbinden TcCOM-Objekt	45
Abb. 26:	Eigenschaften TcCOM-Objekts	46
Abb. 27:	Parametrieren der Transformation in Kanalparameterliste	47
Abb. 28:	Umstellen auf Debug-Konfiguration	47
Abb. 29:	Aktivieren Echtzeit-Debugging	47
Abb. 30:	Breakpoint in Transformation	48
Abb. 31:	Einstellung des Konstruktor nach Generieren mit TwinCAT3-Template	49
Abb. 32:	Angepasster Konstruktor wegen höherer Achszahl	49
Abb. 33:	Anpassung der Ein-/Ausgangszahlen	56
Abb. 34:	Schnittstellen Anpassung an unterschiedlichen Aufrufstellen	57
Abb. 35:	Anzeige der additiven Transformationsposition	59
Abb. 36:	Identifikation der Aufrufstellen der Transformation	60

1 Übersicht

Aufgabe

Mit dieser Funktionalität hat der Anwender die Möglichkeit eigene kinematische Transformationen zu integrieren und diese über eine Schnittstelle der CNC zugänglich zu machen.



Hinweis

Transformationen sind eine lizenzpflichtige Zusatzoption.



Versionshinweis

Diese Funktionalität ist ab TwinCAT 3 verfügbar.

Parametrierung

Über den Kanalparameter P-CHAN-00262 [▶ 28] muss die kinematische Transformation in der CNC angegeben werden

Programmierung

Die An- und Abwahl der kinematischen Transformation erfolgt im NC-Programm über den Befehl #TRAFO ON bzw. #TRAFO OFF. Die Auswahl welche Transformation verwendet werden soll erfolgt über den NC-Befehl #KIN ID[].

Obligatorischer Hinweis zu Verweisen auf andere Dokumente

Zwecks Übersichtlichkeit wird eine verkürzte Darstellung der Verweise (Links) auf andere Dokumente bzw. Parameter gewählt, z.B. [PROG] für Programmieranleitung oder P-AXIS-00001 für einen Achsparameter.

Technisch bedingt funktionieren diese Verweise nur in der Online-Hilfe (HTML5, CHM), allerdings nicht in PDF-Dateien, da PDF keine dokumentenübergreifenden Verlinkungen unterstützt.

2 Beschreibung

Die Anbindung der Transformation an die CNC kann in TwinCAT 3 über die TcCOM Infrastruktur stattfinden.

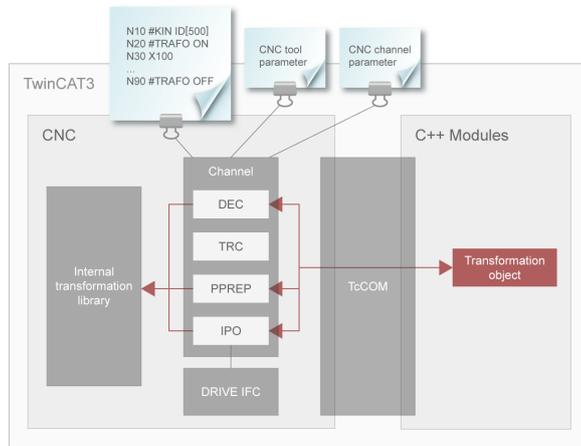


Abb. 1: Anbindung der kinematischen Transformation über TcCOM unter TwinCAT3



Achtung

Die Transformation wird in verschiedenen „zeitlichen“ Phasen der NC-Programmbearbeitung innerhalb eines NC-Kanals sowie in unterschiedlichen Kanälen evtl. zeitgleich eingesetzt. Deshalb muss die kinematische Transformation reentrant-fähig erstellt sein und darf keine globalen Daten verwenden.



Achtung

Die Verkettung der Vorwärts- und Rückwärtstransformation muss wieder die identische Ausgangsposition ergeben. Die übergebenen Positionen werden in [0.1 um] geliefert. In diesem Auflösungsbereich müssen die Transformationsergebnisse liegen.

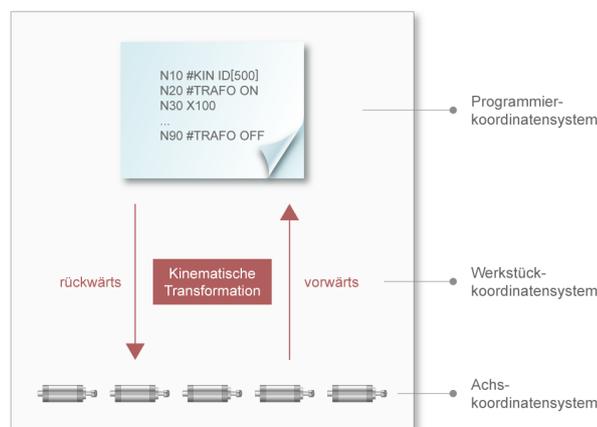


Abb. 2: Funktion der kinematischen Transformation

2.1

Allgemeines zur kinematischen Transformation (TRAFO)

Eine Übersicht der standardmäßig verfügbaren kinematischen Transformationen sind unter Kinematische Transformationen zu finden.

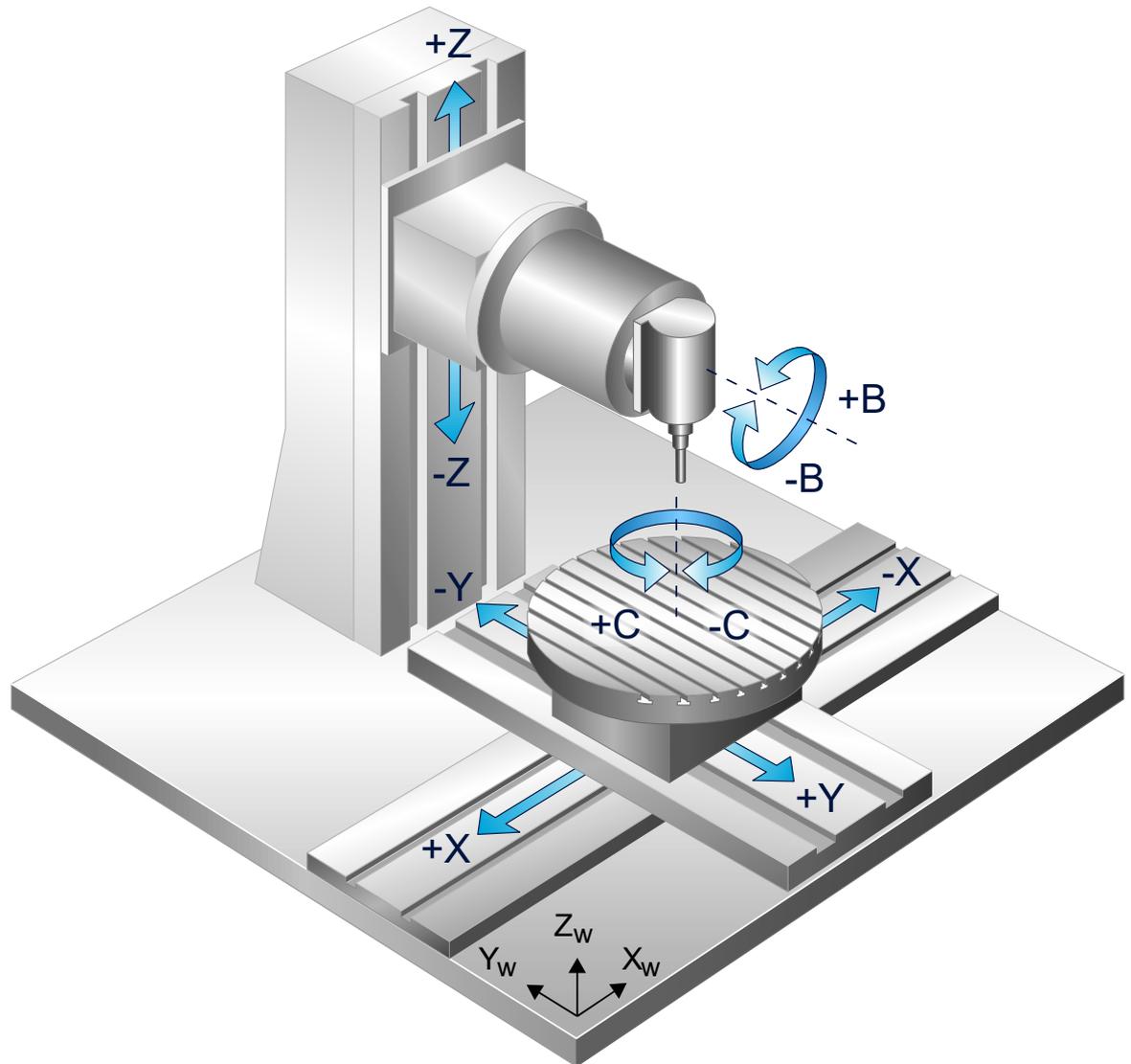


Abb. 3: Beispiel einer kinematischen Transformation

Kinematische Struktur

Um die Programmierung des Werkstücks zu vereinfachen, kapselt die kinematische Transformation die kinematische Struktur der Maschine und abstrahiert die Bewegungen in ein einfaches kartesisches Koordinatensystem.

Vorwärts-/Rückwärts-Transformation Vorwärts-/Rückwärts-Transformation

Je nach Kinematik der Maschinen benötigt die CNC für die Berechnung der Bewegungen die Transformation zwischen Achskoordinaten und Programmierkoordinaten. Mit Hilfe dieser kinematischen Transformation werden aus den physikalischen Positionen der Achsen die Koordinaten des NC-Programms (Vorwärtstransformation, ACS -> MCS) berechnet. Umgekehrt berechnet die Rückwärtstransformation die Achspositionen aus den programmierten NC-Positionen (MCS -> ACS).

An-/Abwahl

Die An- und Abwahl der Transformation findet über NC-Befehle im NC-Programm statt. Die Auswahl erfolgt über den NC-Befehl #TRAFO ON, die Abwahl über #TRAFO OFF.

Für die Auswahl der Transformation steht der NC-Befehl #KIN ID[] zur Verfügung, alternativ kann diese über P-CHAN-00032 [▶ 27] vorgelegt werden.

Erweiterbarkeit

Der Anwender hat die Möglichkeit eine eigene Transformation zu erstellen und diese der CNC unter einer ausgewählten Nummer (ID) zur Verfügung zu stellen. Hierzu stehen als Transformationsnummern der Bereich [500; 999] zur Verfügung. Der Bereich [65; 69] ist aus Kompatibilitätsgründen weiterhin verfügbar.

2.1.1 Koordinatensysteme

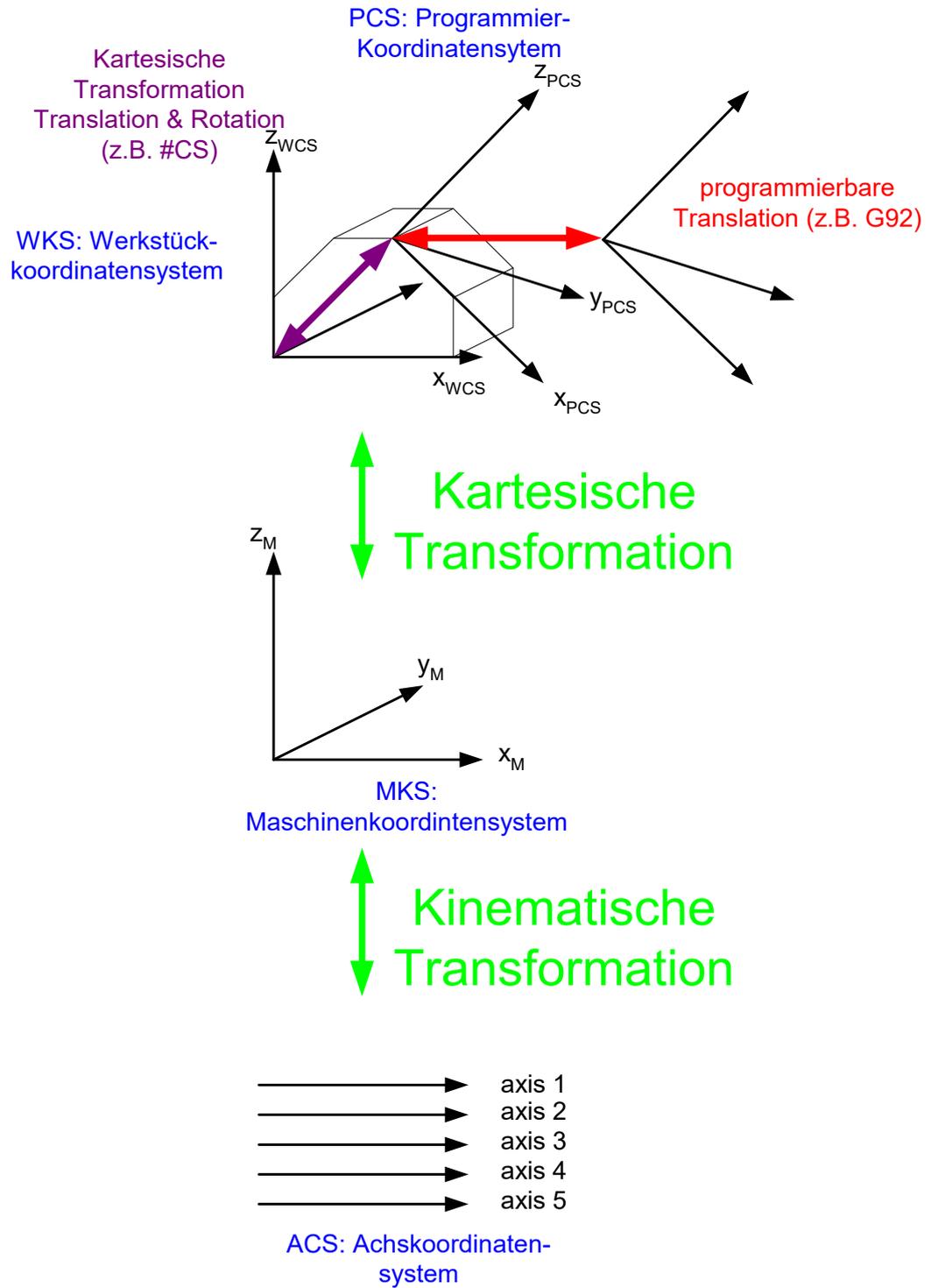


Abb. 4: Koordinatensysteme im Detail

Teilprogramm-Koordinatensystem PCS

Dieses Koordinatensystem wird in der Geometriebeschreibung nach DIN 66025 Programmiersyntax verwendet. Die Daten in einem Teilprogramm stellen Programmkoordinaten dar.

Werkstück-Koordinatensystem WCS

Dieses Koordinatensystem setzt an einem festen Punkt des Werkstücks an. Die Koordinatenbeschreibung des Werkstücks bezieht sich auf dieses System.

Das Werkstückkoordinatensystem ohne Verschiebungen wird als Basiskoordinatensystem verwendet (WCS_0).

Maschinenkoordinatensystem MCS

Das Maschinenkoordinatensystem repräsentiert ein abstraktes Koordinatensystem. Es wird vom Maschinenhersteller festgelegt. Alle anderen Koordinatensysteme beziehen sich auf dieses System.

Wenn die Maschine keinen kartesischen Achsaufbau hat (z.B. Roboter), ist das Maschinenkoordinatensystem nur virtuell.

Achsenkoordinatensystem ACS

Jede Achse besitzt ihr eigenes Koordinatensystem. Jede Achse ist entweder auf dem Maschinenbett selbst oder auf einer anderen Achse angebracht. Das bedeutet, dass das Maschinenbett oder die zugeordnete Achse die Grundlage bilden. Das Koordinatensystem einer Achse wird also entsprechend ihres Befestigungspunktes festgelegt.

2.1.2 Positionsverschiebungen

Verschiebungsverwaltung in den Transformationen PCS – WCS

Wenn eine Verschiebung zwischen den programmierten Koordinaten PCS und den wirklichen physikalischen Achsenpositionen ACS aktiviert werden muss, hat der Anwender verschiedene Möglichkeiten.

Die CNC-programmierbaren Verschiebungen (G54, G92, etc.) werden zwischen PCS und WCS berücksichtigt.

WCS – ACS

Falls die Kinematik einer Maschine Verschiebungen auf dem Achskoordinatensystem erfordert, wird dies innerhalb der Transformation berücksichtigt.

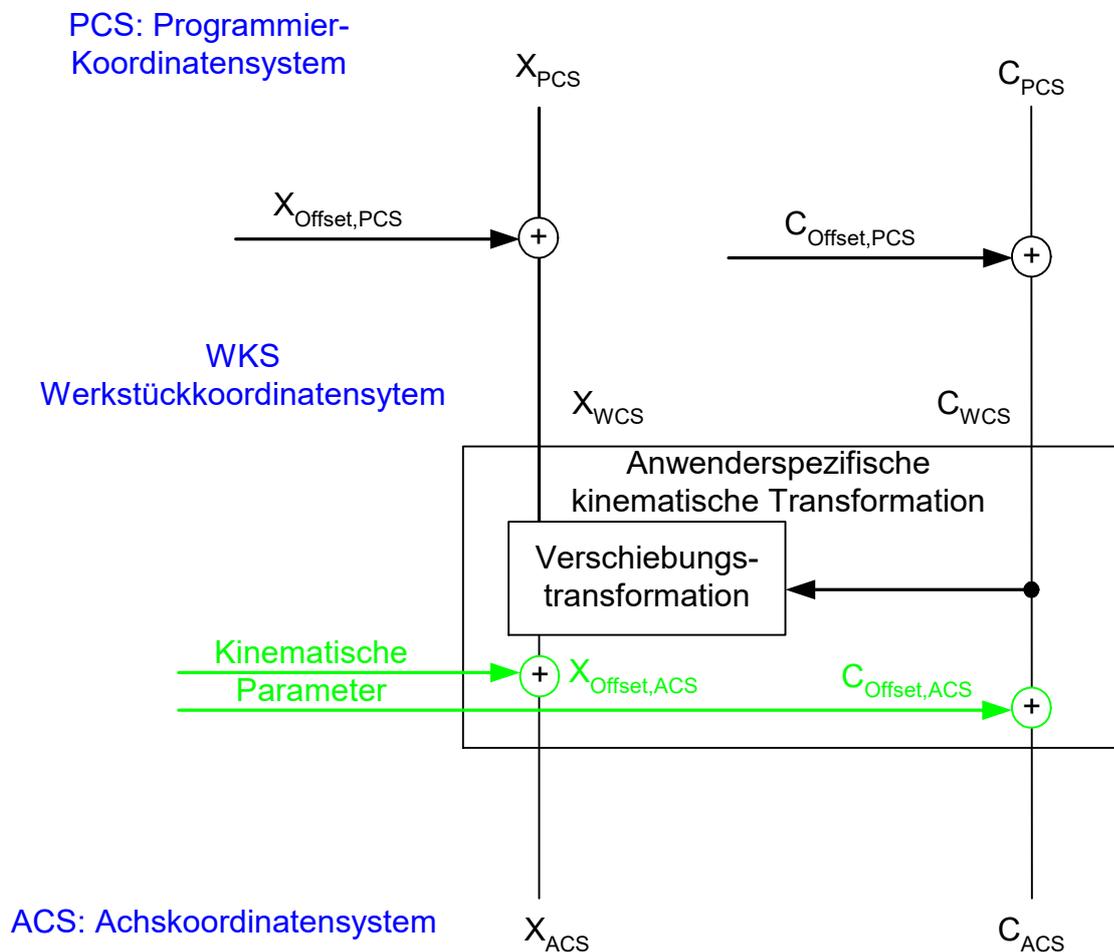


Abb. 5: Koordinatensysteme im Detail



Programmierbeispiel

Gebrauch achsenspezifischer Verschiebungen in kinematischer Transformation

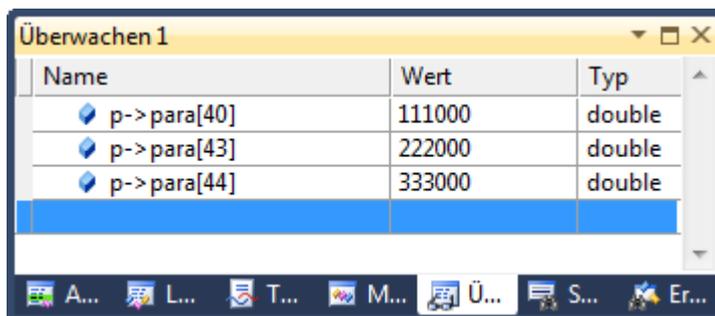
```

N010 G54                ; activate zero point offsets on ACS=PCS-level
N020 G0 X0 Y0 Z0 B0 C0 ; move to zero on PCS level
; ...
N090 G53                ; deactivate PCS-offsets
; ...
N120 V.G.KIN[500].PARAM[40] = <x_offset in [0.1 µm]>
N130 V.G.KIN[500].PARAM[43] = <b_offset in [0.0001 degree]>
N140 V.G.KIN[500].PARAM[44] = <c_offset> in [0.0001 degree]

N200 #KIN ID[500]      ; select kinematic type
N210 #TRAFO ON         ; ACS-offsets are considered inside trafo
N220 G01 X100 C90
;...
N240 G92 X400 C180 ; activate additional offset on PCS-level
N250 G01 X12 C0
...
N340 G56 ; activate additional offset on PCS-level
N350 G01 X2 C50
;...
N999 M30
    
```

Zugriff auf kinematische Parameter

Werden im CNC-Programm kinematische Parameter initialisiert, so werden diese als Eingabeparameter der Transformation an die Vorwärts-/Rückwärts-Algorithmen weitergeleitet (der verwendete Parameterindex ist transformationsspezifisch).



Name	Wert	Typ
p->para[40]	111000	double
p->para[43]	222000	double
p->para[44]	333000	double

Abb. 6: Zugriff auf kinematische Parameter

2.1.3 Modulo-Einstellung der Achsen

MCS – ACS

In Abhängigkeit der Achseigenschaften muss die kinematische Transformation die Modulorechnung der Positionen festlegen. Die Modulobehandlung innerhalb der Transformation muss das gleiche Modulointervall verwenden wie die aufrufende CNC Funktion.

Die angegebene MCS-Moduloeinstellung wird automatisch von der aufrufenden CNC-Funktionalität übernommen:

MCS linear / mod[-180;180]

Die angegebene ACS-Moduloeinstellung wird für eine Plausibilitätsprüfung verwendet. Die CNC prüft, ob die Einstellung mit der konfigurierten Eigenschaft der Achse übereinstimmt.

ACS linear / mod[-180;180] / mod[0;360]

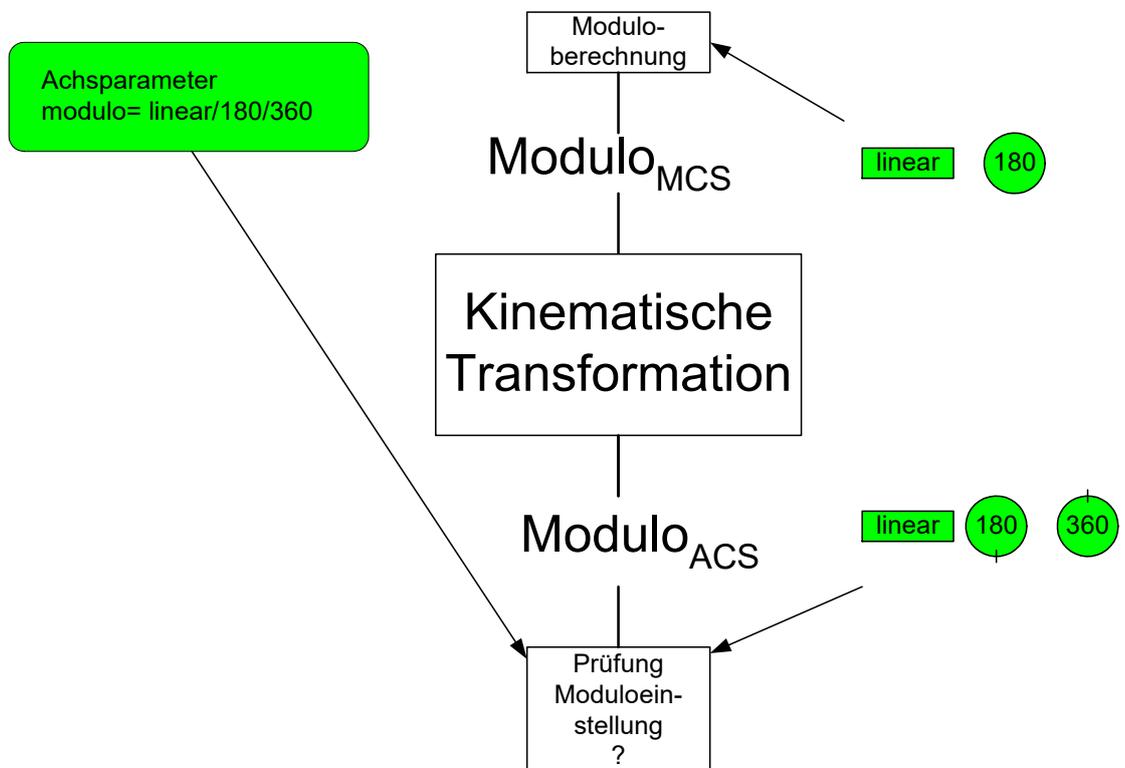


Abb. 7: Modulobehandlung einer Achse

3 Anbindung Transformation via TcCOM

3.1 Methoden der Transformation

Folgende Methoden sind bei Erstellung einer Transformation zu implementieren (TcNcKinematic-Interfaces.h).

- virtual HRESULT TCOMAPI **Forward** (PTcCncTrafoParameter p)=0;
- virtual HRESULT TCOMAPI **Backward** (PTcCncTrafoParameter p)=0;
- virtual HRESULT TCOMAPI **TrafoSupported** (PTcCncTrafoParameter p, bool fwd)=0;
- virtual HRESULT TCOMAPI **GetDimensions** (PULONG pForwardInput, PULONG pForwardOutput)=0;

Forward	Transformation der Achspositionen in das Programmierkoordinatensystem.
PTcCncTrafoParameter *p	Aktuelle Parameter der Transformation

Backward	Transformation der Programmierkoordinaten in das Achskoordinatensystem.
PTcCncTrafoParameter *p	Aktuelle Parameter der Transformation

GetDimension	Bei Anwahl der Transformation wird die Abfrage der Konfiguration (notwendigen Achszahlen) einmalig durchgeführt.
ULONG * pForwardInput	Anzahl der Eingangskoordinaten der Vorwärtstransformation (= Anzahl der Ausgangskoordinaten der Rückwärtstransformation)
ULONG * pForwardOutput	Anzahl der Ausgangskoordinaten der Vorwärtstransformation (= Anzahl der Eingangskoordinaten der Rückwärtstransformation)

TrafoSupported	Initialisierung der Transformation und Abfrage von Optionen
PTcCncTrafoParameter *p	Aktuelle Parameter der Transformation
bool fwd	



Hinweis

Im Konstruktor der Klasse „ITcCncTrafo“ müssen zur Dimensionierung der Eingangs- und Ausgangskoordinaten die entsprechenden Member-Variablen initialisiert werden.

```

////////////////////////////////////
// Constructor
MyKinTrafo::MyKinTrafo(): m_forwardNbrIn(5), m_forwardNbrOut(5)
{

```

Abb. 8: Dimensionierung Eingangs- und Ausgangskoordinaten

3.2 Arbeitsdaten (Instanzdaten) der Transformation

Definition der Arbeitsdaten

Die Implementierung der Transformation kann beliebige Parameter als Arbeitsdaten bereitstellen. Hierbei ist darauf zu achten, dass die Transformation in mehreren zeitlichen Phasen der CNC verwendet wird. Aus diesem Grunde muss die CNC reentrant geschrieben sein. Die Arbeitsdaten dürfen also keinen Zustand der Transformation beinhalten, welcher für die nachfolgende Berechnung weiterverwendet wird

3.2.1 Basis Arbeitsdaten: TcNcTrafoParameter

Parameter der Methoden

Type = EcNcTrafoParameter_Base

Die Parameter für die einzelnen Methoden werden über folgende Struktur **TcNcTrafoParameter** (TcNcKinematicsInterfaces.h) gekapselt übergeben.

```
EcNcTrafoParameter type;
ULONG dim_i;      // dim of input vectors (i, d_i, dd_i)
ULONG dim_o;      // dim of output vectors (o, d_o, dd_o, torque)
ULONG dim_para;  // dim of additional parameter (para)

const double* i;   // input values parameter (dim_i)
const double* d_i;
const double* dd_i;

double* o;         // output values parameter (dim_i)
double* d_o;
double* dd_o;

double* torque;
const double* para; // additional parameter (dim_p)

double payload;    // weight in kg
double tool_len;   // actual tool length in [mm]
```

Bemerkung:

Die kursiven Variablen werden von der CNC nicht verwendet.

3.2.2 Erweiterte Arbeitsdaten: TcNcTrafoParameterExtCnc

Parameter der Methoden

Type = EcNcTrafoParameter_ExtCnc

Die Parameter für die einzelnen Methoden werden über folgende erweiterte Struktur **TcCncTrafoParameter** übergeben. Die von der CNC bereitgestellte Datenstruktur wird anhand des Parametertyps kenntlich gemacht.

Type = EcNcTrafoParameter_ExtCnc

```
struct TcCncTrafoParameter : public TcNcTrafoParameter, TcCncParam
{
    unsigned short kin_id; // in: used kinematic ID
    unsigned long control; // in: control trafo calculation, e.g. EcCncTrafoCtrl_cartesianTrafoInactive
    EcCncTrafoOption ret_option; // out: select option of transformation during TrafoSupported()
    TcCncVersion CncInterfaceVersion; // Interface version TcCncVersionMajor.TcCncVersionMinor

    // orientation
    EcCnc_TrafoOriModeActual actual_orientation_mode; // Treatment of orientation, actual rotation sequence
    EcCnc_TrafoModeSupported supported_modes; // modes supported by the TcCOM transformtion
};
```

Hinweis:

Das Strukturelement EcCnc_TrafoModeSupported supported_modes ersetzt das bisherige Element EcCnc_TrafoOriModeSupported supported_orientation_modes. Aus Gründen der Abwärtskompatibilität wird dieses Datum weiterhin unterstützt.

```
// modulo configuration
ULONG dim_modulo; // dim of modulo vector
EcCnc_McsModulo * mcs_modulo;
EcCnc_AcsModulo * acs_modulo;
```

Identifikation des Aufrufers

Die aktive kinematische Transformation wird aktuell an mehreren Stellen in der CNC verwendet. Die unterschiedliche Aufrufstelle wird in den übergebenen Arbeitsdaten der Transformation vermerkt.

- 0 : EcCncTrafoCallerID_Undefined
- 1 : EcCncTrafoCallerID_DeCode
- 2 : EcCncTrafoCallerID_ToolRadiusCorrection
- 3 : EcCncTrafoCallerID_PathPreparation
- 4 : EcCncTrafoCallerID_Interpolation
- 5 : EcCncTrafoCallerID_Display
- 6 : EcCncTrafoCallerID_BlockSearch



Hinweis

Mit der Kennung des Aufrufers (`caller_id`) kann die Transformation an unterschiedlichen Stellen mit Varianten durchgerechnet werden.

Beispiele siehe Anwenden und Nutzen der Caller-ID [► 58]

Optionen der Transformation

Während der Initialisierung (method `TrafoSupported`) der Transformation können individuelle Optionen der CNC angewählt werden. Diese Optionen ändern die CNC Verwaltung der Schnittstelle und stellen evtl. zusätzliche Parameter bereit. Die einzelnen Optionen sind durch die CNC vordefiniert und müssen zur entsprechenden Transformation passen. Folgende Optionen sind verfügbar:

```
0 : EcCncTrafoOption_None
1 : EcCncTrafoOption_Interpolation_AddInput
```

Kontroll-Input

Folgende Informationen werden zyklisch an die kinematische Transformation übergeben

```
0x0000 0001 EcCncTrafoCtrl_cartesianTrafoInactive
```

Kartesische Transformation in der CNC ist inaktiv. Es erfolgt die direkte Winkelvorgabe durch die CNC.

```
0x0000 0010 EcCncTrafoCtrl_RTCPMode
```

Für eine singuläre kinematische Transformation wird der RTCP-Modus angefordert. Es erfolgt die direkte Winkelvorgabe durch die CNC auch in der singulären Achsstellung.

Die beiden oben aufgeführten Steuerinformationen können z.B. zur Umschaltung der Behandlung in der Singularität innerhalb der TcCOM-Transformation verwendet werden. Es erfolgt in diesen Fällen eine direkte Winkelvorgabe durch die CNC.

Im anderen Fall (beide oben aufgeführte Steuerinformationen nicht gesetzt) erfolgt die Belegung der Winkeleingangswerte über einen Werkzeugrichtungsvektor in Winkeldarstellung z.B. bei einem CA-Kopf: C: +/-180 Grad, A: 0...90 Grad.

Versionsnummer CNC-Interface

Im Datum `TcCncVersion` überträgt die CNC die Versionsnummer des von ihr verwendeten Transformationsinterfaces:

```
struct TcCncVersion
{
    Long    major;
    Long    minor;
};
```

Weitere Informationen zur Versionsnummer: Versionskennung Transformationsinterface [► 51]

Rotationsreihenfolge

Im Datum `actual_rotation_mode` überträgt die CNC die aktive Drehreihenfolge der Orientierungsachsen:

```
EcCncTrafoOri_None = 0
EcCncTrafoOri_YPR  = 1
EcCncTrafoOri_CBC1 = 2
EcCncTrafoOri_CBA  = 3
EcCncTrafoOri_CAB  = 4
EcCncTrafoOri_AB   = 5
EcCncTrafoOri_BA   = 6
EcCncTrafoOri_CA   = 7
EcCncTrafoOri_CB   = 8
```

Die in der Transformation unterstützen Drehreihenfolgen werden der CNC im Datum `supported_rotation_modes` (siehe auch Drehreihenfolge [▶ 51]) mitgeteilt:

```
typedef struct _EcCnc_TrafoModeSupported
{
    unsigned long    f_YPR      : 1;
    unsigned long    f_CBC1    : 1;
    unsigned long    f_CBA     : 1;
    unsigned long    f_CAB     : 1;
    unsigned long    f_UniqueTrafo : 1;
    unsigned long    f_AB      : 1;
    unsigned long    f_BA      : 1;
    unsigned long    f_CA      : 1;
    unsigned long    f_CB      : 1;
    unsigned long    f_SingularOri : 1;
} EcCnc_TrafoModeSupported;
```

Eindeutige TcCOM-Transformation

Das Flag `f_UniqueTrafo` im Datum `supported_modes` ermöglicht dem Anwender die TcCOM-Transformation als eindeutig in Vorwärts- und Rückwärtsrichtung zu markieren. Das Flag kann in der Methode `TrafoSupported` vom Anwender gesetzt werden. Die CNC behandelt TcCOM-Transformationen standardmäßig als nicht eindeutig. Beim Initialisieren der Transformation wird das Flag `f_UniqueTrafo` geprüft.

Das Setzen des Flags beschleunigt alle Vorgänge, bei denen die CNC Positionen der Antriebe lesen muss, um einige Takte. Solche Vorgänge sind zum Beispiel das An- und Abwählen der Transformation, das Ändern von Koordinatensystemen bei aktiver TcCOM-Transformation oder die Verwendung von V.A.ACS.ABS Variablen.

Beispielcode zum Setzen des Flags:

```
virtual HRESULT TCOMAPI TrafoSupported(PTcCncTrafoParameter p, bool fwd)
{
    p->supported_modes.f_UniqueTrafo = TRUE;
    return S_OK;
};
```

Singuläre TcCOM-Transformation

Das Flag `f_SingularOri` im Datum `supported_modes` ermöglicht dem Anwender die TcCOM-Transformation als Kinematik mit singulärer Kopfstellung zu markieren. Das Flag kann in der Methode `TrafoSupported` vom Anwender gesetzt werden. Die CNC aktiviert dann bei Fünfachskinetiken mit CA-, CB-Kopf die Singularitätsbehandlung.

Beispielcode zum Setzen des Flags:

```
virtual HRESULT TCOMAPI TrafoSupported(PTcCncTrafoParameter p, bool fwd)
{
    p->supported_modes.f_SingularOri = TRUE;
    return S_OK;
};
```

Modulo-Einstellungen

Die Dimension der achsspezifischen Objekte `mcs_modulo` und `acs_modulo` stellt die CNC im Objekt `dim_modulo` bereit. Die Modulo-Behandlung im MCS Koordinatensystem wird im achsspezifischen Datum `mcs_modulo` der CNC mitgeteilt:

```
EcCnc_McsModulo_None      = 0,
EcCnc_McsModulo_180_180  = 1,
```

Die erwartete ModuloEinstellung einer Achse im ACS-Koordinatensystem kann der CNC im Datum `acs_modulo` mitgeteilt werden:

```
EcCnc_AcsModulo_None      = 0,
EcCnc_AcsModulo_180_180  = 1,
EcCnc_AcsModulo_0_360    = 2,
```



Beispiel

Schnittpunktberechnung ausschalten bei inaktivem #CS

Soll z.B. die kinematische Transformation variieren je nachdem ob eine übergeordnete kartesische Transformation aktiv ist, so kann dies anhand des Eingangsbits selektiert werden. Dies wird durch die Steuerung angezeigt.

Kinematische Transformation mit aktiver Schnittpunktberechnung (`EcCncTrafoCtrl_cartesianTrafoInactive` gelöscht)

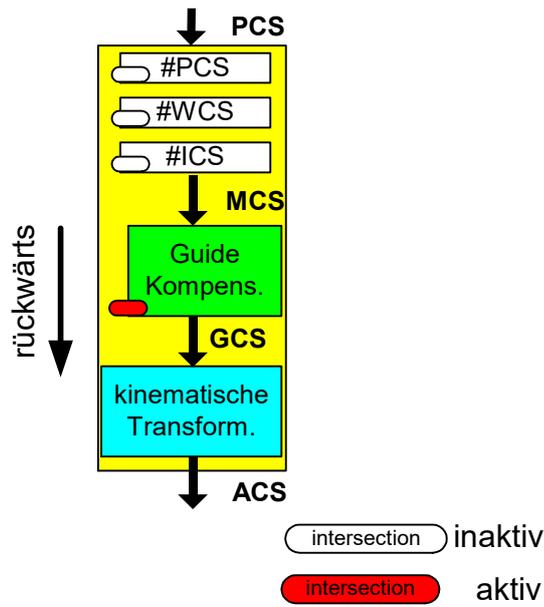


Abb. 9: Kinematische Transformation mit aktiver Schnittpunktberechnung

Kinematische Transformation mit inaktiver Schnittpunktberechnung
(EcCncTrafoCtrl_cartesianTrafoInactive gesetzt)

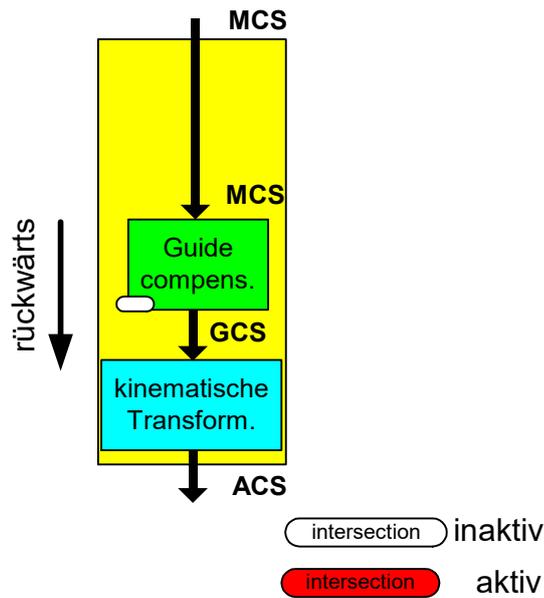


Abb. 10: Kinematische Transformation mit inaktiver Schnittpunktberechnung

3.3 Konfigurieren und Anmeldung der Transformation bei der CNC

Registrieren der Transformation

Zur Anmeldung eines TcCOM werden folgende Daten verwendet (TcCncServices.h)

- Type 1 (s. TCCNC_REGISTEROBJECT_TYPE_TRAFO) fest vorgegeben
- Group Kanalnummer der Transformation [1;12] wählbar bei Konfiguration
- Index Nummer der Kinematik [500; 999] wählbar bei Konfiguration, aus Kompatibilitätsgründen auch [65;69]

Die Anmeldung der Transformation erfolgt über nachfolgendes TcCOM-Interface, welches in der Datei TcCncInterfaces.h definiert ist.

- virtual HRESULT TCOMAPI **RegisterObject**
(TcCncRegisterObject& id, ITcUnknown* ipUnk)=0;
- virtual HRESULT TCOMAPI **UnregisterObject**
(TcCncRegisterObject& id)=0;

Manuelle Bereitstellung der Transformation

Nach Erzeugen der Transformation müssen zwei Dateien für die Verwendungen bereitgestellt werden.

Die Beschreibung der Transformation erfolgt in der TMC-Datei TcCncTrafo1.tmc und wird aus dem Quellcodeverzeichnis unter folgendem Zielverzeichnis abgelegt.

```
<TwinCAT>\3.1\CustomConfig\Modules
```

Der erzeugte Gerätetreiber (z. Bsp. TcCncTrafo1.sys) wird, je nach System, also 64 Bit oder 32 Bit, von

```
<TwinCAT>\3.1\SDK\_products\TwinCAT RT (x86)\Release
```

```
<TwinCAT>\3.1\sdk\_products\TwinCAT RT (x64)\Release
```

unter

```
<TwinCAT>\3.1\Driver\AutoInstall
```

abgelegt.

Für das Debugging werden der erzeugte Gerätetreiber (z. Bsp. TcCncTrafo1.sys) und die Symbol-Datei (z. Bsp. TcCncTrafo1.pdb), je nach System, von

```
<TwinCAT>\3.1\SDK\_products\TwinCAT RT (x86)\Debug
```

```
<TwinCAT>\3.1\SDK\_products\TwinCAT RT (x64)\Debug
```

unter

```
<TwinCAT>\3.1\Driver\AutoInstall
```

abgelegt.

Konfiguration der Transformation

Beim Konfigurieren der Transformation wird das TcCOM-Objekt im Systemmanager angewählt und der Kanal (group) und die Transformations-ID (index) initialisiert.

Die Vorgehensweise ist in Transformation einbinden [► 45] veranschaulicht.

4 Parametrierung

Transformationsparameter

Die Transformation kann durch den Anwender über kanal- und/oder werkzeugspezifische Werte parametrierbar werden. Die Bedeutung der Parameter hängt dabei rein von der Implementierung der Transformation ab. Die Parameter können in folgenden Bereiche initialisiert werden und haben unterschiedliche Gültigkeitsdauern:

- CNC Kanal
Die Kanalparameter können in der Konfiguration der CNC pro Kanal eingestellt werden und gelten bis zum Aktualisieren dieser Kanaldaten (Download im Systemmanager).
- Werkzeug
Parameter des Werkzeugs werden bei der Werkzeuanforderung mitgeliefert (D-Programmierung im NC-Programm) und gelten solange dieses Werkzeug im NC-Programm angewählt ist. Die Parameter können individuell pro Werkzeug initialisiert werden.
- TcCOM
Bei der Konfiguration der Kinematik können globale Parameter angegeben werden. Diese gelten solange die Transformation geladen ist, d.h. solange TwinCAT aktiv ist.

4.1 CNC-Parameter: Kanal und Werkzeug

Die Transformationsparameter für den CNC-Kanal und die Werkzeuge werden der Transformation per Übergabeparameter (Zeiger p auf Struktur TcNcTrafoParameter) bereitgestellt.

Name	Wert
 p->para[0]	1088000
 p->para[1]	1987000
 p->para[2]	342000

Falls ein Werkzeug angewählt ist (D-Wort s. [PROG//Werkzeuggeometriekorrektur] werden die Summe der Kinematikparameter aus der Kanalparameterliste und des Werkzeugs übergeben.

Beispiel:

```
Kanalparameterliste:  trafo[1].param[2]           300000
Angewähltes Werkzeug: wz[5].kinematic.param[2]   500000
```

Übergebener Transformationsparameter: p->para[2] = 800000



Achtung

Der Transformationsparameter mit Index 0 (trafo[..].param[0]) wirkt immer in Richtung der 3. Hauptachse (normalerweise Z-Achse) und wird zusätzlich in die Werkzeuglänge eingerechnet. Falls für die Transformation die unveränderte Länge des Werkzeugs benötigt wird, sollte dieser Parameter daher nicht verwendet werden.

4.1.1 Transformationsparameter des Werkzeugs

Die Werkzeugparameter können in der CNC oder in einer externen Werkzeugverwaltung (z.B. in der SPS) verwaltet werden. Falls die Werkzeugparameter in der CNC verwaltet werden d.h. der Kanalparameter `ext_wzv_vorhanden = 0` (s. P-CHAN-00016) gesetzt ist, ist im TwinCAT3-Projekt der Reiter „Tool Para“ mit der Werkzeugparameterliste verfügbar.



Beispiel

Parametrierbeispiel- Werkzeug

Parametrierung in der Werkzeugliste, siehe P-TOOL-00009

```
wz[5].kinematic.param[0] 1538000 # Kopfversatz 1: 153,8 mm
wz[5].kinematic.param[1] 25000 # Kopfversatz 2: 2,5 mm
wz[5].kinematic.param[2] 0 # Kopfversatz 3: 0 mm
wz[5].kinematic.param[5] 900000 # Kopfversatz 6; 90 mm
```



Hinweis

Die Kinematikparameter des Werkzeugs können nur für die Standardstufe = 0 angegeben werden.

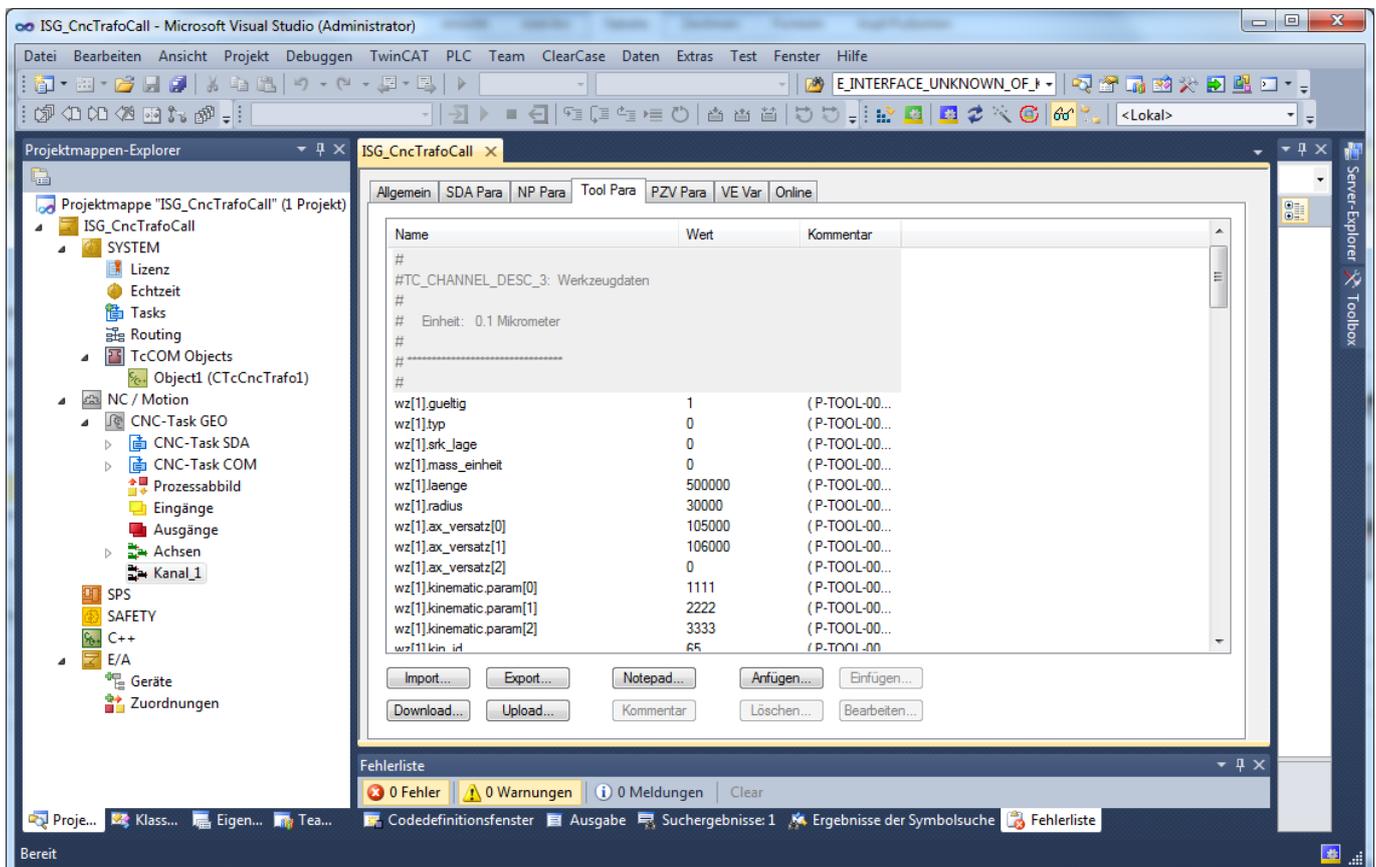


Abb. 11: Transformationsparameter des Werkzeugs

P-TOOL-00009	Kinematikparameter
Beschreibung	Diese Parameter dienen zur werkzeugabhängigen Parametrierung der kinematischen Transformation (RTCP / TLC / TOOL ORI CS [PROG]). Die Belegung wird applikationsspezifisch festgelegt.
Parameter	wz[j].kinematic.param[j] mit j = 0 ... 74 (Maximale Anzahl Kinematikparameter, applikationsspezifisch, Syntax ab V263)
Datentyp	SGN32
Datenbereich	MIN(SGN32) < param[j] < MAX(SGN32)
Dimension	0.1µm
Standardwert	0
Anmerkungen	<p><i>wz[j].kinematic.wz_kopf_ersatz[j]</i> (Syntax bis V260)</p> <p>Es besteht zusätzlich die Möglichkeit, die Versätze für jede Kinematik in den Kanalparametern P-CHAN-00094 einzutragen. Ist ein Element in beiden Listen belegt, so erfolgt in der CNC eine Addition der angegebenen Werte.</p> <p>Nähere Informationen zur Parametrierung der kinematischen Transformation bei der 5-Achs-Bearbeitung finden sich in [KITRA] und [PROG].</p> <p>Parametrierbeispiel:</p> <pre>wz[5].kinematic.param[0] 1538000 #Kopfversatz 1: 153,8 mm wz[5].kinematic.param[1] 25000 #Kopfversatz 2: 2,5 mm wz[5].kinematic.param[2] 0 #Kopfversatz 3: 0 mm wz[5].kinematic.param[5] 900000 #Kopfversatz 6: 90 mm</pre>

4.1.2

Kanalparameter



Beispiel

Parametrierbeispiel- Kanal

```
# Festlegung der Standardtransformation
kinematik_id          500          ( P-CHAN-00032)
# -----
# -- TcCOM Transformation
#
trafo[0].id           500          ( P-CHAN-00262 )
trafo[0].param[0]    1088000      ( P-CHAN-00263 )
trafo[0].param[1]    342000
trafo[0].param[2]    150
trafo[0].param[3]    0
trafo[0].param[4]    0
trafo[0].param[5]    0
trafo[0].param[6]    0
#
trafo[1].id           9
trafo[1].param[0]    120000
trafo[1].param[1]    100000
trafo[1].param[2]    120
trafo[1].param[3]    0
trafo[1].param[4]    0
```

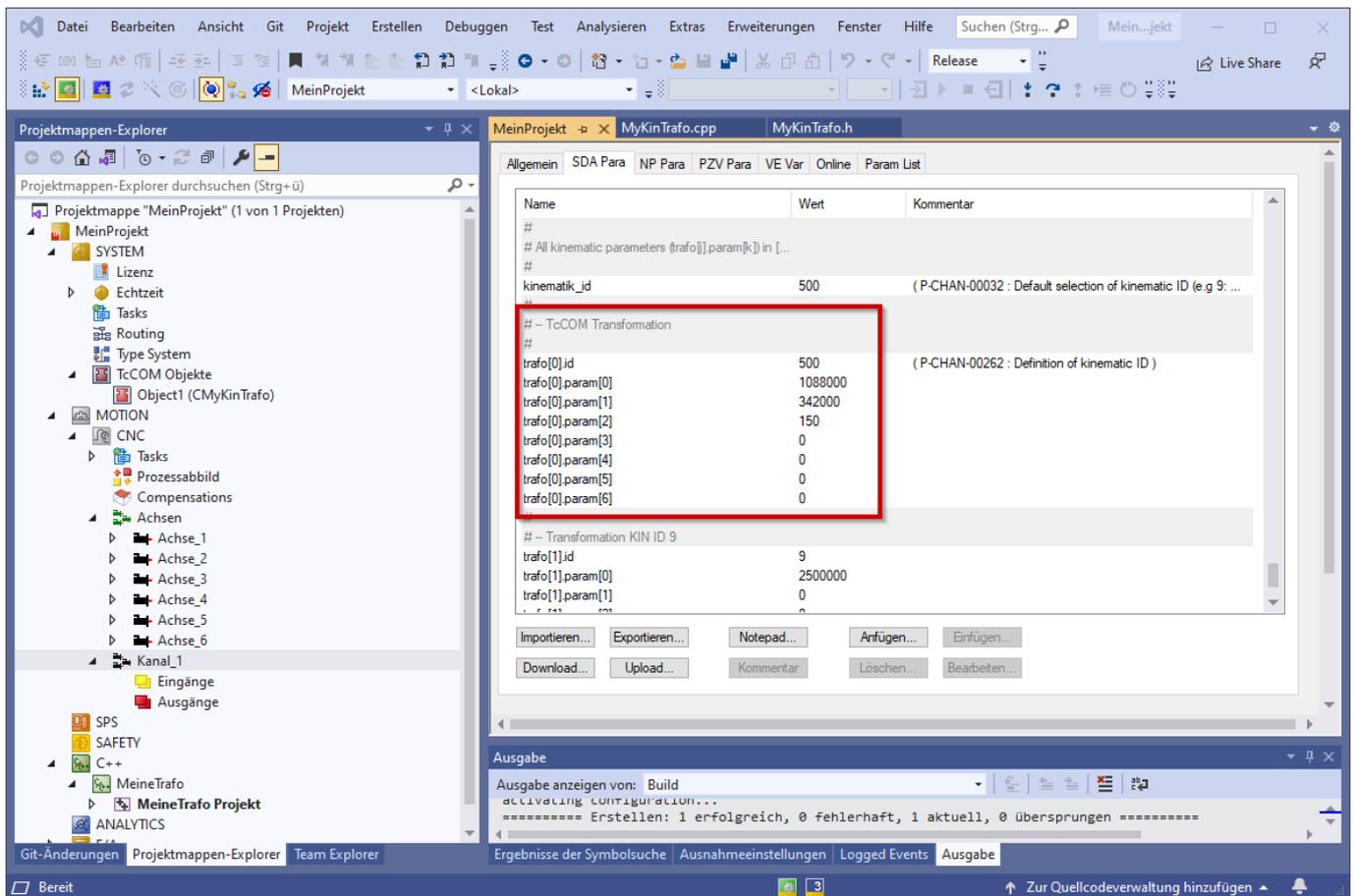


Abb. 12: Transformationsparameter des Kanals

Beschreibung der Kanalparameter

P-CHAN-00032	Auswahl der kinematischen Standardtransformation (Kinematiktyp)
Beschreibung	Die Kinematik-ID dient zur Identifizierung der in der Steuerung implementierten, maschinen- bzw. werkzeugkopfspezifischen Kinematiktypen. Über den Parameter wird die Standardeinstellung für die zu verwendende kinematische Transformation festgelegt.
Parameter	kinematik_id
Datentyp	UNS16
Datenbereich	$1 \leq \text{kinematik_id} < 1000$
Dimension	----
Standardwert	0
Anmerkungen	Parametrierbeispiel: Nach Steuerungshochlauf ist die Transformation mit ID 2 gültig. <i>kinematik_id 2</i> Nähere Informationen zu Maschinenkinematiken finden sich in [KITRA] und [PROG].

P-CHAN-00262	Definition der Kinematik-ID für Transformationen
Beschreibung	<p>Die Kinematik-ID identifiziert als Element des Datensatzes der Kinematikparameter die zugehörige Transformation.</p> <p>Die Definition kann sowohl für ein- als auch für mehrstufige Transformationen sowie für PCS-Transformationen erfolgen.</p>
Parameter	trafo[j].id kin_step[i].trafo[j].id (mehrstufige Transformationen) trafo_pcs[i].id (PCS-Transformation *)
Datentyp	UNS16
Datenbereich	1 ... MAX(UNS16)
Dimension	----
Standardwert	0
Anmerkungen	Parametersyntax ab V300 *Funktionalität PCS-Transformation ist verfügbar ab V3.1.3110.
P-CHAN-00263	Definition der Kinematikparameter für Transformationen
Beschreibung	<p>In dieser Struktur werden für jede Transformation die spezifischen Kinematikversatzmaße eingetragen.</p> <p>Das Spezifizieren der Versatzmaße kann sowohl für ein- als auch für mehrstufige Transformationen sowie für PCS-Transformationen erfolgen.</p>
Parameter	trafo[j].param[k] oder mit k = 0 bis 73 (Maximale Anzahl Kinematikparameter) kin_step[i].trafo[j].param[k] (mehrstufige Transformationen) trafo_pcs[i].param[k] (PCS-Transformation *)
Datentyp	REAL64
Datenbereich	----
Dimension	0.1 µm bzw. 0.0001 Inch
Standardwert	0
Anmerkungen	<p>Es besteht zusätzlich die Möglichkeit, Kinematikparameter in die Werkzeugdatenliste P-TOOL-00009 einzutragen (diese sind dann unabhängig von der Kinematik nur bei angewähltem Werkzeug relevant).</p> <p>Ist ein Kinematikparameter in beiden Listen belegt, so erfolgt in der NC eine Addition der angegebenen Werte. Dies gilt nur für die Transformationsstufe 1.</p> <p>Für die Transformationsstufe 2 können in den Werkzeugdaten keine zusätzlichen Kinematikparameter eingetragen werden!</p> <p>Nähere Informationen zur Parametrierung einer kinematischen Transformation finden sich in [KITRA] und [PROG].</p> <p>(Parametersyntax ab V300)</p> <p>*Funktionalität PCS-Transformation ist verfügbar ab V3.1.3110.</p>

P-CHAN-00829	Kinematiktyp für Transformationen
Beschreibung	Mit diesem Parameter wird die Art der Kinematik festgelegt. Eine Übersicht über die Kinematiken ist in Kinematische Transformationen zu finden. Die Festlegung kann sowohl für ein- als auch für mehrstufige Transformationen sowie für PCS-Transformationen erfolgen.
Parameter	trafo[j].type kin_step[i].trafo[j].type (mehrstufige Transformationen) trafo_pcs[i].type (PCS-Transformation *)
Datentyp	UNS16
Datenbereich	0 ... MAX(UNS16)
Dimension	---
Standardwert	0
Anmerkungen	Sobald eine Kinematik-ID (P-CHAN-00262 [► 28]) ungleich 0 konfiguriert ist und der Kinematiktyp 0 ist, wird dem Kinematiktyp der Wert der Kinematik-ID zugewiesen. Verfügbar ab V3.1.3080.09 *Funktionalität PCS-Transformation ist verfügbar ab V3.1.3110.

Die für die Transformation benötigten TcCOM-Parameter können im TMC-Editor festgelegt werden:

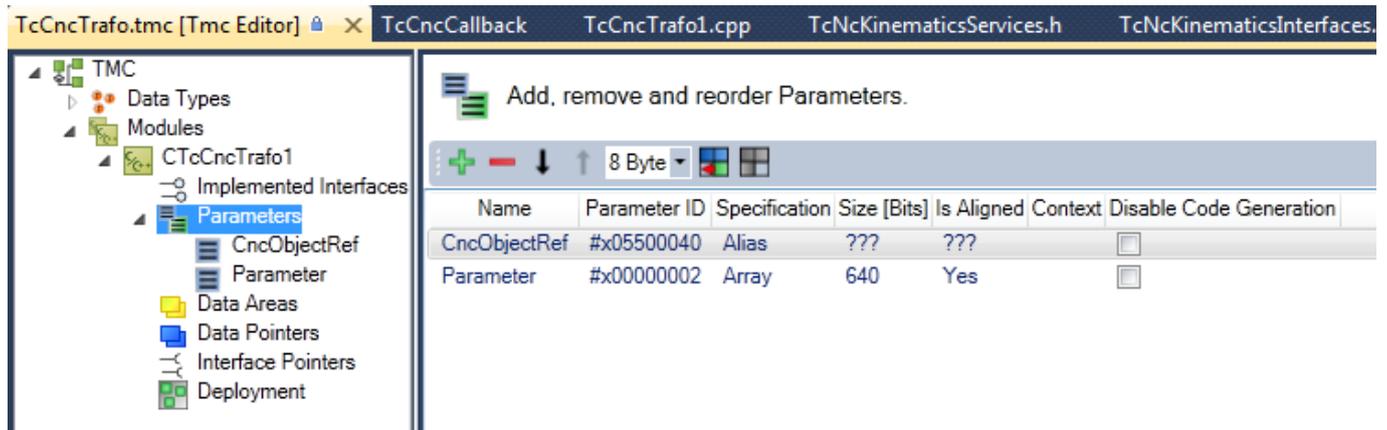


Abb. 14: TMC-Editor

Über den TwinCAT TMC Code Generator (Rechtsklick auf das TcCncTrafo-Projekt -> TwinCAT TMC Code Generator) werden die im TMC-File vorhandenen Parameter automatisch der Klasse für die Transformation CtcCncTrafo1 als Member-Variablen z.B. m_Parameter hinzugefügt und können so in der Vorwärts- bzw. Rückwärtstransformation verwendet werden.

```

///

```

5 Fehlerbehandlung und Diagnose

5.1 Fehlermeldung

Verwaltungsfehler	
	Im Falle eines Fehlers setzt die CNC eine Fehlermeldung ab und die aktuelle Bearbeitung des CNC-Kanals wird abgebrochen.
292019	programmierte Transformation ist nicht geladen, d.h. evtl. in TwinCAT nicht konfiguriert
292020	Speicher für Transformation zu gering (Systemfehler)
292021	intern wurde eine unbekannte Kanalnummer übergeben (Systemfehler)
292022	Programmierte Transformation ist intern nicht bekannt (geladen), d.h. evtl. in TwinCAT nicht richtig konfiguriert
292023	Die Rückwärtstransformation ist nicht invers zur Vorwärtstransformation.
292030	Fehler bei Abfrage der Konfigurationsdaten der kinematischen Transformation (s. GetDimension())
292031	Fehler bei Initialisierung der kinematischen Transformation (s. TrafoSupported()).
292032	Fehler bei kinematischer Vorwärtstransformation (s. Forward()).
292033	Fehler bei kinematischer Rückwärtstransformation (s. Backward()).
292034	Aktuelle MKS-Eingangsposition der kinematischen Vorwärtstransformation.
292035	Aktuelle WKS-Ausgangsposition der kinematischen Vorwärtstransformation.
292036	Aktuelle WKS-Eingangsposition der kinematischen Rückwärtstransformation.
292037	Aktuelle MKS-Ausgangsposition der kinematischen Vorwärtstransformation.
292044	Das Transformationsinterface der CNC ist zu alt und passt nicht zum TcCOM-Objekt.
292045	Die gewählte Orientierungsart wird von der Transformation nicht unterstützt.



Beispiel

Standardfehler Beispiel: Logging in Diagnosedaten

```
(Date/Time): 07.09.2012 / 11:37:38
Version: V3.00.3012.04   Modul: DECU_TRF.C   Cycle: 3108
-----
ERRTXT: Backwardtransformation after forwardtransformation results in
different position.

-----
Fehler-ID   : 292023      BF-Typ : 9                Kanal-ID   : 1
Mehrfach-ID : 1         Line   : 2213             Kommu-ID  : 42
Behebungs-  : 2         Reaktions-  : 2          Rumpftypep: 1
NC-File     : log. Pfadnr. 65535 -> D:\TwinCAT3\test.nc
NC-Programm: trafo_test
NC-Proginfo:
Satznummer : 20         Fileoffset: 55           Satzoffset: 14
----- NC_Satz -----
Ausgabe nicht moeglich! log_pfad_nr nicht in Zuordnungstabelle.
Wert_1: Aktueller Wert ist 500 [-]
Wert_2: Fehlerhafter Wert ist 1005 [-]
----- Ende der Fehlermeldung -----
```

Anwenderspezifische Transformationsfehler

Neben den standardmäßigen Transformationsfehler kann der Anwender bei einigen Methoden (Bsp. Fehler ID 123) individuelle Fehler über den Rückgabewert der Funktion absetzen (0 = OK).

```
HRESULT CTrafo::Forward(PTcNcTrafoParameter p)
{
    if (...)
        return 123; // raise error
    ...
    return S_OK;
}
```

Fehlertexte in TcCncUsersEvents.xml

Im Fehlerfall wird der individuelle Rückgabewert der Methode an die Fehlermeldungsauswertung über die SPS oder den TwinCAT-Eventlogger weitergereicht (siehe auch FCT-M7// Fehlerausgabe TwinCAT3). Die Fehlertexte werden entsprechend in den XML-Fehlertextdatei pro Sprache ergänzt (C:\TwinCAT\3.1\Target\Resource):

```
<Event>
  <Id>123</Id>
  <Message LcId="1033">Kinematic transformation reports error 123</Mes-
  sage>
  <Message LcId="1031">Kinematische Transformation meldet Fehler 123</
  Message>
</Event>
```

Die Ausgabe des Fehlers erfolgt über den Eventlogger.

Direkte Ausgabe von anwenderspezifischen Transformationsfehlermeldungen

Ab CNC-Version V3.1.3081.4 bzw. V3.1.3110 können anwenderspezifische Fehlermeldungen der eigenen Transformation direkt ausgegeben werden. Hierfür ist der Fehlerbereich ID 500000 bis ID 500999 vorgesehen.

Das Vorgehen ist wie folgt:

- Fehlerwert aus Transformation im vorgesehenen Bereich zurückgeben
- Integration des entsprechenden Fehlertextes in TcCncUsersEvents.xml-Datei, siehe obiges Codebeispiel.

Erweiterte Fehlerrückgabewerte

Falls die erweiterten Transformationsparameter **TcNcTrafoParameterExtCnc**. benutzt wird, so können Fehlerfall zusätzliche Fehlerwerte zurückgegeben werden. Diese Werte werden in der Fehlermeldung angezeigt.

```
double    ret_value1; // out: error value
double    ret_value2; // out: error value
char      ret_text[24]; // out: additional error info
```

**Beispiel****Anwenderspezifischer Fehler**

```
<<-----20.06.2013 16:31:06:019 (11862) Version:
V3.00.3017.00
-----
Error : 292033 - Fehler bei kinematischer Rueckwaertstransformation
-----
Program : trafo_test
Path : D:\TwinCAT3\ (No: 65535)
File : _trafo-error-test.nc
Block no: N60 Fileoffset: 151
Line : N060 Y42 ; util_error_Id = -12
-----
Channel : (No.: 1)
Value : 500
Class : ERROR (5) Reaction : PROGRAM_ABORT (2)
=====
Value 1 : Actual value : 500
Value 2 : Actual value : 0
Value 3 : Actual value :
-----
Utility : Error 123 - ...
Modul : Line : 0
-----
Config : EIN_KANAL_KONFIGURIERUNG
Modul : BAVO_5AX.C Line : 6438
BF-Type : BAVO (5) Commu: BAVO_1 (44) Multiple-ID: 0
Content : NC_PROGRAM (1)
-----

<<-----20.06.2013 16:31:06:019 (11862) Version: V3.00.3017.00
-----
Error : 292036 - Aktuelle WKS-Ausgangsposition der kinematischen Vor-
waertstransformation.
-----
Program : trafo_test
Path : D:\TwinCAT3\ (No: 65535)
File : _trafo-error-test.nc
Block no: N60 Fileoffset: 151
Line : N060 Y42 ; util_error_Id = -12
-----
Channel : (No.: 1)
Value : 000 [mm]
Class : WARNING (0) Reaction : PROGRAM_ABORT (2)
=====
Value 1 : Actual value : 0 / 1.05E+005 / 0 [0.1*10^-3 mm resp.
]
Value 2 : Actual value : 0 / 0 / 0 [0.1*10^-3 mm resp. ]
Value 3 : Actual value : 0 / 0 / 0 [0.1*10^-3 mm resp. ]
Value 4 : Actual value : 0 / 0 / 0 [0.1*10^-3 mm resp. ]
Value 5 : Actual value : 0 / 0 / 0 [0.1*10^-3 mm resp. ]
-----
Config : EIN_KANAL_KONFIGURIERUNG
Modul : BAVO_5AX.C Line : 6438
BF-Type : BAVO (5) Commu: BAVO_1 (44) Multiple-ID: 2
Content : NC_PROGRAM (1)
```

5.2 Diagnosedaten

Protokoll der Achspositionen

Die <n> zuletzt verwendeten Ein-/Ausgangspositionen der kinematischen Transformation werden aufgezeichnet. Beim Anfordern der Diagnosedaten (s. dump.bat) werden diese Werte in den Diagnosedaten diag_data.txt protokolliert. Folgende Transformationen werden in den Diagnosedaten erfasst:

- Positionen der Vorwärtstransformation des Dekoders
- Positionen der Rückwärtstransformation während Interpolation



Beispiel

Logging in Diagnosedaten

```
DECODER : KINEMATIC FORWARD-TRAFO, CHANNEL-NO.: 1
```

```
=====
TIME  ID0  ID1    IN[00]      IN[01]      ...    OUT[00]      ...
153441 500   0 -12600000.000 -12600000.000 ... -40365738.845 ...
153448 500   0 -12600000.000 -12600000.000 ... -40365738.845 ...
153455 500   0 -12600000.000 -12600000.000 ... -40365738.845 ...
153508 500   0 -12683073.658 -12663380.896 ... -40375276.063 ...
```

```
BAHN : KINEMATIC BACKWARD-TRAFO, CYCLIC, CHANNEL-NO.: 1
```

```
=====
TIME  ID0  ID1    IN[00]      IN[01]      ...    OUT[00]      ...
215242 500   0 -40243827.115 11630715.707 ... -14546527.976 ...
215244 500   0 -40243827.115 11630715.706 ... -14546527.977 ...
215243 500   0 -40243827.115 11630715.706 ... -14546527.976 ...
215245 500   0 -40243827.115 11630715.705 ... -14546527.977 ...
215246 500   0 -40243827.115 11630715.705 ... -14546527.977 ...
215247 500   0 -40243827.115 11630715.704 ... -14546527.977 ...
```

```
BAHN : KINEMATIC BACKWARD-TRAFO, DEST, CHANNEL-NO.: 1
```

```
=====
TIME  ID0  ID1    IN[00]      IN[01]      ...    OUT[00]      ...
199502 500   0 -40245718.482 11643365.168 ... -12937949.950 ...
199503 500   0 -40245466.676 11641681.090 ... -12938699.356 ...
199504 500   0 -40245236.484 11640141.566 ... -12939384.586 ...
199505 500   0 -40245027.043 11638740.826 ... -12940008.169 ...
199506 500   0 -40244837.558 11637473.552 ... -12940572.438 ...
```

6 Verkettung von Transformationen, Multistep Transformationen

Mehrstufigkeit - Additive kinematische Transformation

Normalerweise wird nur eine kinematische Transformation genutzt, aber die CNC bietet die Möglichkeit, mehrere kinematische Teiltransformationen zu kaskadieren. Momentan kann eine zusätzliche Transformation zur normalen Transformation verkettet werden.

Durch diese Möglichkeit kann der Anwender seine Transformationen unabhängig strukturieren:

- Standard kinematische Trafo (Stufe=0): bildet die kinematische Basiskette der Maschine ab (Konfigurationstyp = TCCNC_REGISTEROBJECT_TYPE_TRAFO)
- Additive kinematische Trafo (Stufe=1): kompensiert z.B. dynamische Effekte der Maschine (Konfigurationstyp = TCCNC TCCNC_REGISTEROBJECT_TYPE_TRAFO_ADD)

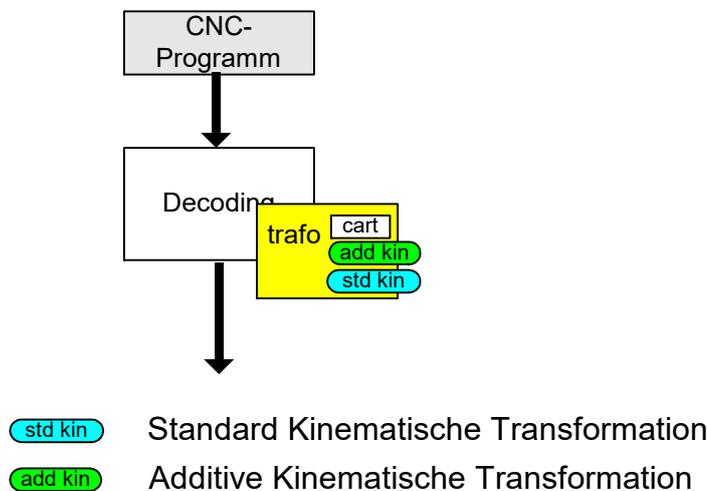


Abb. 15: Verkettung von kinematischen Transformationen

Initialisierung der Kinematikparameter

Die kinematischen Parameter für jede Stufe der kinematischen Transformation können in folgender Form in der Kanalliste initialisiert werden.

```
kin_step[0].id[83].param[0]      10000
kin_step[1].id[51].param[0]      55000
kin_step[1].id[51].param[1]      80000
```

Initialisierung der Standardtransformation

Die Standardtransformation jeder Stufe kann in der Kanalliste in folgender Form definiert werden.

```
default_id_of_kin_step[0]        83
default_id_of_kin_step[1]        51
```

Parameter Zugriff im NC Programm

Die kinematischen Parameter jeder Stufe können im NC-Programm folgendermaßen adressiert werden.

```
N10 V.G.KIN_STEP[1].ID[1].PARAM[0] = 55000
N20 V.G.KIN_STEP[1].ID[1].PARAM[1] = 80000
```

Aktivierung einer Transformation für jede Stufe

Die einzelnen Kinematikstufen können über folgende NC-Befehle angewählt werden:

```
#TRAFO [<kin-id-step0>, <kin-id-step1>]
#TRAFO [DEFAULT, DEFAULT]
      ;DEFAULT = Wert von Parameter default_id_of_kin_step[]

#TRAFO [ OFF, <kin-id-step1>]
#TRAFO [<kin-id-step0>, OFF]

#TRAFO [ OFF, OFF]
#TRAFO OFF
```

7 Erstellen einer Transformation

Bei der Erstellung eines TcCOM-Objekts mit dem TwinCAT3-Template wird standardmäßig eine sogenannte erweiterte Transformation angelegt.

7.1 Erstellungsablauf

Mindestanforderung für die Nutzung der McCOM-Assistenten

- TwinCAT3 Version 4024
- Microsoft Visual Studio 2019 Professional/ Enterprise, bei der Installation muss die Option "Desktop development with C++" zusätzlich selektiert sein.

Die Transformation wird mit Hilfe eines TwinCAT3-Templates erstellt

Folgende Schritte sind vom Anwender durchzuführen:

- Anlegen oder Öffnen eines TwinCAT3 XAE-Projekts mit integrierter CNC-Konfiguration
- Erstellen des Rahmens für Transformation über Template wie exemplarisch gezeigt wird.
- Eigenen C++ Code für Transformation erstellen (dieser Schritt kann auch später erfolgen, es muss allerdings dann auch der Treiber erneut erstellt werden)
- Treiber (MyTrafo.sys) erzeugen
- Einbinden der Transformation als TcCOM-Objekt im XAE-Projekt Konfiguration
- Aktivieren der Konfiguration



Hinweis

Der Treiber MyTrafo.sys wird beim Aktivieren der Konfiguration automatisch in das Verzeichnis <TwinCAT>\3.1\Driver\Autoinstall kopiert.
In diesem Verzeichnis werden alle additiven Treiber platziert.

7.1.1 Neues Projekt anlegen

Der nachfolgende exemplarische Ablauf zur Erstellung einer eigenen kinematischen Transformation mittels TcCOM-Objekt wurde mit Visual Studio 2019 durchgeführt.

TwinCAT3 XAE Projekt mit CNC-Konfiguration

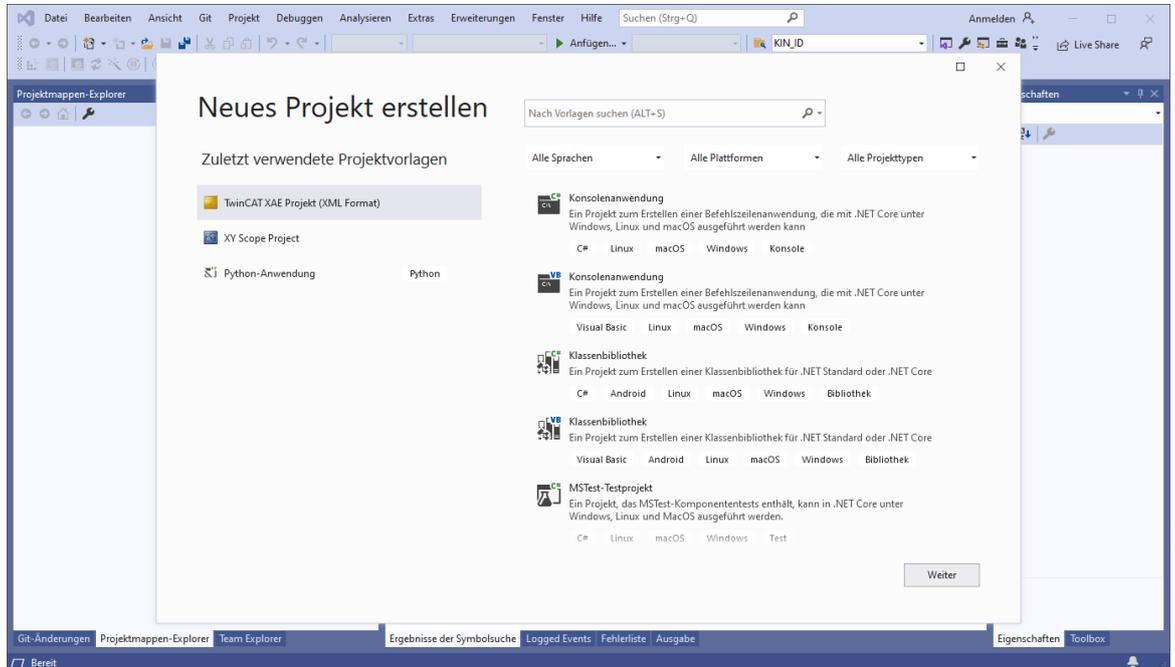


Abb. 16: Erstellung eines neuen Projekts

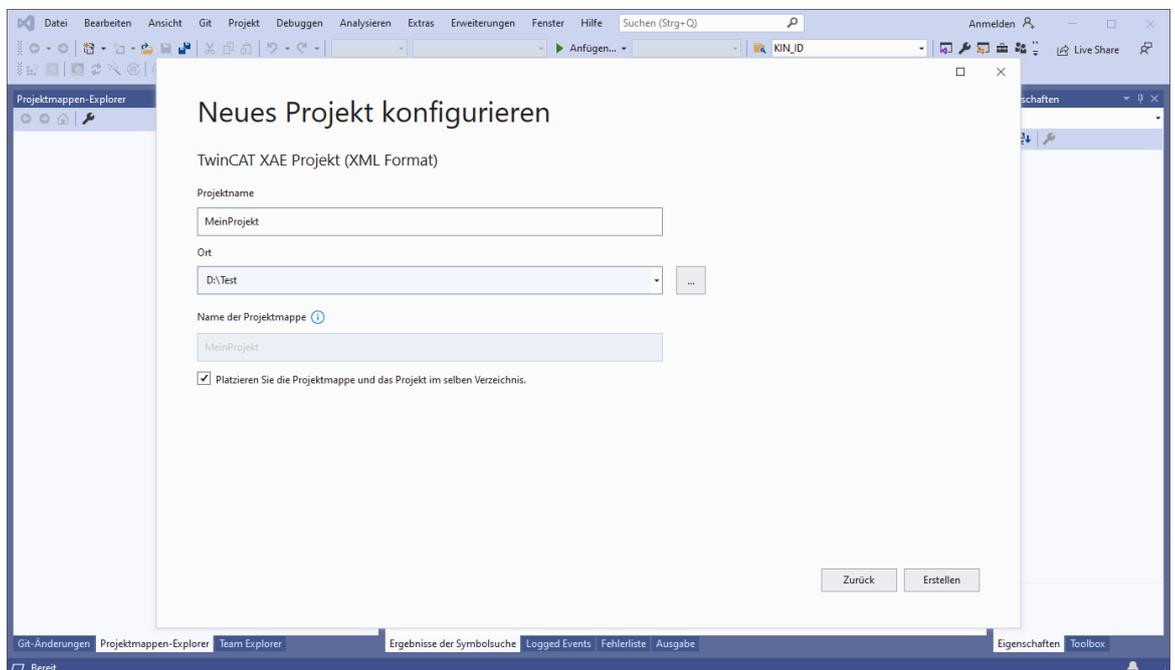


Abb. 17: Konfiguration des neuen Projekts

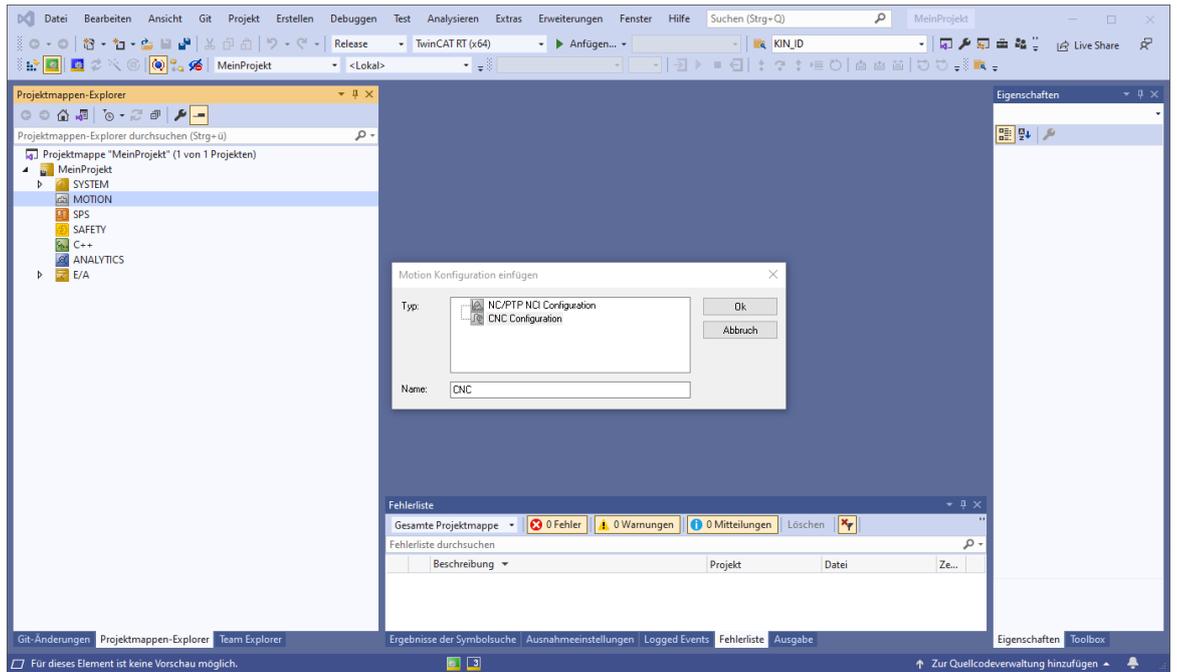


Abb. 18: Anlegen einer CNC-Konfiguration

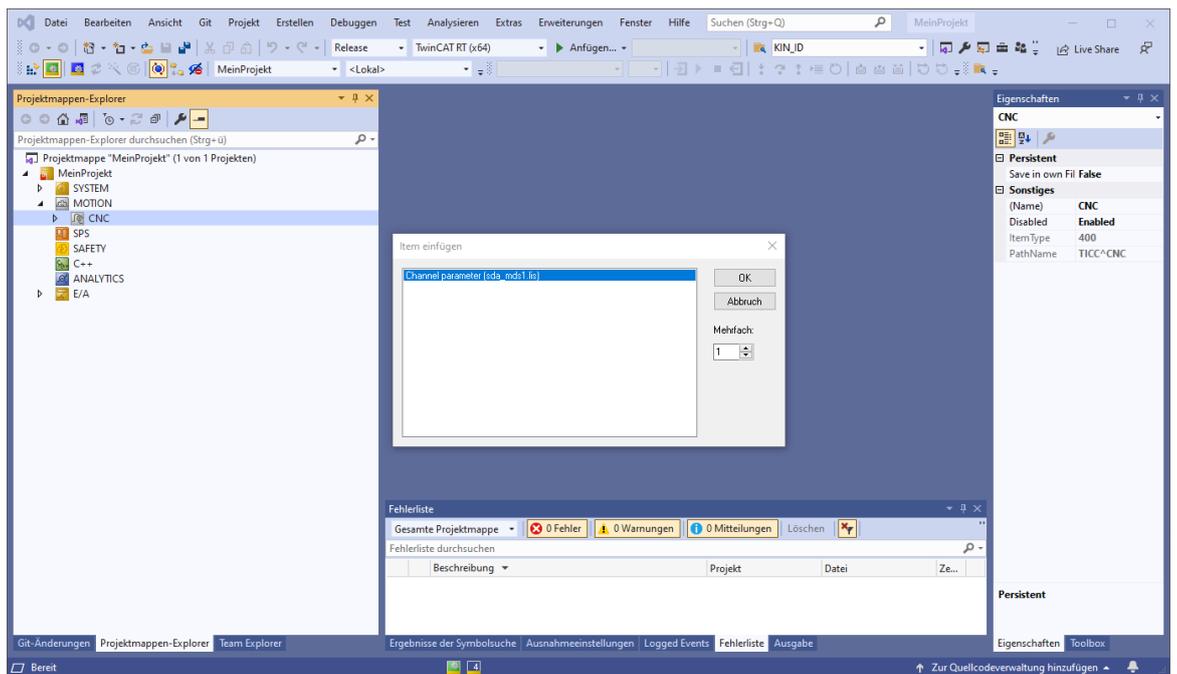


Abb. 19: Anlegen eines Kanals

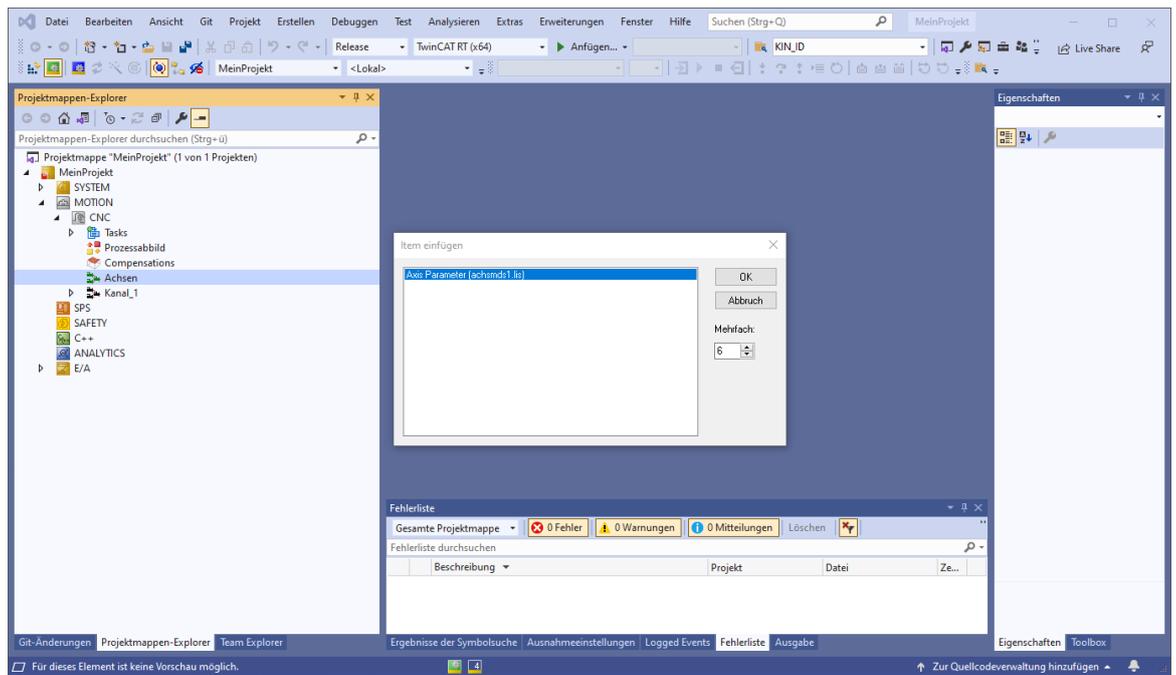


Abb. 20: Anlegen einer Achse

7.1.2 Transformation erstellen

Über TwinCAT3-Template für eigene Transformation anlegen.

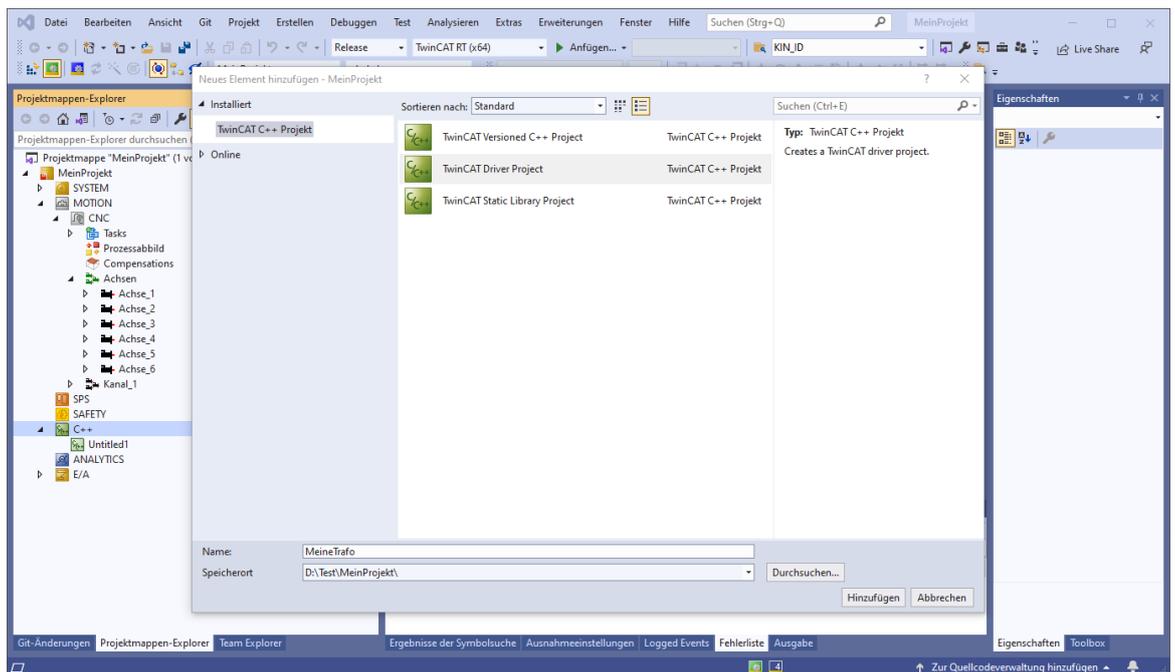


Abb. 21: TwinCAT Treiber-Projekt anlegen

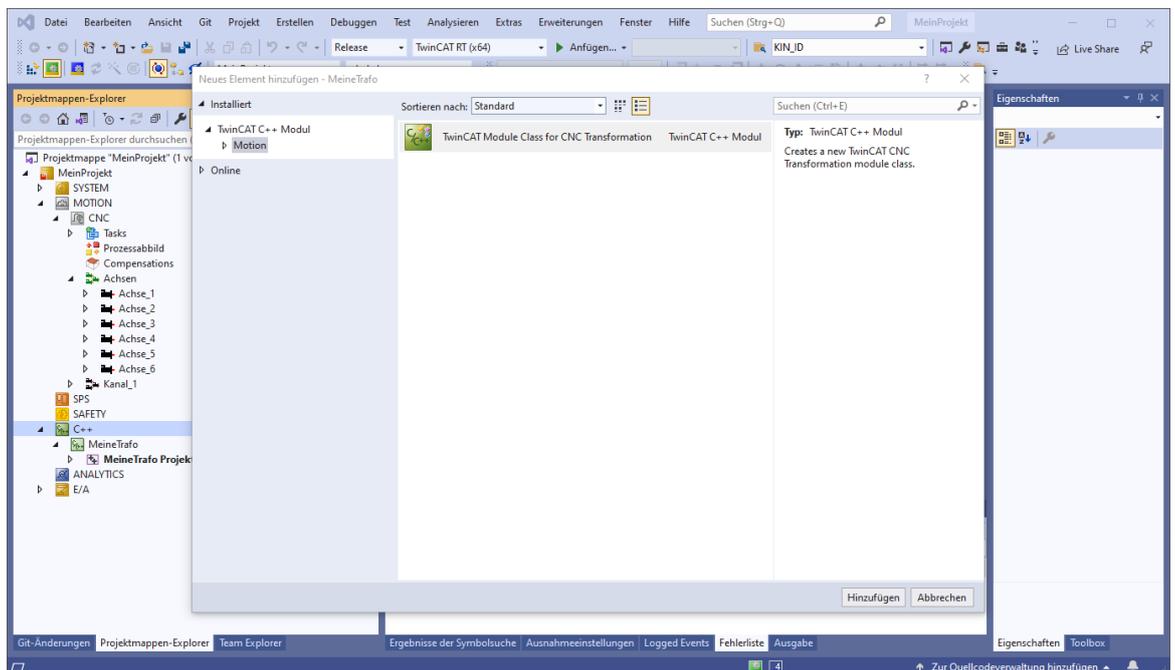


Abb. 22: Transformations-Klasse anlegen

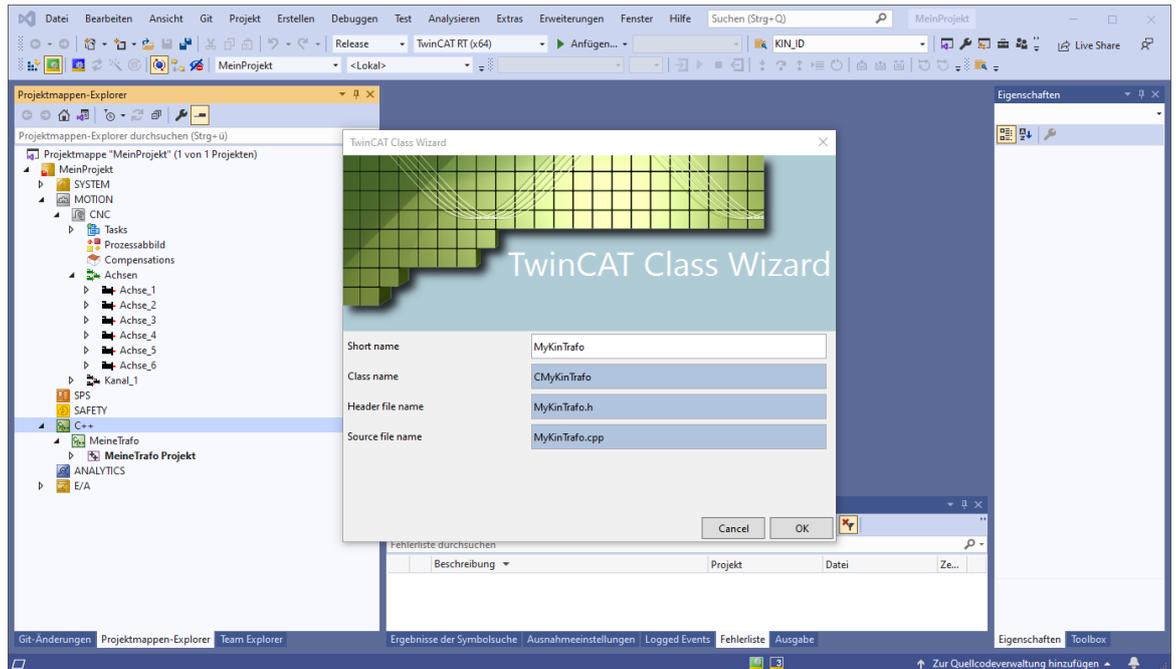


Abb. 23: Benennung Transformations-Klasse

Damit ist der Rahmen für das TcCOM-Objekt im Visual Studio festgelegt.

Erstellen des Treibers

Über einen Rechtsklick auf das Projekt kann mit „Erstellen“ der Treiber erstellt werden.

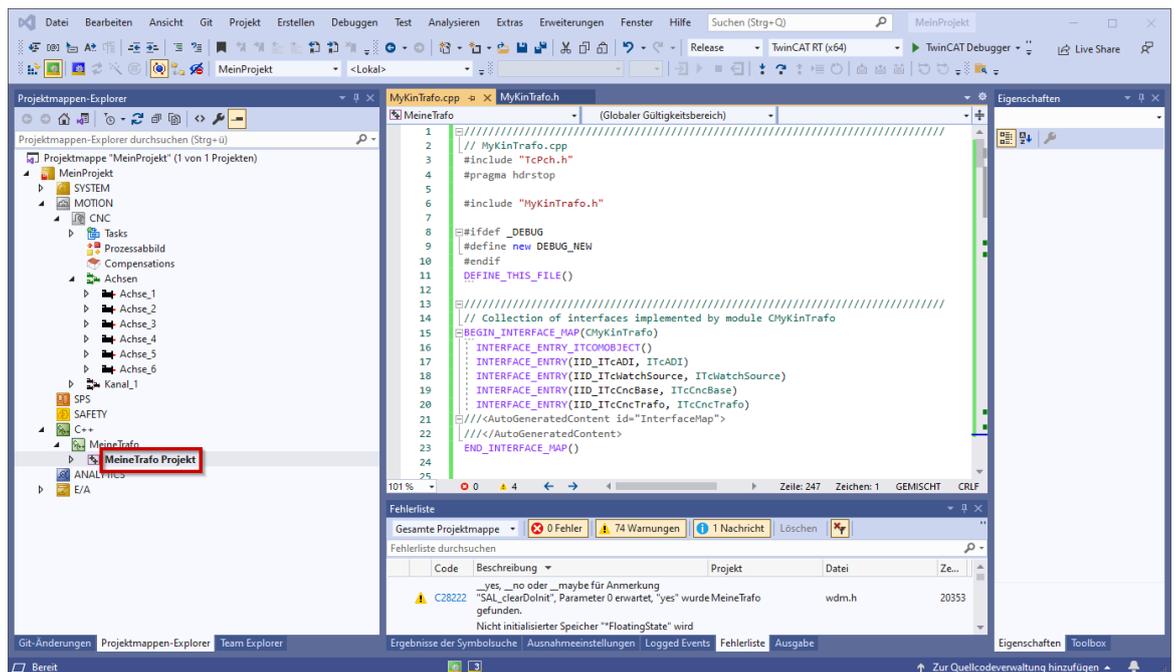


Abb. 24: Treiber erstellen

7.1.3

Transformation einbinden

Die Transformation wird in die vorhandene CNC-Konfiguration wie folgt integriert:

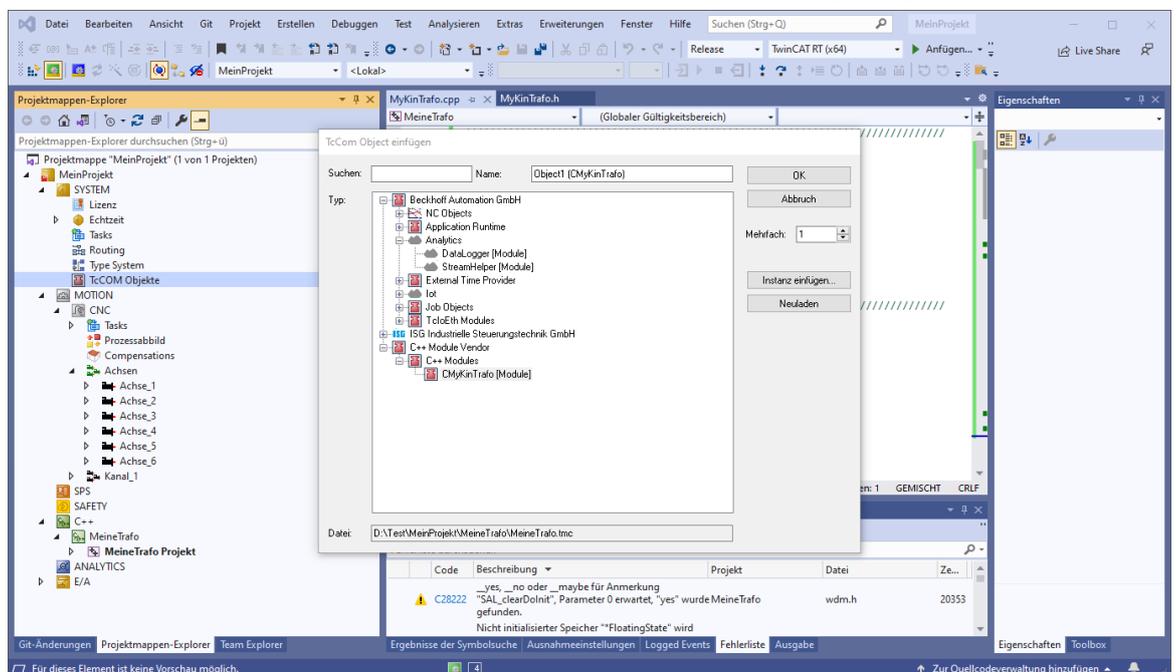


Abb. 25: Einbinden TcCOM-Objekt

Mit Bestätigen des „OK“-Buttons ist die Integration abgeschlossen.

Durch Doppelklicken auf das TcCOM-Objekt werden die Eigenschaften angezeigt:

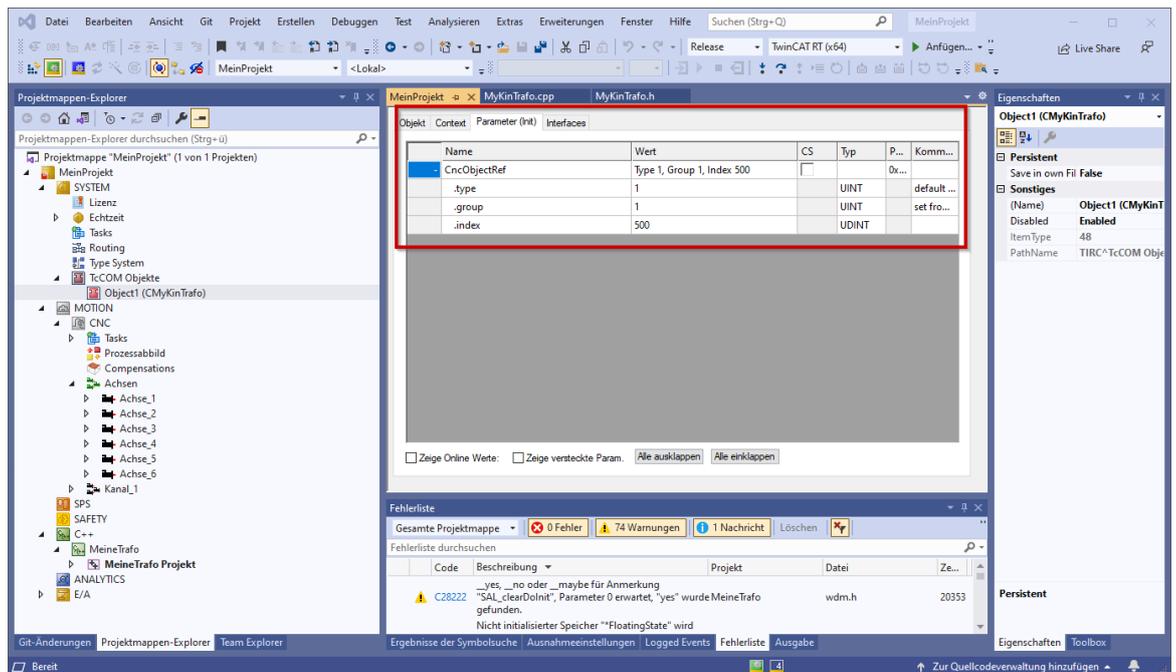


Abb. 26: Eigenschaften TcCOM-Objekts

Parameter	Zulässige Werte	Beschreibung
Type	1	Type = 1 gibt an, dass es sich bei dem TcCOM-Objekt um eine kinematische Transformation handelt.
Group	$0 \leq \text{group} \leq \text{Kanalanzahl}$	Im Parameter group wird angegeben, welcher CNC-Kanal auf die Transformation zugreifen darf. Bei group = 0 ist die Transformation für alle CNC-Kanäle verfügbar.
Index	$500 \leq \text{index} \leq 999$ Aus Kompatibilitätsgründen: $65 \leq \text{index} \leq 69$	Über den Parameterindex erhält die Transformation eine im CNC-Kanal eindeutige ID, über die sie in der CNC angesprochen werden kann wie z.B. im NC-Programm mit dem Befehl #KIN ID [500].

Parametrieren der Transformation in der CNC

Die angelegte Kinematik muss in der CNC noch parametrieren werden. Dies kann in der Standardkanalparameterliste oder in der jeweiligen Kanalparameterliste erfolgen.

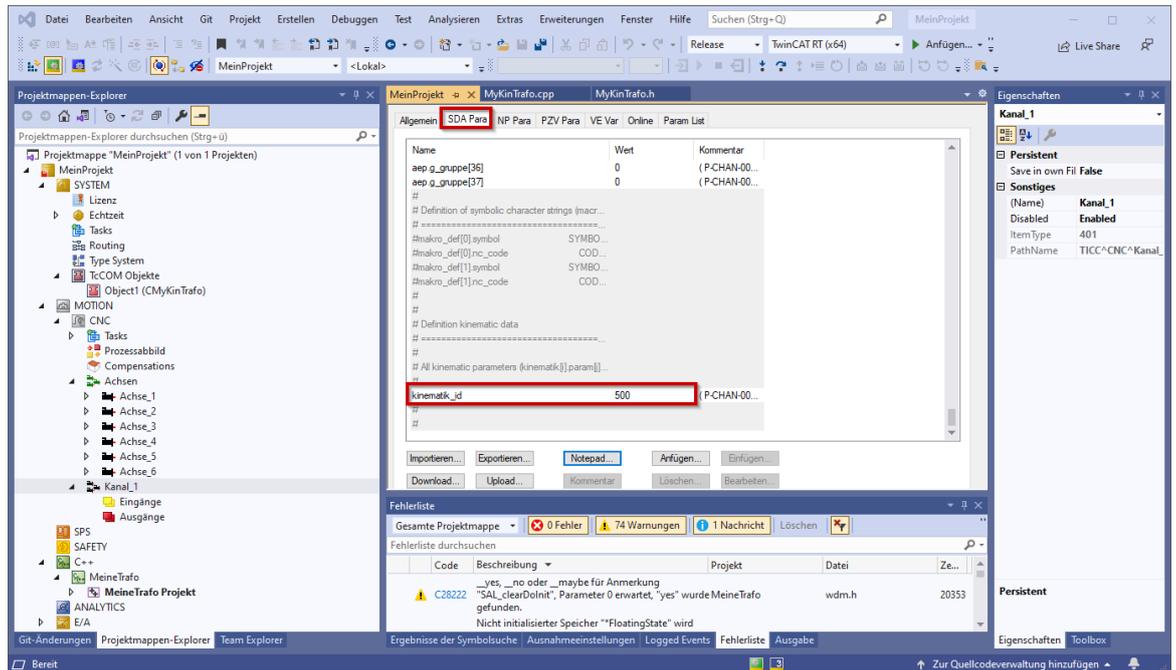


Abb. 27: Parametrieren der Transformation in Kanalparameterliste

Die einzutragende Transformations-ID ist der Index des TcCOM-Objekts.

7.1.4

Transformation debuggen

Zum Debuggen der Transformation im TwinCAT3 Projekt auf Debug umgestellt werden und der Echtzeit-Debugger muss aktiviert werden.



Abb. 28: Umstellen auf Debug-Konfiguration

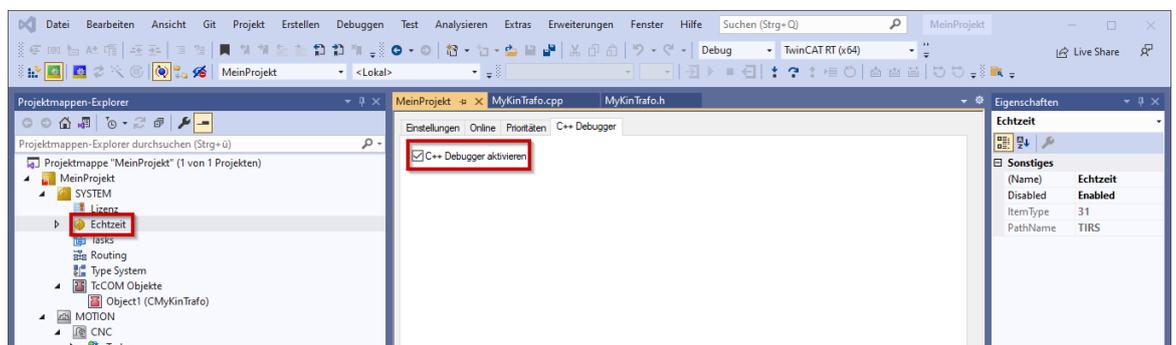


Abb. 29: Aktivieren Echtzeit-Debugging



Hinweis

Der Debug-Treiber MyTrafo.sys und die entsprechende pdb-Datei werden automatisch mit der Aktivierung der Konfiguration in das Autoinstall-Verzeichnis kopiert.

Nach dem Start von TwinCAT in den Zustand „RUN“ kann das Debuggen des Trafo-Projekts gestartet werden und die entsprechenden Breakpoints gesetzt werden.

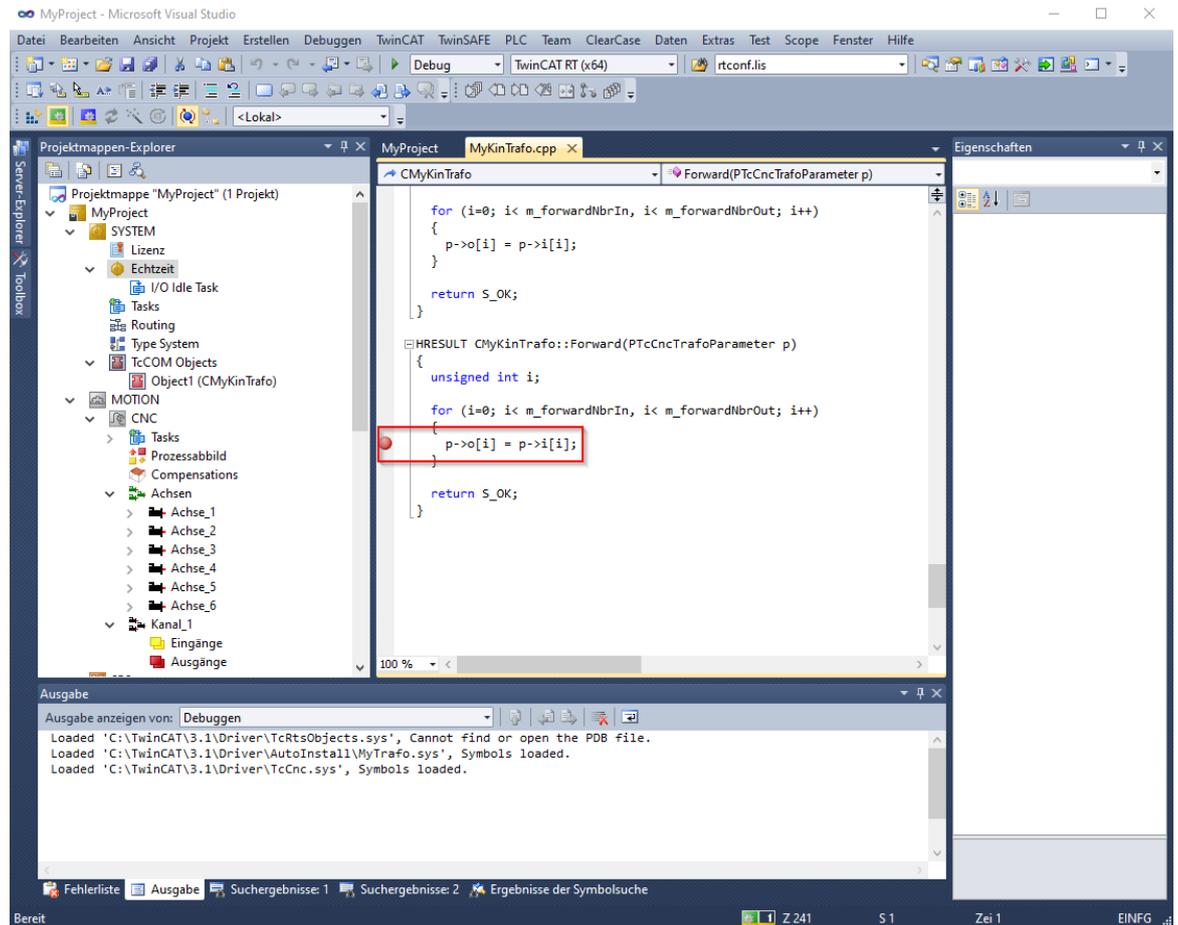


Abb. 30: Breakpoint in Transformation

7.1.5 Quellcodeerweiterung / Codierung

Für die Erstellung müssen die eigenen Transformationsgleichungen in den Funktionen

- Forward
- Backward
- TrafoSupported
- GetDimensions

integriert werden. Diese sind exemplarisch in der Datei MyKinTrafo.cpp bereits über das TwinCAT3-Template erstellt worden.

Anwendertipp

Wenn die Transformation mehr als 5 Achsen benötigt muss der Konstruktor wie folgt angepasst werden. Bei weniger als 5 Achsen müssen die Werte entsprechend reduziert werden.

```
////////////////////////////////////  
// Constructor  
CMyKinTrafo::CMyKinTrafo(): m_forwardNbrIn(5), m_forwardNbrOut(5)  
{
```

Abb. 31: Einstellung des Konstruktor nach Generieren mit TwinCAT3-Template

```
////////////////////////////////////  
// Constructor  
CMyKinTrafo::CMyKinTrafo(): m_forwardNbrIn(7), m_forwardNbrOut(7)  
{  
    ///<AutoGeneratedContent id="MemberInitialization">
```

Abb. 32: Angepasster Konstruktor wegen höherer Achszahl

Wird der Wert im Konstruktor höher eingegeben als Achsen im Kanal sind dann wird die Fehlermeldung 20658 ausgegeben. Diese Fehlermeldung wird ebenfalls ausgegeben wenn die Konfiguration der Achsen im Kanal lückend ist.

Lösungsmöglichkeiten:

- Prüfen und korrigieren der Lücke in der Konfiguration
- Anpassen des Konstruktors an Achszahl im verwendeten Kanal

Nach Implementierung der Funktionen muss der Treiber neu erstellt werden und die Konfiguration erneut aktiviert werden.

7.2 Unterschiede erweiterte Transformation / Standardtransformation

Bei Verwendung des TwinCAT3-Templates wird standardmäßig die erweiterte Transformation angelegt. Sie ist ausschließlich für die Nutzung mit der CNC ausgelegt.

Die Standard Transformation wird dann verwendet wenn sowohl die CNC als auch die NCI zum Einsatz kommen.

(NCI –Steuerungslösung von Beckhoff)

Erweiterte Transformation	Standard Transformation
Erweitertes Interface: ITcCncTrafo GUID erweitertes Interface: IID_ITcCncTrafo	Standardinterface: ITcNcTrafo GUID Standardinterface: IID_ITcNcTrafo
Erweiterte Transformationsparameter: PTcCncTrafoParameter	Standardparameter: PTcNcTrafoParameter

8 Zusätzliche Optionen der erweiterten Transformation

8.1 Versionskennung Transformationsinterface

Das Transformationsinterface besitzt eine eindeutige Versionskennung (<Major>.<Minor>). Die Versionsnummer der CNC wird der TcCOM-Transformation im Datum p->CncInterfaceVersion bereitgestellt. Die eigene Versionsnummer kann das TcCOM-Objekt über die GetInterfaceVersion() Methode erfragen. Das Transformationsinterface der CNC ist abwärtskompatibel d.h. TcCOM-Objekte mit einer älteren Interface-Version können mit neueren CNC-Versionen weiterhin eingesetzt werden. Umgekehrt gilt dies jedoch nicht: Die Interface-Version der CNC muss mindestens so aktuell wie das Transformationsinterface des TcCOM-Objekts sein, ansonsten generiert die CNC die Fehlermeldung ID 292044.



Beispiel

```
HRESULT <UserTrafo>::TrafoSupported(PTcCncTrafoParameter p, bool fwd)
```

```
{
    ...
    TcCncVersion TcCOMInterfaceVersion;
    this->GetInterfaceVersion(&TcCOMInterfaceVersion);

    if ( (TcCOMInterfaceVersion.major <= p->CncInterfaceVersion.major)
        && (TcCOMInterfaceVersion.minor <= p->CncInterfaceVersion.minor) )
    {
        return S_OK;
    };
}
```

8.2 Drehreihenfolge

Bei vollständigen Transformationen kann die Reihenfolge der ausgeführten Rotationen um die 3 Drehachsen beliebig festgelegt werden (s. P-CHAN-00112). Falls dies gewünscht ist, muss dies auch die TcCOM-Transformation berücksichtigen. Die aktuelle Einstellung wird der Transformation daher im Parameter p->actual_orientation_mode übergeben. Die in der Transformation unterstützten Drehsequenzen können der CNC im Datum p->supported_orientation_modes bei Aufruf der Funktion TrafoSupported() mitgeteilt werden. Bei angewählter Transformation prüft die CNC die Einstellung in P-CHAN-00112 auf Plausibilität und generiert die Fehlermeldung ID 292045 falls die gewählte Rotationsreihenfolge von der Transformation nicht unterstützt wird.

CNC --> TcCOM-Transformation:

p->actual_orientation_mode	Bedeutung
EcCncTrafoOri_None	Keine Rotation
EcCncTrafoOri_YPR	Yaw-Pitch-Roll Rotationsreihenfolge: 1. Drehung um Z, 2. Drehung negativ um Y, 3. Drehung um X
EcCncTrafoOri_CBC1	Euler Rotationsreihenfolge: 1. Drehung um Z, 2. Drehung um Y, 3. Drehung um Z'
EcCncTrafoOri_CBA	1. Drehung im Z, 2. Drehung um Y, 3. Drehung um X
EcCncTrafoOri_CAB	1. Drehung um Z, 2. Drehung um X, 3. Drehung um Y
EcCncTrafoOri_AB	1. Drehung um X, 2. Drehung um Y
EcCncTrafoOri_BA	1. Drehung um Y, 2. Drehung um X
EcCncTrafoOri_CA	1. Drehung um Z, 2. Drehung um X
EcCncTrafoOri_CB	1. Drehung um Z, 2. Drehung um Y

TcCOM-Transformation --> CNC:

p->supported_orientation_modes	Bedeutung
.f_YPR	= TRUE, Trafo unterstützt Drehreihenfolge YPR
.f_CBC1	= TRUE, Trafo unterstützt Drehreihenfolge CBC'
.f_CBA	= TRUE, Trafo unterstützt Drehreihenfolge CBA
.f_CAB	= TRUE, Trafo unterstützt Drehreihenfolge CAB
.f_AB	= TRUE, Trafo unterstützt Drehreihenfolge AB
.f_BA	= TRUE, Trafo unterstützt Drehreihenfolge BA
.f_CA	= TRUE, Trafo unterstützt Drehreihenfolge CA
.f_CB	= TRUE, Trafo unterstützt Drehreihenfolge CB

Standardmäßig verwendet die CNC die Einstellung EcCncTrafoOri_YPR (Yaw->Pitch->Roll), entsprechend ist das Datum p->supported_orientation_mode.f_YPR standardmäßig auf den Wert TRUE gesetzt.


Beispiel

```

HRESULT <UserTrafo>::TrafoSupported(PTcCncTrafoParameter p, bool fwd)
{
    ...
    /* Transformation unterstuetzt YPR und Euler Drehreihenfolge. */
    p->supported_orientation_modes.f_YPR = TRUE;
    p->supported_orientaiton_modes.f_CBC1 = TRUE;
    ...
  
```

```

    return S_OK;
}

HRESULT <UserTrafo>::Backward(PTcCncTrafoParameter p)
{
    ...
    if (EcCncTrafoOri_CBC1 == p->actual_orientation_mode)
    {
/* Drehreihenfolge nach Euler aktiv */
    }
    else
    {
        ...
    }
    return S_OK;
}

```

8.3 Modulobehandlung der Achspositionen

Normalerweise werden die Positionen im MCS Koordinatensystem von der CNC linear behandelt d.h. es findet keine Modulkorrektur statt. Falls die Transformation die MCS Positionen im Modulintervall $[-180^\circ - +180^\circ[$ erwartet (z.B. für Shortest-Way Programmierung), kann dies in der Funktion TrafoSupported() über das Datum mcs_modulo eine Modulkorrektur für eine Achse im MCS-Koordinatensystem aktiviert werden.

p->mcs_modulo[i]	Bedeutung
EcCnc_McsModulo_None	Lineare MCS-Positionen, keine Modulorechnung für diese Achse
EcCnc_McsModulo_180_180	Modulorechnung der MCS-Positionen für diese Achse im Intervall $[-180^\circ, +180^\circ[$.

Die berechneten ACS-Koordinaten müssen mit den Eigenschaften der Achsen übereinstimmen. Falls die Achse Modulopositionen verwendet müssen auch die ACS-Koordinaten in der Transformation eine Modulkorrektur durchgeführt werden. Im achsspezifischen Datum acs_modulo kann daher die in der Transformation verwendete Moduloeinstellung der CNC mitgeteilt werden. Die CNC prüft dann ob die Transformation zu den Achseigenschaften passt und generiert gegebenenfalls die Fehlermeldung P-ERR-50534.

p->acs_modulo[i]	Bedeutung
EcCnc_AcsModulo_None	Lineare ACS-Positionen, für diese Achse ist keine Modulbehandlung notwendig.
EcCnc_AcsModulo_180_180	Für diese Achse ist eine Modulorechnung der ACS-Positionen im Intervall $[-180^\circ, +180^\circ[$ notwendig.
EcCnc_AcsModulo_0_360	Für diese Achse ist eine Modulorechnung der ACS-Positionen im Intervall $[0^\circ, 360^\circ[$ notwendig.



Programmierbeispiel

Modulobehandlung der Achspositionen

```
HRESULT <UserTrafo>::TrafoSupported(PTcCncTrafoParameter p, bool fwd)
{
    ...
    /* 3 Achsen lineare MCS-Positionen,
       Modulobehandlung fuer die 4. Achse */
    p->mcs_modulo[0] = EcCnc_McsModulo_None
    p->mcs_modulo[1] = EcCnc_McsModulo_None
    p->mcs_modulo[2] = EcCnc_McsModulo_None
    p->mcs_modulo[3] = EcCnc_McsModulo_180_180

    /* 2 Achsen lineare ACS Positionen,
       Modulobehandlung fuer 2 Achsen */
    p->acs_modulo[0] = EcCnc_AcsModulo_None
    p->acs_modulo[1] = EcCnc_AcsModulo_180_180
    p->acs_modulo[2] = EcCnc_AcsModulo_0_360
    p->acs_modulo[3] = EcCnc_AcsModulo_None
}
```

8.4 Anwendung erweiterter Parameter

Nachfolgendes Beispiel demonstriert die Anwendung der erweiterten Transformationsparameter.

```

HRESULT <UserTrafo>::TrafoSupported(PTcNcTrafoParameterExtCnc p, bool fwd)
{
  if ( p == NULL )
  {
    return E_POINTER;
  }
  if ( p->type != EcNcTrafoParameter_ExtCnc )
  {
    p->ret_value1 = (double)p->type;
    strcpy(p->ret_text, "EcNcTrafoParameter");
    return S_FALSE;
  }
  if ( p->i == NULL || p->o == NULL )
  {
    return E_INVALIDARG;
  }
  if ( p->dim_i != m_forwardNbrIn )
  {
    p->ret_value1 = p->dim_i;
    p->ret_value2 = m_forwardNbrIn;
    strcpy(p->ret_text, "m_forwardNbrIn");
    return S_FALSE;
  }
  if ( p->dim_o != m_forwardNbrOut )
  {
    p->ret_value1 = p->dim_o;
    p->ret_value2 = m_forwardNbrOut;
    strcpy(p->ret_text, "m_forwardNbrOut");
    return S_FALSE;
  }

  /*
  +-----+
  | request addition input parameters ... if backward interpolation trafo |
  +-----+
  */
  if ((FALSE == fwd) && (p->caller_id == EcCncTrafoCallerID Interpolation))
  {
    ""
  }

  return S_OK;
}

```

8.5 Verwendung erweiterter Optionen

Ein-/Ausgangszahl

Normalerweise ist die Anzahl der Eingänge und der Ausgänge symmetrisch in Vorwärts- und Rückwärtsrichtung. Diese grundsätzliche Zahl wird über die Methode **GetDimension** definiert.

Für spezielle Anforderungen kann die Transformation zusätzliche Eingänge auswerten. Über die Methode **TrafoSupported** kann die Ein-Ausgangszahl entsprechend den Anforderungen angepasst werden.

- CNC-Option (`ret_option`)
- Anzahl der zusätzlichen Eingangswerte (`dim_i`)

In diesem Fall muss die CNC die zusätzlichen Werte an der Schnittstelle zur Verfügung stellen. Wird dies nicht durch die CNC unterstützt, so wird eine Fehlermeldung ausgegeben.

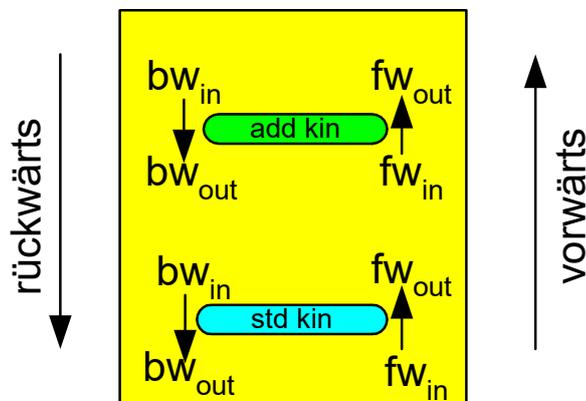


Abb. 33: Anpassung der Ein-/Ausgangszahlen

Die Transformationsoptionen können während der Konfigurationsabfrage (Methode `TrafoSupported`) gesetzt werden.

```

HRESULT <UserTrafo>::TrafoSupported(PTcNcTrafoParameterExtCnc p, bool fwd)
{
    /*
    +-----+
    | request addition input parameters ... if backward interpolation trafo |
    +-----+
    */
    if ((FALSE == fwd) && (p->caller_id == EcCncTrafoCallerID Interpolation))
    {
        p->ret_option = EcCncTrafoOption Interpolation AddInput;
        p->dim_i += 8;
    }

    return S_OK;
}
    
```

Forward/ backward Anpassung

Diese Anpassung kann individuell pro Vorwärts-/Rückwärtstransformation gemacht werden. Zusätzlich kann dies auch abhängig von den Aufrufstellen innerhalb der CNC durchgeführt werden (Decodierung, Werkzeugradiuskorrektur, usw.).

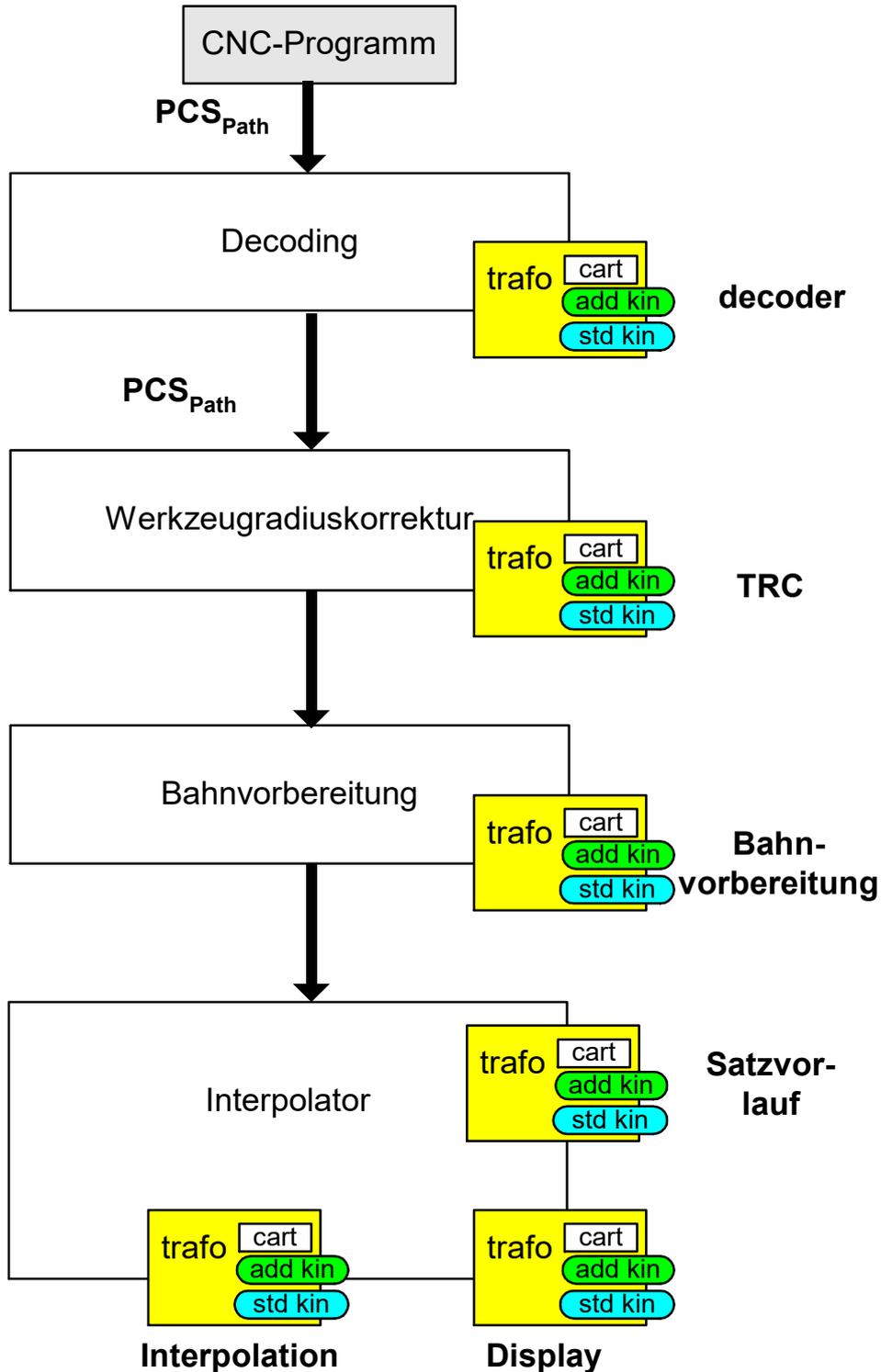


Abb. 34: Schnittstellen Anpassung an unterschiedlichen Aufrufstellen.

9 Anwenden und Nutzen der Caller-ID

Beispiel 1: Ressourcen einsparen in der Bahnvorberung (PathPreparation)

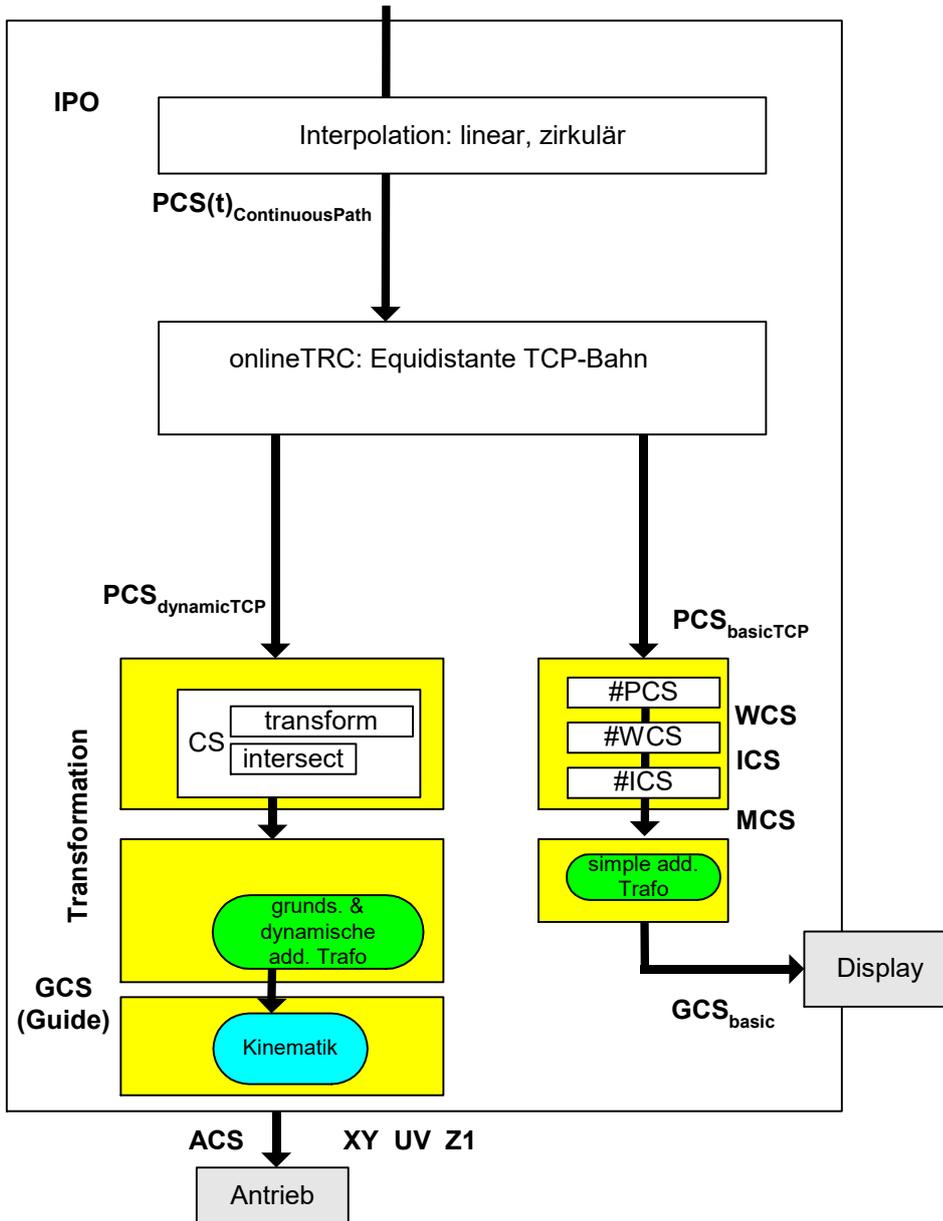
Eine Transformation kann rechen- und damit auch zeitintensiv sein.

In der PathPreparation kann es ablaufbedingt zu vielen Aufrufen der Rückwärtstransformation kommen, wodurch die interne Satzversorgung unter Umständen beeinträchtigt werden kann. Um dem entgegen zu wirken kann dann beispielsweise explizit für die PathPreparation eine Transformation mit geringerer Genauigkeit und damit weniger benötigter Ressourcen (Rechenzeit) genutzt werden. Die Verwendung einer Transformation mit geringerer Genauigkeit kann an ausgewählten Stellen, wie beispielsweise der PathPreparation, in Kauf genommen werden, wenn nur verhältnismäßig geringe Auswirkungen auf die Funktion der CNC auftreten bzw. zu erwarten sind. Dies ist daher eine denkbare Möglichkeit zur Lösung, eine Beeinträchtigung der Satzversorgung zu vermeiden.

Beispiel 2: Anzeige der Position der additiven Transformation

Während der Interpolation wird die additive Transformation für Anzeigezwecke aufgerufen (caller-ID = 5 = EcCncTrafoCallerID_Display). Diese Positionswerte können über CNC-Objekte pro Achse abgegriffen werden.

```
mc_ax_<i>_add_kin_pos_r
```



Reale Positionen Theoretische Positionen

Abb. 35: Anzeige der additiven Transformationsposition

Die additiven Transformationspositionen können auch über die Task COM mittels ISG Objektbrowser verifiziert werden.



Achtung

Die Vorwärtstransformation muss grundsätzlich invers zu Rückwärtstransformation sein.
 $position = forward(backward(position))$

Wird die Transformation in Abhängigkeit des Aufrufers (`caller_id`) variiert, so muss diese Variation bei der Initialisierung der Steuerung im Stillstand ausgeschaltet sein.

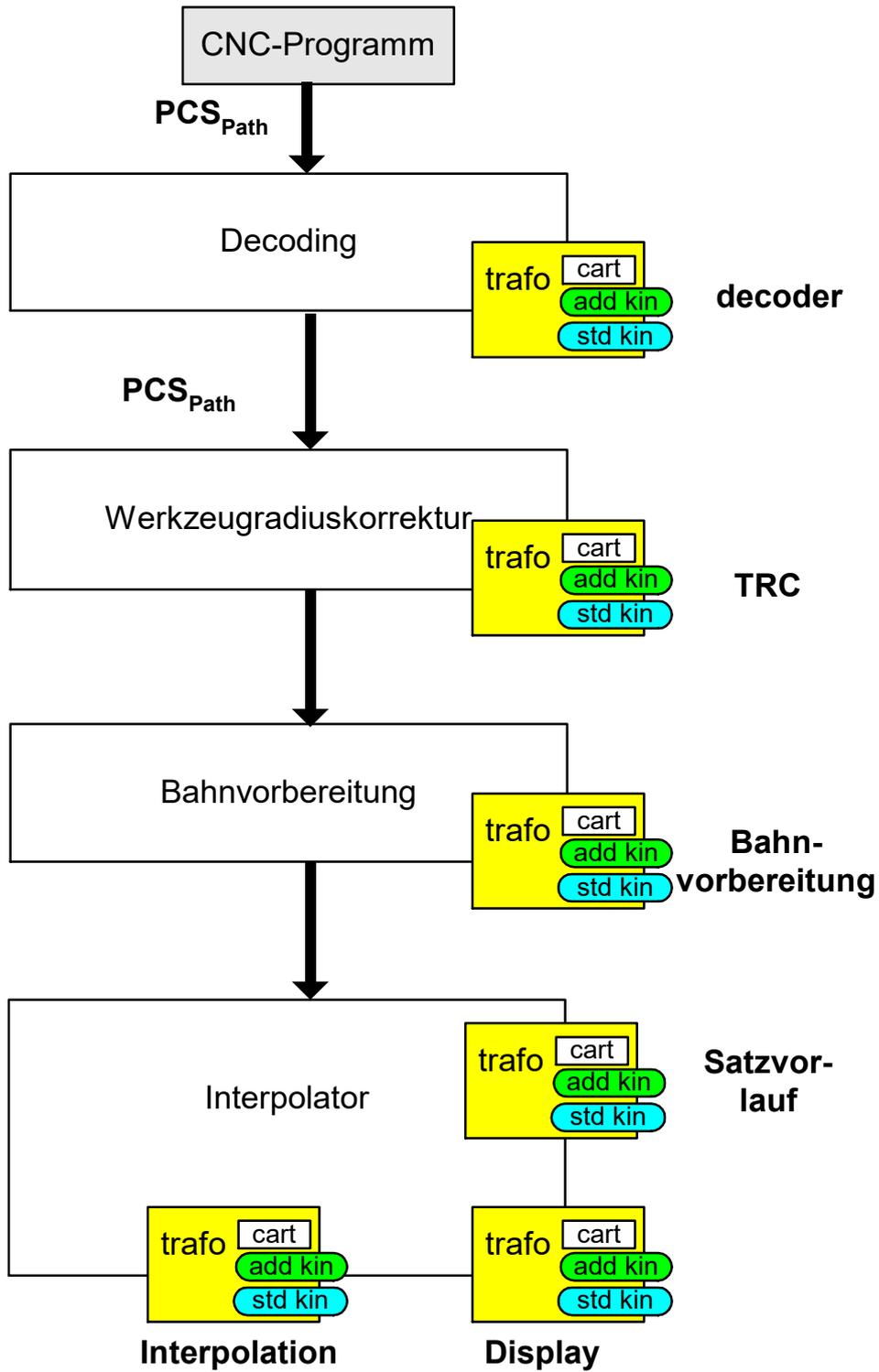


Abb. 36: Identifikation der Aufrufstellen der Transformation

10 Anhang

10.1 Anregungen, Korrekturen und neueste Dokumentation

Sie finden Fehler, haben Anregungen oder konstruktive Kritik? Gerne können Sie uns unter documentation@isg-stuttgart.de kontaktieren. Die aktuellste Dokumentation finden Sie in unserer Onlinehilfe (DE/EN):



QR-Code Link: <https://www.isg-stuttgart.de/documentation-kernel/>

Der o.g. Link ist eine Weiterleitung zu:

<https://www.isg-stuttgart.de/fileadmin/kernel/kernel-html/index.html>



Hinweis

Mögliche Änderung von Favoritenlinks im Browser:

Technische Änderungen der Webseitenstruktur betreffend der Ordnerpfade oder ein Wechsel des HTML-Frameworks und damit der Linkstruktur können nie ausgeschlossen werden.

Wir empfehlen, den o.g. „QR-Code Link“ als primären Favoritenlink zu speichern.

PDFs zum Download:

DE:

<https://www.isg-stuttgart.de/produkte/softwareprodukte/isg-kernel/dokumente-und-downloads>

EN:

<https://www.isg-stuttgart.de/en/products/softwareproducts/isg-kernel/documents-and-downloads>

E-Mail: documentation@isg-stuttgart.de

Stichwortverzeichnis

P

P-CHAN-00032	27
P-CHAN-00262	28
P-CHAN-00263	28
P-CHAN-00829	29
P-TOOL-00009	26



© Copyright
ISG Industrielle Steuerungstechnik GmbH
STEP, Gropiusplatz 10
D-70563 Stuttgart
Alle Rechte vorbehalten
www.isg-stuttgart.de
support@isg-stuttgart.de

