



DOKUMENTATION ISG-kernel

Funktionsbeschreibung Koppelkinematik

Kurzbezeichnung:
FCT-C35

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

Dokumentation Version: 1.07
12.11.2024

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	6
2 Beschreibung	7
2.1 Anwendungen	7
2.2 Voraussetzungen	8
2.3 Nomenklatur	8
2.4 Einschränkungen und Begrenzungen	9
3 Konfiguration	10
3.1 Konfiguration der Teilkinematiken	10
3.1.1 Besonderheiten der Universalkinematik ID 91	11
3.2 Konfiguration der Gruppen	12
3.2.1 Konfiguration eines Werkstück CS	13
3.3 Konfiguration der kinematischen Kette einer Gruppe	15
3.4 Konfiguration der Priorität für den Lösungsalgorithmus	17
3.5 Konfiguration der Achsen	18
3.6 Werkzeugdaten	20
3.7 Modifikation im NC-Programm	21
3.8 Konfigurationsbeispiele	22
3.8.1 Roboter auf einer XY-Lineareinheit	22
3.8.2 Roboter auf einer XY-Lineareinheit an einem Dreh-Schwenktisch	23
4 Programmierung	25
4.1 Direkte Programmierung der Teilkinematiken	26
4.2 Programmierung des TCP	27
4.3 Mischprogrammierung	28
4.4 Sperren von Freiheitsgraden	29
4.5 Verfügbare V.G.-Variablen	29
5 Parameter	31
5.1 Übersicht	31
5.2 Beschreibung	32
5.2.1 Achsparameter	32
5.2.2 Kanalparameter	33
6 Anhang	39
6.1 Anregungen, Korrekturen und neueste Dokumentation	39
Stichwortverzeichnis	40

Abbildungsverzeichnis

Abb. 1:	Kinematische Kette mit Linearachsen und Roboter	7
Abb. 2:	Kinematische Kette mit X- und Z-Basisversätzen	11
Abb. 3:	Roboter auf einer XY-Lineareinheit an einem Dreh-Schwenktisch	13
Abb. 4:	Aufbau einer Gruppe	15
Abb. 5:	Roboter auf einer XY-Lineareinheit	22
Abb. 6:	Roboter auf einer XY-Lineareinheit an einem Dreh-Schwenktisch	23

1 Übersicht

Aufgabe

Die Koppelkinematik ist eine kinematische Transformation, die es ermöglicht, aus mehreren einzelnen Kinematiken eine größere, komplexe Kinematik aufzubauen. Sie beschreibt den Zusammenhang der einzelnen Teilkinematiken und stellt keine neue Maschinenkinematik zur Verfügung.

Einsatzmöglichkeiten

Die Teilkinematiken können

- als kinematische Kette gestapelt werden,
- als kinematischer Baum parallel betrieben werden,
- oder als Mischung der beiden Varianten eingesetzt werden.

Damit lässt sich beispielsweise zur Erweiterung des Arbeitsraumes eine kinematische Kette aus einem Roboter, der auf einer Linearachse steht, bilden.

Der Parallelbetrieb ermöglicht es z.B. einem Roboter, der an einem Dreh-Schwenktisch arbeitet, den Bewegungen des Tisches zu folgen.



Versionshinweis

Diese Funktionalität ist verfügbar ab CNC-Version V3.1.3080 bzw. V3.1.3108



Hinweis

Diese Funktionalität ist eine lizenzpflichtige Zusatzoption.

Parametrierung

Die Koppelkinematik hat den **Kinematiktyp 210**. Die detaillierte Parametrierung ist vom Aufbau der jeweiligen Maschine abhängig und wird im Kapitel Parameter [▶ 31] näher beschrieben.

Programmierung

Die Programmierung kann als Mischform aus TCP-Programmierung und/oder Programmierung der einzelnen Kinematik erfolgen.

Im Kapitel Programmierung [▶ 25] werden die Programmiervarianten erläutert.

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 Koppelkinematik ermöglicht es, vorhandene kinematische Transformationen zu kombinieren. Sie bietet dabei einen generischen Mechanismus, der es ermöglicht, einzelne Kinematiken als Teile einer großen Kinematik anzuordnen.

Die Kinematiken, die Bestandteil einer Koppelkinematik sind, werden nachfolgend als Teilkinematik bezeichnet.

Diese Teilkinematiken können in 2 Arten angeordnet werden:

1. Seriell als kinematische Kette.
2. Parallel als kinematischer Baum.

Jeder Ast eines kinematischen Baums kann wiederum aus einer kinematischen Kette bestehen. Eine serielle Verkettung kann nicht weiter aufgeteilt werden und kann nur Kinematiken aus der Kinematikbibliothek, ausgenommen der Koppelkinematik, beinhalten.

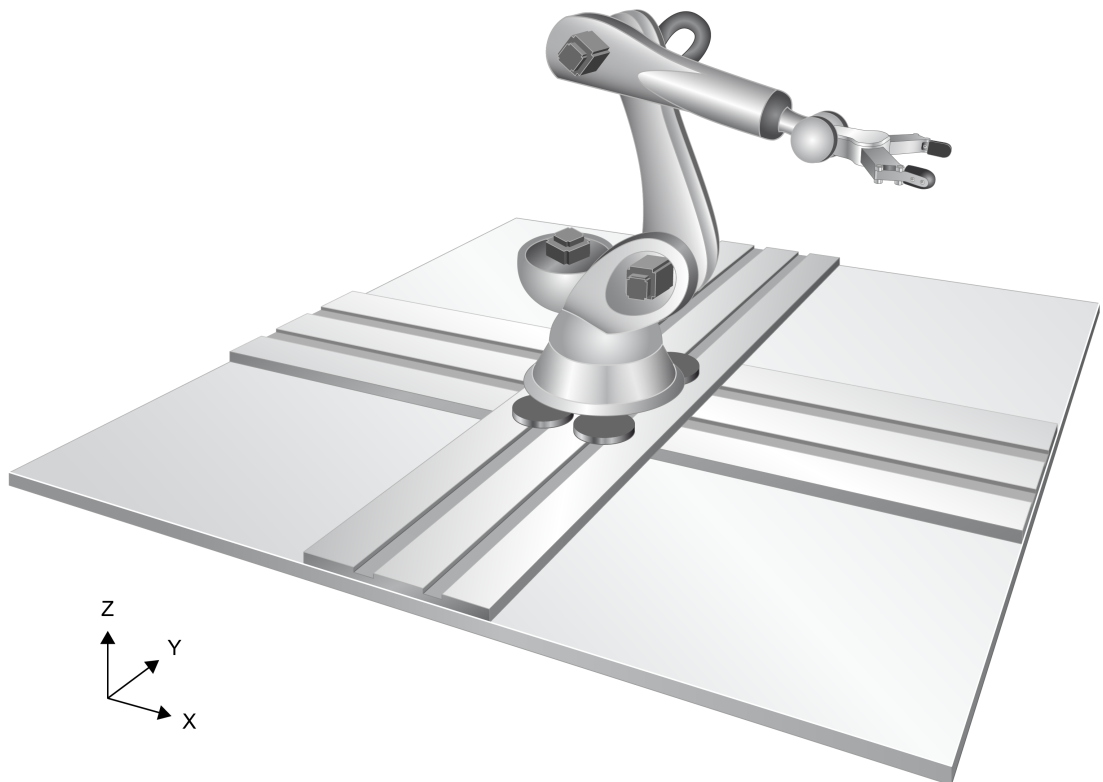


Abb. 1: Kinematische Kette mit Linearachsen und Roboter

2.1 Anwendungen

Die nachfolgenden Punkte beschreiben einige Anwendungsmöglichkeiten für die Koppelkinematik:

- Arbeitsraumerweiterung, z.B. Roboter auf einer Linearachse.
- Kooperation zwischen mehreren Kinematiken, z.B. Roboter am Drehschwenktisch.
- Einwechseln eines hochdynamischen Aggregats an eine tragende Kinematik, z.B. XYZ-Aggregat, das an einem Roboter befestigt ist.

2.2 Voraussetzungen

Für die Koppelkinematik können folgende Kinematiktypen als Teilkinematik verwendet werden:

Kinematik typ	Beschreibung
5	Vierachs-Kinematik mit Kreuzkopf für 4 Werkzeuge
6	Vierachs-Kinematik mit Unterflur-Fräswerkzeug
7	Fünfachs-Kinematik mit manueller Hilfsachse (Bohren)
8	Fünfachs-Kinematik mit manueller Hilfsachse (Sägen)
9	Fünfachs-Kinematik (Bohr- und Fräsaggregat)
10	Fünfachs-Kinematik (Sägen)
45	Sechssachs-Gelenkarm-Roboter
96	Vierachs-Palletierroboter
91	Universelle Kinematik

2.3 Nomenklatur

Die folgende Tabelle soll eine Übersicht über die eingeführten Begriffe der Koppelkinematik bieten.

Begriff	Beschreibung
Teilkinematik	Kinematik aus der Kinematikbibliothek, die Teil einer Koppelkinematik ist.
Gruppe	Einheit aus einer oder mehreren seriell verketteten Teilkinematiken.
Werkstück CS (-Gruppe)	Gruppe, die ein Werkstück hält und deren Bewegung andere Gruppen folgen.
TCP	Tool Center Point; Werkzeugspitze und Ende einer kinematischen Kette. Die Bewegung des TCP ergibt sich aus der überlagerten Bewegung der Teilkinematiken der jeweiligen Gruppe.

2.4 Einschränkungen und Begrenzungen

Folgende Einschränkungen gelten bei der Verwendung der Koppelkinematik

- Es können maximal 3 Kinematiken Teil einer Koppelkinematik sein
- Pro Gruppe können maximal 3 Kinematiken konfiguriert werden
- Es können maximal 2 Gruppen konfiguriert werden
- Der programmierte TCP ist immer der ersten Gruppe (Index 0) zugeordnet
- Es kann nur ein programmierter TCP konfiguriert werden
- Im Handbetrieb (G200) dürfen die Achsen des TCP und einer Teilkinematik nicht gleichzeitig bewegt werden, wenn diese denselben Freiheitsgrad besitzen (z.B. Bewegung in X-Richtung).
Sonst wird die Warnung ID 51065 ausgegeben.

Funktionseinschränkungen der CNC

Folgende Funktionen der CNC sind bei aktiver Koppelkinematik nicht verwendbar und werden mit einer entsprechenden Fehlermeldung unterbunden:

- Handbetrieb mit paralleler Interpolation (G201) (ID 22092)
- Mehrstufige Transformationen (ID 22093)
- 2-Pfadprogrammierung (ID 22094)
- Referenzpunktfahrt mit den TCP-Achsen (ID 22091)
- PTP-Überschleifverfahren, verwenden von `-#CONTOUR MODE[PTP ..]` (ID 22122)

3 Konfiguration

Die Konfiguration der Kinematik erfolgt in mehreren Schritten und wird in den folgenden Unterkapiteln beschrieben.

Die empfohlene Reihenfolge ist:

1. Konfiguration der Teilkinematiken.
2. Konfiguration der Gruppen.
3. Konfiguration der kinematischen Kette jeder Gruppe.
4. Konfiguration der Priorität für den Lösungsalgorithmus, in der die Teilkinematiken einer kinematischen Kette die programmierte TCP-Bewegung ausfahren sollen.
5. Konfiguration der Achsen.

3.1 Konfiguration der Teilkinematiken

Die grundlegende Konfiguration der Teilkinematiken erfolgt wie bisher über die jeweiligen Parameter der Kinematik.



Achtung

Für jede Teilkinematik muss die Kinematik-ID (P-CHAN-00262) eindeutig konfiguriert sein, da die Koppelkinematik ihre Teilkinematiken über diese ID identifiziert. Bei Nichtbeachtung wird der Fehler ID 21931 ausgegeben.

Konfiguration der Basisversätze

Die Basisversätze einer Teilkinematik beschreiben einen statischen, kartesischen Versatz zwischen ihrem MCS und dem TCP der vorherigen Kinematik innerhalb einer kinematischen Kette (P-CHAN-00446 [► 33]).

Der Basisversatz der untersten Teilkinematik einer kinematischen Kette (chain[0]) beschreibt den Versatz zwischen dem MCS dieser Kinematik und dem MCS der Koppelkinematik.

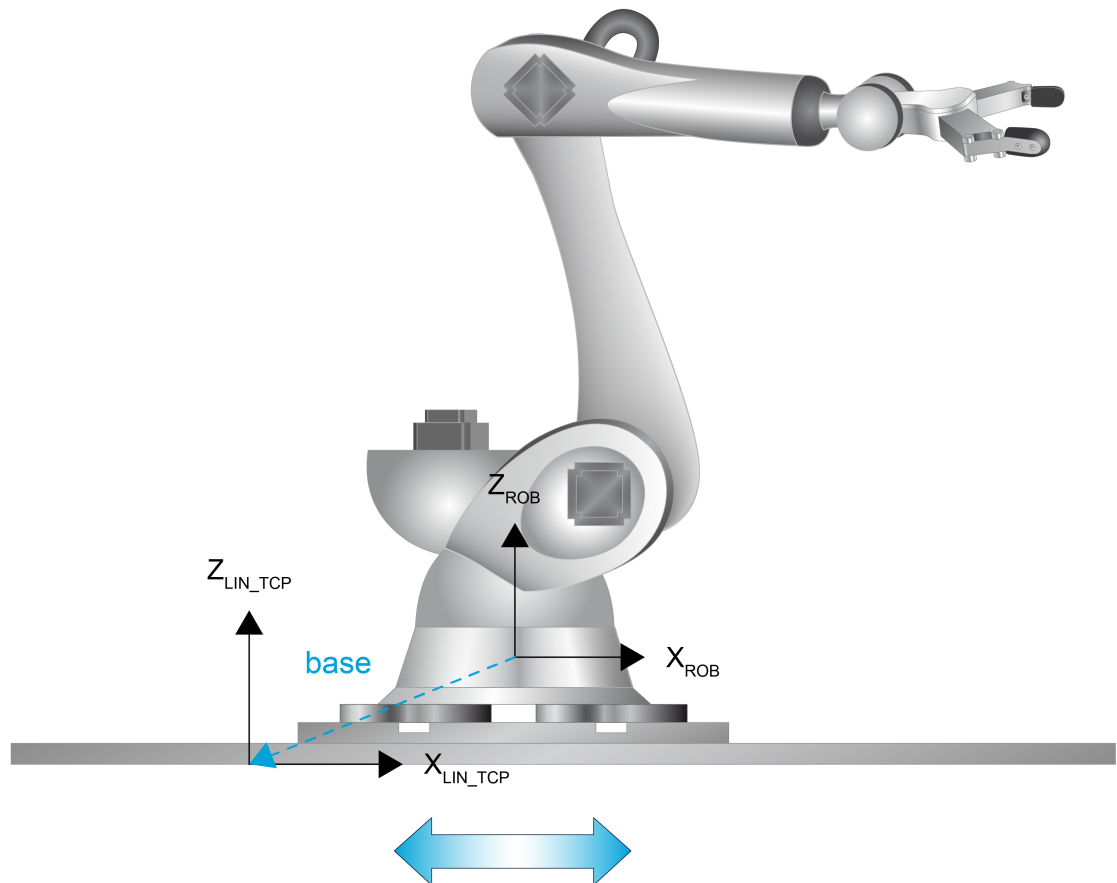


Abb. 2: Kinematische Kette mit X- und Z-Basisversätzen

Aufgrund der Höhe des Schlittens auf dem der Roboter befestigt ist, kann dieser Versatz beispielsweise in Z-Richtung sein.

Eigenschaften

- Bezugssystem: MCS Teilkinematik.
- Sechs Koordinaten können angegeben werden: X-, Y-, Z-, A-, B- und C-Versatz.
- Die Drehreihenfolge der Winkel entspricht der Drehreihenfolge der jeweiligen Kinematik.

```

trafo[0].id      1
trafo[0].type   45
trafo[0].base[0] -100000
trafo[0].base[1] 0
trafo[0].base[2] -350000
...
    
```

3.1.1

Besonderheiten der Universalkinematik ID 91

Für die Universalkinematik gelten folgende Besonderheiten:

- Wenn sie Teil einer Koppelkinematik ist, kann sie mit weniger als drei Linearachsen konfiguriert werden.
- Eine Konfiguration mit null Linearachsen ist ebenfalls möglich.
- Der Parameter `zero_orientation` (P-CHAN-00285) gibt an, in welche Richtung das Basiskoordinatensystem der darauffolgenden Kinematik in der kinematischen Kette orientiert ist.

3.2 Konfiguration der Gruppen

Jede Gruppe steht für eine unabhängige kinematische Kette. Daher müssen so viele Gruppen innerhalb der Koppelkinematik konfiguriert werden, wie kinematische Ketten vorhanden sind. Die notwendigen Parameter zur Konfiguration einer Gruppe sind.

Kanalparameter	Beschreibung
P-CHAN-00447 [▶ 33]	ID der Gruppe (optional)
P-CHAN-00448 [▶ 34]	Das Werkstück der Gruppe wird von der hier angegebenen Gruppe getragen
P-CHAN-00449 [▶ 35]	Kinematische Kette der Gruppe [▶ 15]
P-CHAN-00450 [▶ 36]	Reihenfolge für den Lösungsalgorithmus [▶ 17]

Die Konfiguration des Gruppen-ID (P-CHAN-00447) ist optional.

Die Angabe einer Gruppe, die das Werkstück trägt (P-CHAN-00448), ist ebenfalls optional und nur dann zwingend zu konfigurieren, falls eine Kooperation zwischen den Gruppen gewünscht ist, siehe dazu „Konfiguration eines Werkstück CS [▶ 13]“.



Achtung

Ist P-CHAN-00448 konfiguriert und die angegebene ID entspricht keiner ID einer anderen Gruppe, wird der Fehler ID 292066 ausgegeben.

3.2.1 Konfiguration eines Werkstück CS

Mit dem Parameter P-CHAN-00448 [▶ 34] kann eine Gruppe mit einer anderen kooperieren. Die Gruppe folgt dann der Bewegung der in P-CHAN-00448 [▶ 34] angegebenen Gruppe. Falls die Gruppe einen TCP besitzt, verschiebt sich das PCS des TCP in das Effektorsystem der angegebenen Gruppe.

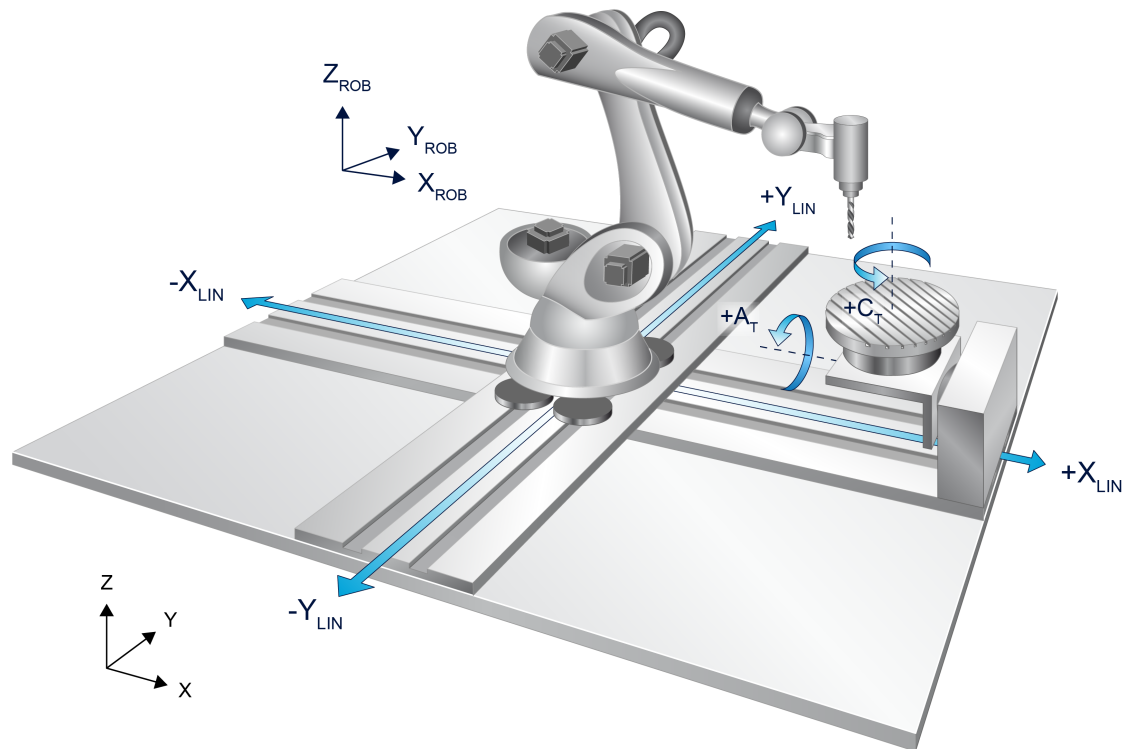


Abb. 3: Roboter auf einer XY-Lineareinheit an einem Dreh-Schwenktisch



Beispiel

Ein Roboter arbeitet am Dreh-Schwenktisch

Der Roboter soll dabei der Bewegung des Dreh-Schwenktisches folgen können. In diesem Beispiel sind zwei unabhängige kinematische Ketten vorhanden:

- Dreh-Schwenktisch
- Roboter

Daher sind zwei Gruppen zu konfigurieren.

Um eine Beziehung zwischen den beiden Ketten herzustellen, kann die Gruppe des Tisches als Werkstück CS der Robotergruppe definiert werden. Das hat zur Folge, dass der Roboter der Bewegung des Tisches folgt.



Beispiel

Das PCS der Gruppe mit aktivem Werkstück CS wird in das bewegte Werkstück verschoben

Im nachfolgenden Beispiel verschiebt sich das PCS der Gruppe mit ID100 in das von der Gruppe ID 200 bewegte Werkstück.

```
trafo[0].id          1
trafo[0].type       45
...
trafo[1].id         2
trafo[1].type       91
...
trafo[2].id         3
trafo[2].type       210
trafo[2].group[0].id 100
trafo[2].group[0].workpiece_cs 200
trafo[2].group[0].chain[0] 1
trafo[2].group[1].id 200
trafo[2].group[1].chain[0] 2
```

Die vollständige Parametrierung ist im Roboter auf einer XY-Lineareinheit an einem Dreh-Schwenktisch [► 23] zu finden.

3.3

Konfiguration der kinematischen Kette einer Gruppe

Die Konfiguration erfolgt über den Parameter `chain[i]` (P-CHAN-00449 [▶ 35]).



Achtung

Nicht vorhandener Kinematik-ID

Kann eine in P-CHAN-00449 angegebene Kinematik_ID nicht in der Liste aller Transformationen gefunden werden, so wird der Fehler ID 292064 ausgegeben.

Die kinematische Kette wird gemäß ihres physikalischen Aufbaus konfiguriert:

- `chain[0]`: Kinematik, die von keiner anderen Kinematik getragen wird
- `chain[i]`: Kinematik, die von der Kinematik `chain[i-1]` getragen wird
- Höchster Index: Kinematik, die keine andere Kinematik trägt

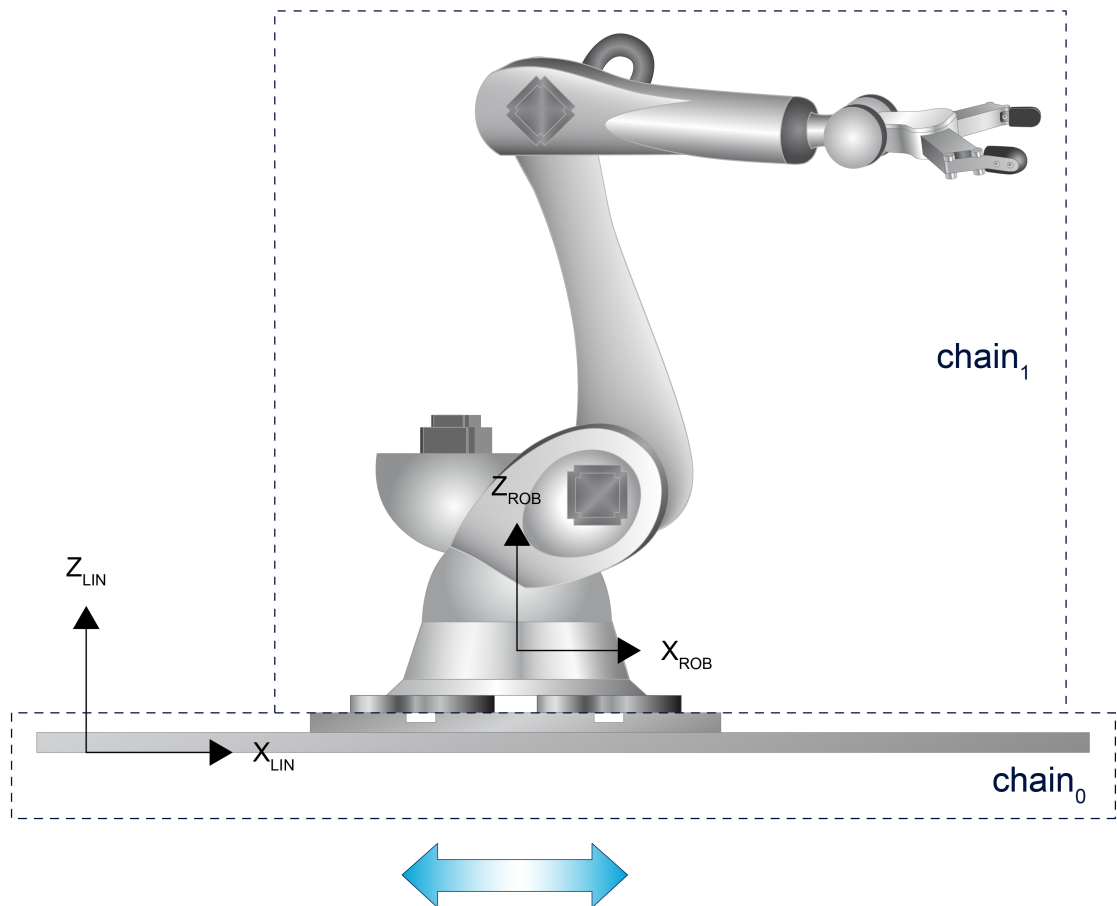


Abb. 4: Aufbau einer Gruppe



Beispiel

Ein Roboter steht auf einer Linearachse

```
trafo[0].id          1
trafo[0].type       45
...
trafo[1].id          2
trafo[1].type       91
...
trafo[2].id          3
trafo[2].type       210
trafo[2].group[0].id 100
trafo[2].group[0].chain[0] 2
trafo[2].group[0].chain[1] 1
...
```


3.4 Konfiguration der Priorität für den Lösungsalgorithmus

Die Teilkinematiken einer kinematischen Kette können direkt mittels der konfigurierten Achsbezeichner programmiert werden. Weiterhin kann auch der TCP programmiert werden. Die Bewegung des TCPs wird dann auf die Teilkinematiken aufgeteilt. Dabei wird jeder Teilkinematik der Anteil der Bewegung zugewiesen, den sie aufgrund ihrer kartesischen Freiheitsgrade fahren kann.



Achtung

Nicht ausfahrbare Bewegungen möglich

Die Grenzen des Arbeitsraumes der Teilkinematiken werden nicht berücksichtigt. Einer Teilkinematik kann eine Bewegung zugewiesen werden, die diese konstruktionsbedingt nicht ausfahren kann.

Die dabei auftretenden Fehler sind abhängig von der jeweiligen Teilkinematik.

Die Konfiguration der Reihenfolge erfolgt durch den Parameter `move_prio[i]` (P-CHAN-00450 [▶ 36]). Ist der Parameter für eine Gruppe nicht konfiguriert, entsprechen die Parameterwerte denjenigen aus der kinematischen Kette (P-CHAN-00449 [▶ 35]): `move_prio[i] = chain[i]`.



Achtung

Unvollständige Konfiguration

Es muss entweder für jeden Eintrag in der kinematischen Kette ein Eintrag in der Prioritätenliste vorhanden, oder es darf überhaupt kein Eintrag in der Prioritätenliste vorhanden sein. Diese Bedingung gilt für jede Gruppe individuell. Ist dies nicht der Fall, wird bei Anwahl der Transformation der Fehler ID 292067 ausgegeben.



Beispiel

Ein Roboter steht auf einer Linearachse

Bei programmiertem TCP wird die Bewegung zunächst der Linearachse zugewiesen. Enthält die programmierte Bewegung Freiheitsgrade, die die Linearachse nicht ausfahren kann, werden diese Bewegungen an den Roboter weitergegeben.

```

trafo[0].id           1
trafo[0].type        45
...
trafo[1].id           1
trafo[1].type        91
...
trafo[2].id           3
trafo[2].type        210
trafo[2].group[0].id 100
trafo[2].group[0].chain[0] 2
trafo[2].group[0].chain[1] 1
trafo[2].group[0].move_prio[0] 2    (Linearachse führt)
                                     (zuerst die TCP-Bewegung aus)
trafo[2].group[0].move_prio[1] 1    (Roboter führt)
                                     (übrige Bewegung aus)
    
```

3.5 Konfiguration der Achsen

Die Achskonfiguration der Teilkinematiken wird im Unterkapitel Konfiguration der Teilkinematiken [▶ 10] detaillierte beschrieben.



Achtung

Neben den Achsen, die für die Teilkinematiken benötigt werden, müssen sechs weitere Simulationsachsen für den TCP konfiguriert werden. Diese müssen auf den Achsindizes 0-5 des Kanals liegen.



Hinweis

Es ist nur ein TCP erlaubt. Dieser bezieht sich immer auf die erste Gruppe (Index 0). Die Kinematiken der übrigen Gruppen müssen direkt programmiert werden.

Folgende Parameter müssen für alle sechs TCP-Achsen konfiguriert sein:

Parameter	Parameternummer	Wert
kenngr.antr_typ	P-AXIS-00020 [▶ 32]	4
kenngr.messachse	P-AXIS-00118 [▶ 32]	0



Hinweis

Messen mit TCP-Achsen

Die TCP-Achsen latchen keine Messwerte, da sie keine physikalischen Achsen sind. Deshalb müssen sie auch nicht als Messachse konfiguriert werden.

Es kann dennoch eine Messfahrt mit den TCP-Achsen bei aktiver Transformation durchgeführt werden. Die kartesischen Messwerte des TCPs werden dann aus den Messwerten der einzelnen Teilkinematiken errechnet.

Die Achsreihenfolge im Kanal ist nach folgendem Schema anzulegen:

1. Beginnend beim Gruppenindex 0 werden die Achsen für die erste Kinematik der kinematischen Kette (chain[0]) nach den TCP-Achsen im Kanal einsortiert.
2. Es folgen die Achsen der nächsten Kinematik (chain[1]) usw.
3. Für die übrigen Gruppen (Index 1 bis n) wird nach demselben Prinzip verfahren.



Beispiel

Achskonfiguration eines 6-Achs-Roboters auf Linearachse

Achskonfiguration eines 6-Achs-Roboter, der auf einer X-Linearachse steht und an einem Dreh-Schwenktisch (2-Drehachsen) arbeitet.

```
trafo[2].id           3
trafo[2].type        210
trafo[2].group[0].id 100
trafo[2].group[0].chain[0] 2
trafo[2].group[0].chain[1] 1
trafo[2].group[1].id 200
trafo[2].group[1].chain[1] 3
```

Die Achsreihenfolge ist folgendermaßen festzulegen:

- Achsindex 0-5: TCP-Achsen
- Achsindex 6: Achse der Kinematik mit ID 2 (X-Linearachse)
- Achsindex 7-12: Achsen der Kinematik mit ID 1 (6-Achs-Roboter)
- Achsindex 13-14: Achsen der Kinematik mit ID 3 (Dreh-Schwenktisch)



Achtung

Die Achsreihenfolge muss zwingend nach dem beschriebenen Schema eingehalten werden.

Die auftretenden Fehler hängen davon ab, welche Achsen fälschlicherweise einer anderen Kinematik zugewiesen werden.

3.6

Werkzeugdaten

Das Werkzeug wird auf die Teilkinematik mit dem größten Index der ersten Gruppe eingewechselt.

Die im Werkzeug konfigurierten Kinematikparameter addieren sich auf die Kinematikparameter dieser Teilkinematik.

3.7 Modifikation im NC-Programm

Folgende Parameter sind im NC-Programm über V.G.KIN-Variablen konfigurierbar.

ID	V.G.KIN-Variable
P-CHAN-00446	V.G.KIN[i].BASE[j]
P-CHAN-00447	V.G.KIN[210].GROUP[i].ID
P-CHAN-00448	V.G.KIN[210].GROUP[i].WORKPIECE_CS
P-CHAN-00449	V.G.KIN[210].GROUP[i].CHAIN[j]
P-CHAN-00450	V.G.KIN[210].GROUP[i].MOVE_Prio[j]
P-CHAN-00458	V.G.KIN[i].LOCK_DOF[j]

Werden die V.G. Variablen bei aktiver Transformation programmiert sind die neuen Kinematikparameter erst bei erneuter Aktivierung der Transformation aktiv.

Um bei aktiver Transformation die gesperrten Freiheitsgrade zu verändern, steht der Befehl #KIN DATA [LOCKDOF | UNLOCKDOF] zur Verfügung.

3.8 Konfigurationsbeispiele

3.8.1 Roboter auf einer XY-Lineareinheit



Beispiel

Konfiguration mit einer Gruppe und mehreren Teilkinematiken

Bei dieser Konfiguration wird eine Gruppe mit den Kinematiktypen 45 und 91 benötigt. Die Lineareinheit wird mit der Universalkinematik 91 mit zwei Achsen realisiert.

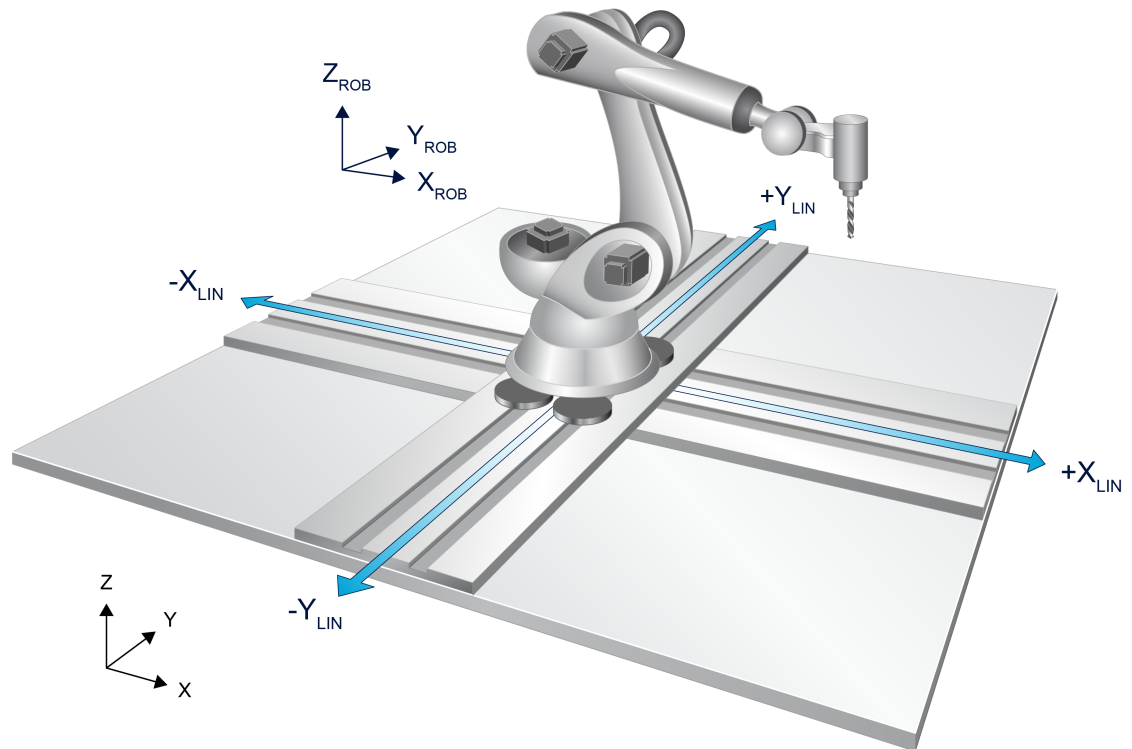


Abb. 5: Roboter auf einer XY-Lineareinheit

```

trafo[0].id                1
trafo[0].type              45
trafo[0].base[2]          200000
trafo[0].param[0] ...

trafo[1].id                2
trafo[1].type              91
trafo[1].zero_orientation[0] 0.0
trafo[1].zero_orientation[1] 0.0
trafo[1].zero_orientation[2] 1.0
trafo[1].zero_position[0]   0.0
trafo[1].zero_position[1]   0.0
trafo[1].zero_position[2]   0.0
trafo[1].number_of_axes    2
trafo[1].chain[0]          0
trafo[1].chain[1]          1
trafo[1].axis[0].type      1
trafo[1].axis[0].orientation[0] 1.0
trafo[1].axis[0].orientation[1] 0.0
    
```

```

trafo[1].axis[0].orientation[2] 0.0
trafo[1].axis[1].type          1
trafo[1].axis[1].orientation[0] 0.0
trafo[1].axis[1].orientation[1] 1.0
trafo[1].axis[1].orientation[2] 0.0
trafo[1].programming_mode      12
trafo[1].rtcp                  1
trafo[1].base[2]               100000

trafo[2].id                    3
trafo[2].type                  210
trafo[2].group[0].id          100
trafo[2].group[0].chain[0]    2
trafo[2].group[0].chain[1]    1
trafo[2].group[0].move_prio[0] 2
trafo[2].group[0].move_prio[1] 1
    
```

3.8.2 Roboter auf einer XY-Lineareinheit an einem Dreh-Schwenktisch



Beispiel

Konfiguration mit mehreren Gruppen, die untereinander verknüpft sind

Bei dieser Konfiguration werden zwei Gruppen benötigt. Die erste Gruppe besteht aus der Lineareinheit und dem Roboter. Die Konfiguration dieser Gruppe ist analog zum vorherigen Beispiel. Die zweite Gruppe beinhaltet den Dreh-Schwenktisch, der mit der Universalkinematik 91 aus zwei Rotationsachsen konfiguriert wird.

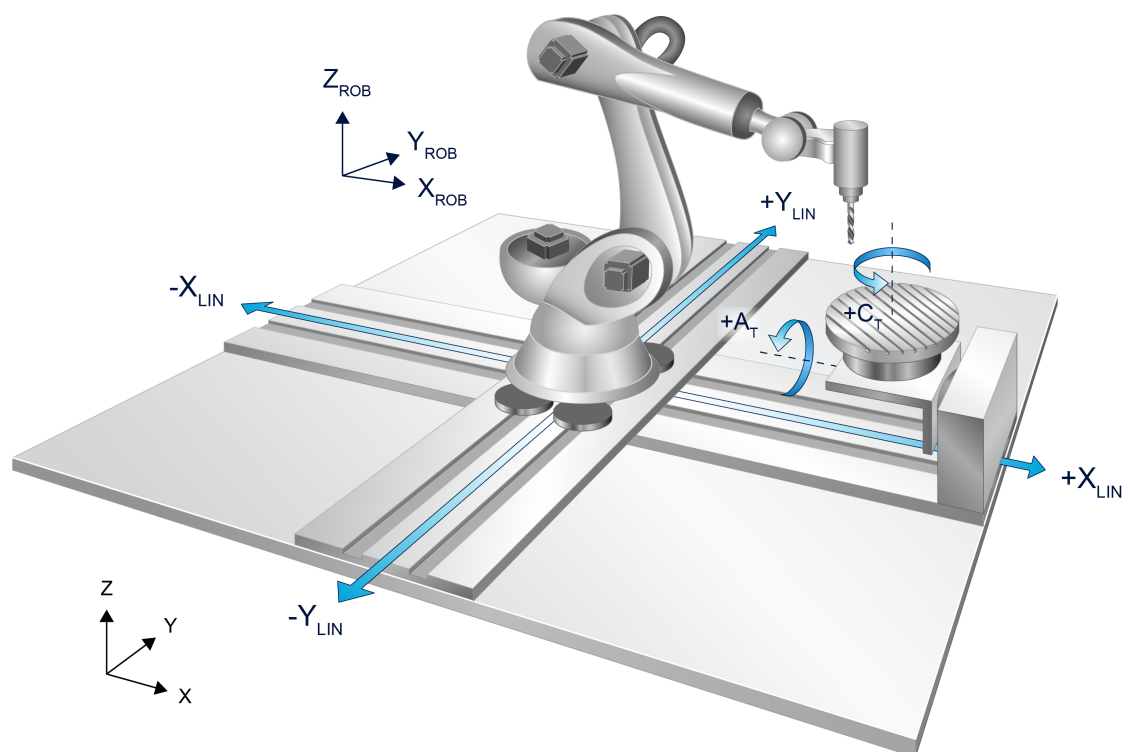


Abb. 6: Roboter auf einer XY-Lineareinheit an einem Dreh-Schwenktisch

Da die Konfiguration von XY-Lineareinheit und Roboter exakt dem vorherigen Beispiel [► 22] entsprechen, wird hier nicht mehr auf die Konfiguration der betroffenen Elemente eingegangen.

```
trafo[0].id 1
trafo[0].type 45
...
trafo[1].id 2
trafo[1].type 91
...
trafo[2].id 3
trafo[2].type 91
trafo[2].zero_orientation[0] 0.0
trafo[2].zero_orientation[1] 0.0
trafo[2].zero_orientation[2] 1.0
trafo[2].zero_position[0] 0.0
trafo[2].zero_position[1] 0.0
trafo[2].zero_position[2] 0.0
trafo[2].number_of_axes 2
trafo[2].chain[0] 1
trafo[2].chain[1] 0
trafo[2].axis[0].type 2
trafo[2].axis[0].orientation[0] -1.0
trafo[2].axis[0].orientation[1] 0.0
trafo[2].axis[0].orientation[2] 0.0
trafo[2].axis[0].point[0] 0.0
trafo[2].axis[0].point[1] 0.0
trafo[2].axis[0].point[2] 0.0
trafo[2].axis[1].type 2
trafo[2].axis[1].orientation[0] 0.0
trafo[2].axis[1].orientation[1] 0.0
trafo[2].axis[1].orientation[2] -1.0
trafo[2].axis[1].point[0] 0.0
trafo[2].axis[1].point[1] 0.0
trafo[2].axis[1].point[2] 0.0
trafo[2].programming_mode 12
trafo[2].rtcp 1
trafo[2].base[0] 30000000
trafo[2].base[2] 10000000

trafo[3].id 4
trafo[3].type 210
trafo[3].group[0].id 100
trafo[3].group[0].chain[0] 2
trafo[3].group[0].chain[1] 1
trafo[3].group[0].move_prio[0] 2
trafo[3].group[0].move_prio[1] 1
trafo[3].group[0].workpiece_cs 200
trafo[3].group[1].id 200
trafo[3].group[1].chain[0] 3
trafo[3].group[1].move_prio[0] 3
```


4 Programmierung



Hinweis

Die volle CNC-Funktionalität steht nur bei der TCP-Programmierung zur Verfügung. Bei der Programmierung von Teilkinematiken sind nur Linearsätze möglich.

Die gesamten Funktionen des NC-Kerns stehen nur für die Programmierung des TCPs und damit für die erste Gruppe zur Verfügung. Für alle Teilkinematiken können lediglich Linearsätze programmiert werden. Es ist nicht möglich, für Teilkinematiken weitere NC Funktionen zu nutzen, wie z.B. Kreisinterpolation oder die Nutzung von Bearbeitungskoordinatensystemen.

Das Programmierkoordinatensystem (PCS) der Teilkinematiken kann nicht verändert werden und befindet sich im Maschinennullpunkt der jeweiligen Koordinaten.

Das PCS des TCPs kann zusätzlich zu den bereits vorhandenen Methoden mit der Angabe des Werkstück CS (P-CHAN-00448 [▶ 34]) in das bewegte Effektorsystem einer anderen Gruppe verschoben werden.

Die folgende Koppelkinematik besteht aus

- einer XY-Lineareinheit
- darauf montiert ein 6-Achs-Roboter
- Lineareinheit und Roboter sind in X- und Y-Richtung ausgerichtet



Beispiel

Beispielkonfiguration Roboter auf XY-Lineareinheit

Nachfolgende Konfiguration soll in den Programmierbeispielen verwendet werden. Aus Gründen der Übersichtlichkeit sind die Teilkinematiken und Achsen hier nicht vollständig konfiguriert.

```
trafo[0].id          1
trafo[0].type       45
...
trafo[1].id          2
trafo[1].type       91
...
trafo[2].id          3
trafo[2].type       210
trafo[2].group[0].id 100
trafo[2].group[0].chain[0] 2
trafo[2].group[0].chain[1] 1
trafo[2].group[0].move_prio[0] 2
trafo[2].group[0].move_prio[1] 1
...
```

Die Achsbezeichner lauten:

- XY-Lineareinheit
 - X_LIN, Y_LIN
- Roboter
 - X_ROB, Y_ROB, Z_ROB, A_ROB, B_ROB, C_ROB
- TCP
 - X_TCP, Y_TCP, Z_TCP, A_TCP, B_TCP, C_TCP

Programmierung von RTCP-Kinematiken

Ausgangssituation: Die Koppelkinematik besteht aus einer 5-Achs RTCP-Kinematik und weiteren Teilkinematiken.

Der Orientierungswinkelmodus (P-CHAN-00112) muss so eingestellt werden, dass die Drehreihenfolge der Reihenfolge der Drehachsen in der RTCP-Kinematik entspricht. Im TCP sollten nur die Winkel programmiert werden, wie sie auch für die RTCP-Kinematik programmiert werden würden.



Beispiel

RTCP-Kinematiken – zu programmierende Achsen

Kinematik	Programmiermodus	Zu programmierende TCP-Achsen
CA	CAB (3)	A, C
AC	ABC (5)	A, C
BA	CBA_STD (4)	B, C
AB	ABC (5) oder CAB (3)	A, B

4.1 Direkte Programmierung der Teilkinematiken

Die Teilkinematiken können nur durch Linearsätze programmiert werden. Für erweiterte Bewegungen wie Kreise muss der TCP programmiert werden.



Programmierbeispiel

Direkte Programmierung von Roboter und Linearachsen

```
N010 #KIN ID [3]
N020 #TRAFO ON
N030 G00 G90 X_ROB=1000 X_LIN=500
N040 G00 G90 Y_ROB=300 Y_LIN=0
N050 #TRAFO OFF
```

Nach Programmende lauten die Koordinaten:

	X	Y
Lineareinheit	500	0
Roboter	1000	300
TCP	1500	300

4.2 Programmierung des TCP



Programmierbeispiel

Lineare und rotatorische Bewegung des TCPs

```

N010 #KIN ID [3]
N020 #TRAFO ON
N030 G00 G90 X_TCP=1000 Y_TCP=500 Z_TCP=50 C_TCP=10
N040 #TRAFO OFF
    
```

Die Bewegung wird folgendermaßen aufgeteilt:

	X	Y	Z	A	B	C
Lineareinheit	1000	500				
Roboter	0	0	50	0	0	10
TCP	1000	500	50	0	0	10

Die Lineareinheit kann aufgrund ihrer Freiheitsgrade nur Bewegungen in X- und Y-Richtung ausführen. Die in N030 programmierte Bewegung in Z- und C-Richtung werden an den Roboter weitergegeben.

4.3 Mischprogrammierung

Programmierung von TCP-Koordinaten und Koordinaten der Teilkinematiken im selben NC-Befehl.



Programmierbeispiel

Lineare und rotatorische Bewegung des TCPs sowie feste Programmierung der Lineareinheit.

Zusätzliche Programmierung der Linearachsen in Satz N030.

```
N010 #KIN ID [3]
N020 #TRAFO ON
N030 G00 G90 X_TCP=1000 Y_TCP=500 Z_TCP=50 C_TCP=10 X_LIN=800 Y_LIN=0
N040 #TRAFO OFF
```

Die Bewegung wird folgendermaßen aufgeteilt:

	X	Y	Z	A	B	C
Lineareinheit	800	0				
Roboter	200	500	50	0	0	10
TCP	1000	500	50	0	0	10

Der Roboter muss die Differenz von TCP-Position und Position der Linearachsen in X- und Y-Richtung ausfahren.



Programmierbeispiel

Kreisinterpolation im TCP und feste Programmierung einzelner Koordinaten

```
N010 #KIN ID [3]
N020 #TRAFO ON
; Anfahren der Startpose von Lineareinheit und Roboter
N030 G00 G90 X_LIN=0 Y_LIN=0
N040 G00 G90 X_ROB=500 Y_ROB=0 Z_ROB=1500
; Kreisinterpolation
N050 G02 G91 F5000 X_TCP=0 Y_TCP=-500 R250 X_LIN=0 Y_ROB=0
N060 #TRAFO OFF
```

Die Bewegung wird folgendermaßen aufgeteilt:

	X	Y	Z	A	B	C
Lineareinheit	0	-500				
Roboter	500	0	1500	0	0	10
TCP	500	-500	1500	0	0	10

Im Satz N050 übernimmt die Y-Linearachse die Bewegung in Y-Richtung. Der Roboter führt simultan die Bewegung in X-Richtung aus. Die programmierten Teilkinematiken fahren hierbei nur lineare Bewegungen.

4.4 Sperren von Freiheitsgraden

Programmierung von TCP Koordinaten und Sperren von Freiheitsgraden von Teilkinematiken.



Programmierbeispiel

Lineare Bewegung des TCPs, sperren der Y-Achse der Lineareinheit.

```
N010 #KIN ID [3]
N020 #TRAFO ON
N030 #KIN DATA [LOCKDOF AX=Y_LIN]
N040 G00 G90 X_TCP=1000 Y_TCP=500 Z_TCP=50 X_LIN=800
N050 #KIN DATA [UNLOCKDOF AX=Y_LIN]
N060 #TRAFO OFF
```

Die Bewegung wird folgendermaßen aufgeteilt:

	X	Y	Z	A	B	C
Lineareinheit	800	0				
Roboter	200	500	50	0	0	0
TCP	1000	500	50	0	0	0

Da in Satz N030 die Y-Achse der Lineareinheit (Y_LIN) von der Aufteilung der TCP-Bewegung ausgenommen wird, fährt der Roboter die programmierte Y-Bewegung des TCPs.

4.5 Verfügbare V.G.-Variablen

Zusätzlich stehen für den Zugriff auf die Daten der Koppelkinematik die nachfolgend aufgeführten Variablen zur Verfügung.

V.G.KIN[j].*	Bedeutung	Datentyp	Einheit der Ein/ Ausgabe	L/S*
BASE[k]	Basisversatz einer Kinematik (P-CHAN-00446 [► 33]) <k>:= 0 - 5	Real	[0.1 µm, 10 ⁻⁴ °]	L/S
GROUP[k].NAME	Name einer Gruppe (P-CHAN-00447 [► 33]) max. 16 Zeichen <k>:= 0, 1	Integer	-	L/S
GROUP[k].WORKPIECE_CS	Gruppe trägt das Werkstück (P-CHAN-00448 [► 34]) max. 16 Zeichen <k>:= 0, 1	Integer	-	L/S
GROUP[k].CHAIN[l]	Definition der kinematischen Kette der Gruppe (P-CHAN-00449 [► 35]) max. 16 Zeichen- <k>:= 0, 1 <l>:= 0, 1, 2	Integer	-	L/S
GROUP[k].MOVE_PRIO[l]	Definition der Reihenfolge zur Aufteilung der TCP-Bewegung auf die Teilkinematiken (P-CHAN-00450 [► 36]) max. 16 Zeichen <k>:= 0, 1 <l>:= 0, 1, 2	Integer	-	L/S
LOCK_DOF[k]	Gesperrte, kartesische Freiheitsgrade, die vom Lösungsalgorithmus nicht bewegt werden können. (P-CHAN-00458 [► 38]) <k>:= 0 -15 Bsp.: Sperren der Achse mit Index 4 V.G.KIN[210].LOCK_DOF[4] = 1	Boolean	-	L/S

5 Parameter

5.1 Übersicht

Achsparameter

ID	Parameter	Beschreibung
P-AXIS-00020	kenngr.antr_typ	Antriebstyp
P-AXIS-00118	kenngr.messachse	Achse als Messachse kennzeichnen

Kanalparameter

ID	Parameter	Beschreibung
P-CHAN-00446	base[i]	Basisversatz einer Kinematik
P-CHAN-00447	id	ID einer Gruppe
P-CHAN-00448	workpiece_cs	Diese Gruppe trägt das Werkstück (nur für Kooperationen)
P-CHAN-00449	chain[i]	Definition der kinematischen Kette der Gruppe
P-CHAN-00450	move_prio[i]	Definition der Reihenfolge zur Aufteilung der TCP-Bewegung auf die Teilkinematiken
P-CHAN-00452	default_name_of_kin_step[i]	Name der Standardtransformation für mehrstufige Transformationen
P-CHAN-00454	kinematik_name	Name der Standardtransformation
P-CHAN-00458	lock_dof[i]	Gesperrte, kartesische Freiheitsgrade, die vom Lösungsalgorithmus nicht bewegt werden können.

5.2 Beschreibung

5.2.1 Achsparameter

P-AXIS-00020	Antriebstyp	
Beschreibung	Über den Parameter wird der Antriebstyp der jeweiligen Achse festgelegt. Für jede Achse kann zwischen den folgenden Antriebstypen gewählt werden.	
Parameter	kenngr.antr_typ	
Datentyp	UNS16	
Datenbereich	1 : Konventionelle Antriebsschnittstelle 2 : Antriebsschnittstelle SERCOS 3 : Antriebsschnittstelle PROFIDRIVE MC 4 : Antriebssimulation 5 : Antriebsschnittstelle Beckhoff Lightbus 6 : Antriebsschnittstelle +-10V über Feldbus (Terminal) 7 : Antriebsschnittstelle RT-Ethernet 8 : Antriebsschnittstelle CANopen 16 : Virtuelle Achse 32 : CAN-Bus	
Achstypen	T, R, S	
Dimension	T: ----	R,S: ----
Standardwert	0x0004	
Antriebstypen	----	
Anmerkungen	Dieser Eintrag wird beim Aktualisieren der Achsparameterliste nicht übernommen, zur Aktualisierung ist ein Neustart der Steuerung notwendig. Die Daten des Achsparametersatzes können in antriebstypabhängige und antriebstypunabhängige Parameter gegliedert werden.	

P-AXIS-00118	Achse als Messachse kennzeichnen	
Beschreibung	Der Parameter muss für alle Achsen mit TRUE belegt werden, die an einer Messfahrt beteiligt sind.	
Parameter	kenngr.messachse	
Datentyp	BOOLEAN	
Datenbereich	0/1	
Achstypen	T, R	
Dimension	T: ----	R: ----
Standardwert	0	
Antriebstypen	----	
Anmerkungen		

5.2.2 Kanalparameter

P-CHAN-00446	Definition der kartesischen Basisversätze für gestapelte Kinematiken.
Beschreibung	<p>Die Basisversätze sind für gestapelte Kinematiken aktiv und beschreiben die kartesischen Versätze zwischen dem MCS einer Kinematik und dem TCP der im Stapel darunterliegenden Kinematik.</p> <p>Ein Roboter sitzt auf dem Schlitten einer Linearachse. Durch den Schlitten ergibt sich ein Versatz in z-Richtung zwischen Linearachse und Roboterfuß von 10cm. Dieser Versatz für den Roboter ist hier einzutragen:</p> <pre>trafo[0].id 45 trafo[0].base[2] -1000000</pre>
Parameter	trafo[j].base[k] mit k=0 ... 5 kin_step[i].trafo[j].base[k] (mehrstufige Transformationen)
Datentyp	REAL64
Datenbereich	MIN(REAL64) ... MAX(REAL64)
Dimension	0.1µm für Linearversätze, 0.0001° für Winkel
Standardwert	0.0
Anmerkungen	Die Basisversätze werden nur bei aktiver Koppelkinematik berücksichtigt.

P-CHAN-00447	Definition einer Gruppen-ID einer Koppelkinematik
Beschreibung	Eine Gruppe einer Koppelkinematik vereint eine Anzahl an gestapelten Teilkinematiken und ist durch die ID eindeutig identifizierbar.
Parameter	trafo[j].group[k].id kin_step[i].trafo[j].group[k].id (mehrstufige Transformationen)
Datentyp	UNS16
Datenbereich	0 ... MAX(UNS16)
Dimension	----
Standardwert	0
Anmerkungen	Ab V3.1.3080.09 Eine Gruppen-ID darf innerhalb einer Koppelkinematik nicht mehrfach vergeben werden.

P-CHAN-00448	Definition eines Werkstück-CS für eine Gruppe einer Koppelkinematik.
Beschreibung	<p>Falls ein Werkstück-CS angegeben wird (Gruppen-ID einer anderen Gruppe der Koppelkinematik), werden alle TCP Koordinaten dieser Gruppe im TCP-System der als Werkstück-CS angegebenen Gruppe interpretiert.</p> <p>Diese Gruppe folgt also den Bewegungen der Werkstück-CS Gruppe.</p>
Parameter	<p>trafo[j].group[k].workpiece_cs</p> <p>kin_step[i].trafo[j].group[k].workpiece_cs (mehrstufige Transformationen)</p>
Datentyp	UNS16
Datenbereich	0 ... MAX(UNS16)
Dimension	----
Standardwert	0
Anmerkungen	<p>Konfigurationsbeispiel:</p> <pre>trafo[0].id 210 trafo[0].group[0].id 100 trafo[0].group[1].id 200 trafo[0].group[1].workpiece_cs 100</pre> <p>Nach Anwahl der Kinematik (ID 210) folgt die Gruppe 200 den Bewegungen der Gruppe 100, da diese als Werkstückkoordinatensystem von 200 konfiguriert ist.</p> <p>Ab V3.1.3080.09</p>

P-CHAN-00449	Definition der kinematischen Kette für eine Gruppe einer Koppelkinematik
Beschreibung	Beschreibung der Reihenfolge, in der die Teilkinematiken einer Koppelkinematikgruppe aufeinander gestapelt sind. Index l = 0 zeigt auf die unterste Kinematik.
Parameter	trafo[j].group[k].chain[l] kin_step[i].trafo[j].group[k].chain[l] (mehrstufige Transformationen)
Datentyp	UNS16
Datenbereich	0 ... MAX(UNS16)
Dimension	----
Standardwert	0
Anmerkungen	<p>Jede hier verwendete Kinematik muss als eigenständige Kinematik mit entsprechender ID konfiguriert sein.</p> <p>Konfigurationsbeispiel: Roboter auf einer Linearachse</p> <pre> trafo[0].id 1 trafo[0].type 210 trafo[0].group[0].id 100 trafo[0].group[0].chain[0] 3 trafo[0].group[0].chain[1] 2 trafo[1].id 2 trafo[1].type 45 trafo[2].id 3 trafo[2].type 91 </pre> <p>Ab V3.1.3080.09</p>

P-CHAN-00450	Priorität, nach der die Koppelkinematik die TCP Bewegung auf die Teilkinematiken aufteilt.
Beschreibung	Der Algorithmus, der die programmierten TCP Koordinaten auf die Teilkinematiken aufteilt, geht nach der hier definierten Priorität vor: Beginnend beim Index $m = 0$ soll der größtmögliche Teil der Bewegung von dieser Kinematik gefahren werden. Die resultierende Differenz zum programmierten TCP wird an die nächste Kinematik in der Liste weitergegeben. Dieser Schritt wird solange wiederholt, bis die TCP Koordinaten erreicht sind, oder der letzte Eintrag der hier definierten Reihenfolge erreicht ist.
Parameter	trafo[j].group[k].move_prio[m] mit $m = 0 \dots$ Länge der kinematischen Kette kin_step[i].trafo[j].group[k].move_prio[m] (mehrstufige Transformationen)
Datentyp	UNS16
Datenbereich	0 ... MAX(UNS16)
Dimension	----
Standardwert	0
Anmerkungen	<p>Jeder hier verwendete Name muss in der kinematischen Kette vorhanden sein.</p> <p>Konfigurationsbeispiel: Roboter auf einer Linearachse</p> <pre> trafo[0].id 1 trafo[0].type 210 trafo[0].group[0].id 100 trafo[0].group[0].chain[0] 3 trafo[0].group[0].chain[1] 2 trafo[0].group[0].move_prio[0] 2 trafo[0].group[0].move_prio[1] 3 trafo[1].id 2 trafo[1].type 45 trafo[2].id 3 trafo[2].type 91 </pre> <p>In diesem Beispiel wird zunächst die gesamte TCP Bewegung an den Roboter übergeben. Was dieser nicht ausfahren kann, wird dann von der Linearachse übernommen.</p> <p>Ab V3.1.3080.09</p>

P-CHAN-00452	Auswahl der Standardkinematiknamen für mehrstufige Transformationen
Beschreibung	<p>Der Name einer Kinematik dient zur eindeutigen Identifikation, falls eine Kinematik-ID mehrfach konfiguriert sein sollte.</p> <p>Für jede Transformationsstufe kann ein Standardkinematikname festgelegt werden. Dieser ist nach Hochlauf der Steuerung gültig. Die Standardkinematiken müssen in den jeweiligen Transformationsstufen konfiguriert sein, ansonsten erfolgt bei Anwahl der Transformationen (#TRAFO ...) eine Fehlermeldung.</p>
Parameter	default_name_of_kin_step[i] mit i = 0, 1
Datentyp	STRING
Datenbereich	Maximal 16 Zeichen
Dimension	----
Standardwert	-
Anmerkungen	<p>Parametrierbeispiel:</p> <p>Nach Steuerungshochlauf ist die Transformation mit ID 2 und Name "test" für die erste Transformationsstufe (.step[0]) gültig.</p> <p><i>default_id_of_kin_step[0] 2</i> <i>default_name_of_kin_step[0] test</i></p>

P-CHAN-00454	Name der Standardkinematik nach Hochlauf
Beschreibung	<p>Falls eine Kinematik-ID mehrfach konfiguriert ist, wird mit diesem Parameter der Name der Standardkinematik nach Hochlauf eindeutig bestimmt.</p>
Parameter	kinematik_name
Datentyp	STRING
Datenbereich	Maximal 16 Zeichen
Dimension	----
Standardwert	-
Anmerkungen	<p>Parametrierbeispiel:</p> <p>Die nach Hochlauf gültige Standardkinematik hat die ID 2 mit dem Namen "test "</p> <p><i>kinematik_id 2</i> <i>kinematik_name test</i></p>

P-CHAN-00458	Sperrungen von kartesischen Freiheitsgraden für die Koppelkinematik
Beschreibung	<p>Dieser Parameter beeinflusst das Verhalten der Koppelkinematik bei programmiertem TCP. Bei der Aufteilung der TCP Bewegung auf die einzelnen Teilkinematiken werden die gesperrten Achsen nicht bewegt.</p> <p>Der Index k beschreibt den zu sperrenden Achsindex. Ein Parameterwert > 0 markiert die Achse mit dem Achsindex k als gesperrt.</p>
Parameter	<code>trafo[i].lock_dof[k]</code> <code>kin_step[i].trafo[j].lock_dof[k]</code> (mehrstufige Transformationen)
Datentyp	BOOLEAN
Datenbereich	0 / 1
Dimension	----
Standardwert	0
Anmerkungen	<p>Eine gesperrte Achse kann weiterhin direkt per Achsbezeichner programmiert werden.</p> <p>Konfigurationsbeispiel:</p> <pre><i>trafo[0].id 210</i> <i>trafo[0].lock_dof[7] 1</i> (Die Achse mit dem Index 7 wird bei der Aufteilung der TCP Bewegung nicht bewegt</pre>

6 Anhang

6.1 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-AXIS-00020	32
P-AXIS-00118	32
P-CHAN-00446	33
P-CHAN-00447	33
P-CHAN-00448	34
P-CHAN-00449	35
P-CHAN-00450	36
P-CHAN-00452	37
P-CHAN-00454	37
P-CHAN-00458	38



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

