



DOCUMENTATION ISG-kernel

Functional description Coupling kinematics

Short description:
FCT-C35

Preface

Legal information

This documentation was produced with utmost care. The products and scope of functions described are under continuous development. We reserve the right to revise and amend the documentation at any time and without prior notice.

No claims may be made for products which have already been delivered if such claims are based on the specifications, figures and descriptions contained in this documentation.

Personnel qualifications

This description is solely intended for skilled technicians who were trained in control, automation and drive systems and who are familiar with the applicable standards, the relevant documentation and the machining application.

It is absolutely vital to refer to this documentation, the instructions below and the explanations to carry out installation and commissioning work. Skilled technicians are under the obligation to use the documentation duly published for every installation and commissioning operation.

Skilled technicians must ensure that the application or use of the products described fulfil all safety requirements including all applicable laws, regulations, provisions and standards.

Further information

Links below (DE)

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

or (EN)

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

contains further information on messages generated in the NC kernel, online help, PLC libraries, tools, etc. in addition to the current documentation.

Disclaimer

It is forbidden to make any changes to the software configuration which are not contained in the options described in this documentation.

Trade marks and patents

ISG®, ISG kernel®, ISG virtuos®, ISG dirigent®, TwinStore® and the associated logos are registered and licensed trade marks of ISG Industrielle Steuerungstechnik GmbH.

The use of other trade marks or logos contained in this documentation by third parties may result in a violation of the rights of the respective trade mark owners.

Copyright

© ISG Industrielle Steuerungstechnik GmbH, Stuttgart, Germany.

No parts of this document may be reproduced, transmitted or exploited in any form without prior consent. Non-compliance may result in liability for damages. All rights reserved with regard to the registration of patents, utility models or industrial designs.

General and safety instructions

Icons used and their meanings

This documentation uses the following icons next to the safety instruction and the associated text. Please read the (safety) instructions carefully and comply with them at all times.

Icons in explanatory text

- Indicates an action.
- ⇒ Indicates an action statement.



⚠ DANGER

Acute danger to life!

If you fail to comply with the safety instruction next to this icon, there is immediate danger to human life and health.



⚠ CAUTION

Personal injury and damage to machines!

If you fail to comply with the safety instruction next to this icon, it may result in personal injury or damage to machines.



Attention

Restriction or error

This icon describes restrictions or warns of errors.



Notice

Tips and other notes

This icon indicates information to assist in general understanding or to provide additional information.



Example

General example

Example that clarifies the text.



Programing Example

NC programming example

Programming example (complete NC program or program sequence) of the described function or NC command.



Release Note

Specific version information

Optional or restricted function. The availability of this function depends on the configuration and the scope of the version.

Table of contents

Preface.....	2
General and safety instructions	3
1 Overview.....	6
2 Description	7
2.1 Applications.....	7
2.2 Prerequisites	8
2.3 Nomenclature	8
2.4 Constraints and limitations	9
3 Configuration	10
3.1 Configuring the partial kinematics	10
3.1.1 Special features of Universal Kinematic ID 91	11
3.2 Configuring groups.....	12
3.2.1 Configuring a workpiece CS	13
3.3 Configuring the kinematic chain of a group.....	15
3.4 Configuring the priority for the solution algorithm.....	17
3.5 Configuring axes	18
3.6 Tool data	20
3.7 Modification in the NC program.....	21
3.8 Configuration examples	22
3.8.1 Robot on an XY linear unit.....	22
3.8.2 Robot on an XZ linear unit at a rotary/swivel table	23
4 Programming	25
4.1 Direct programming of partial kinematics.....	26
4.2 Programming the TCP	27
4.3 Mixed programming	28
4.4 Locking degrees of freedom.....	29
4.5 Available V.G variables	29
5 Parameter	31
5.1 Overview	31
5.2 Description	32
5.2.1 Axis parameters	32
5.2.2 Channel parameters	33
6 Appendix	39
6.1 Suggestions, corrections and the latest documentation.....	39
Keyword index	40

List of figures

Fig. 1:	Kinematic chain with linear axes and robot	7
Fig. 2:	Kinematic chain with X and Z basic offsets	11
Fig. 3:	Robot on an XY linear unit on a rotary/swivel table	13
Fig. 4:	Group design	15
Fig. 5:	Robot on an XY linear unit.....	22
Fig. 6:	Robot on an XY linear unit on a rotary/swivel table	23

1 Overview

Task

Coupling kinematics are a kinematic transformation which permits the design of a larger complex kinematic from several single kinematics. It describes the relationship between the single partial kinematics but does not provide any new machine kinematics.

Possible applications

Partial kinematics

- can be stacked as a kinematic chain;
- can be operated in parallel as a kinematic tree;
- or can be used as a mixture of the two variants.

For example, this can be used to form a kinematic chain from a robot standing on a linear axis to extend the workspace.

Operation in parallel allows a robot to work on a rotary/swivel table to and follow the movements of the table.



Release Note

This function is available as of CNC Build V3.1.3080 or V3.1.3108.



Notice

This function is an additional option requiring a license.

Parameterisation

The coupling kinematic has the **kinematic type 210**. The detailed parameter definition is dependent on the machine layout and is described in detail in the chapter Parameters [► 31].

Programming

Programming can be carried out as a mixed form of TCP programming and/or as the programming of the single kinematics.

The program variants are explained in the chapter Programming [► 25].

Mandatory note on references to other documents

For the sake of clarity, links to other documents and parameters are abbreviated, e.g. [PROG] for the Programming Manual or P-AXIS-00001 for an axis parameter.

For technical reasons, these links only function in the Online Help (HTML5, CHM) but not in pdf files since pdfs do not support cross-linking.

2 Description

Coupling kinematics permit the combination of existing kinematic transformations, providing a generic mechanism to arrange single kinematics to form a large kinematic.

Kinematics which are part of a coupling kinematic are referred to below as partial kinematics.

Partial kinematics can be divided into 2 different types:

1. Serial, such as a kinematic chain.
2. Parallel, such as a kinematic tree.

Every type of a kinematic tree can also consist of a kinematic chain. A serial chain cannot be further distributed and can only consist of kinematics from a kinematic library, with the exception of coupling kinematics.

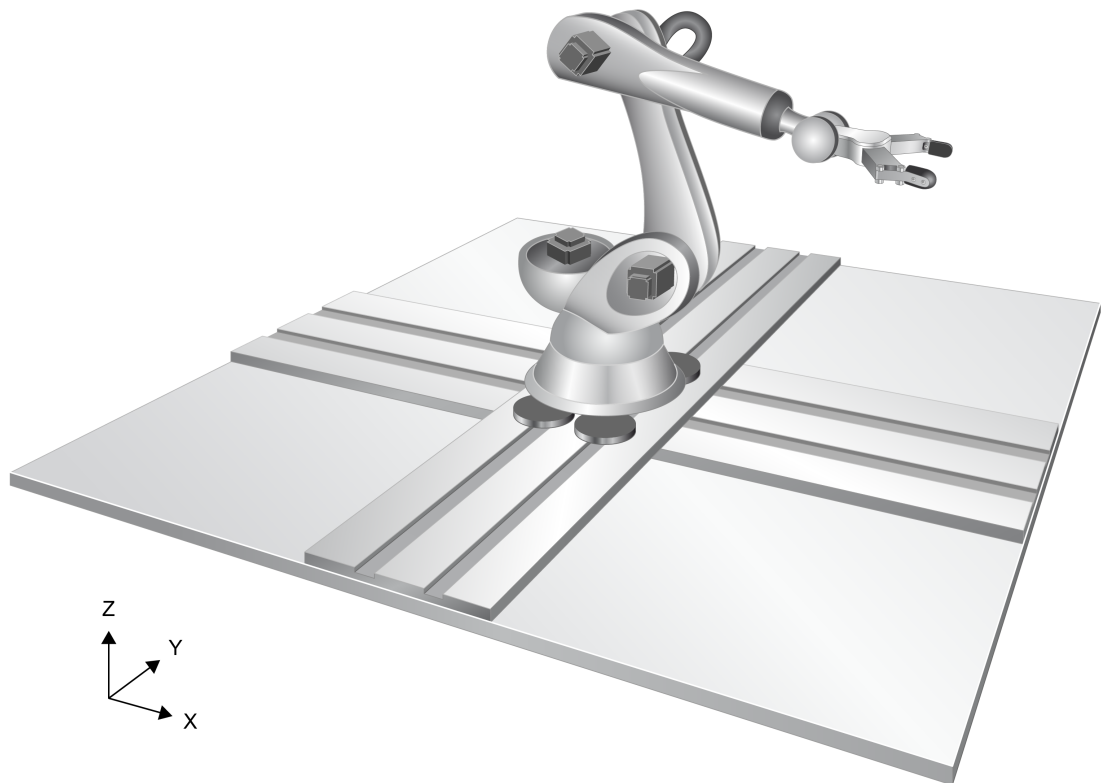


Fig. 1: Kinematic chain with linear axes and robot

2.1 Applications

The sections below describe some of the application options for coupling kinematics:

- Workspace extension, e.g. robot on a linear axis.
- Cooperation between several kinematics, e.g. robot at a rotary/swivel table.
- Changing a highly dynamic machine into load-bearing kinematics, e.g. XYZ machine attached to a robot.

2.2 Prerequisites

The following kinematic types can be used as partial kinematics for coupling kinematics:

Kinematic type	Description
5	4-axis kinematic with crosshead for 4 tools
6	4-axis kinematic with underfloor milling tool
7	5-axis kinematic with manual auxiliary axis (drilling)
8	5-axis kinematic with manual auxiliary axis (sawing)
9	5-axis kinematic (drilling and milling unit)
10	5-axis kinematic (sawing)
45	6-axis articulated robot
96	4-axis palletising robot
91	Universal Kinematics
500-999	TcCOM transformation with orientation (see [McCOM-TRAFO])

2.3 Nomenclature

The table below provides an overview of the terms introduced for coupling kinematics.

Term	Description
Partial kinematic	Kinematic taken from the kinematic library and which is part of a coupling kinematic.
Group	Unit consisting of one or several partial kinematics concatenated in series.
Workpiece CS (group)	Group which holds a workpiece and whose movements are followed by other groups.
TCP	Tool Centre Point; tool tip and end of a kinematic chain. The movement of the TCP results from the superimposed movement of the partial kinematics of a particular group.

2.4 Constraints and limitations

The following constraints apply to the use of coupling kinematics

- A maximum of 3 kinematics can be part of a coupling kinematic.
- A maximum of 3 kinematics can be configured in each group.
- A maximum of 2 groups can be configured.
- The programmed TCP is always assigned to the first group (Index 0).
- Only one programmed TCP can be configured.
- In manual mode (G200), TCP axes and a partial kinematic may not be moved simultaneously if they have the same degree of freedom (e.g. motion in X direction). Otherwise, the warning ID 51065 is output.

Functional constraints of the CNC

The following CNC functions cannot be used when a coupling kinematic is active. They are suppressed and a related error message is output:

- Manual mode with parallel interpolation (G201) (ID 22092)
- Multi-step transformations (ID 22093)
- 2-path programming (ID 22094)
- Homing with TCP axes (ID 22091)
- PTP contouring mode, use of #CONTOUR MODE[PTP ..] (ID 22122)

3 Configuration

The kinematic is configured in several steps which are described in the sections below.

The recommended sequence is:

1. Configure the partial kinematics.
2. Configure the groups.
3. Configure the kinematic chain of each group.
4. Configure the priority for the solution algorithm in which the partial kinematics of a kinematic chain are to travel the programmed TCP movement.
5. Configure the axes.

3.1 Configuring the partial kinematics

The basic configuration of the partial kinematics is carried out as before by means of kinematic parameters.



Attention

For each partial kinematic, a unique ID must be given to the kinematic (P-CHAN-00262) in order to configure it since the coupling kinematic identifies its partial kinematic by this ID. If this is ignored, error ID 21931 is output.

Configuring the basic offsets

The basic offsets of a partial kinematic describe a static Cartesian offset between its MCS and the TCP of the previous kinematic within a kinematic chain (P-CHAN-00446 [► 33]).

The basic offset of the lowest partial kinematic of a kinematic chain (chain[0]) describes the offset between the MCS of this kinematic and the MCS of the coupling kinematic.

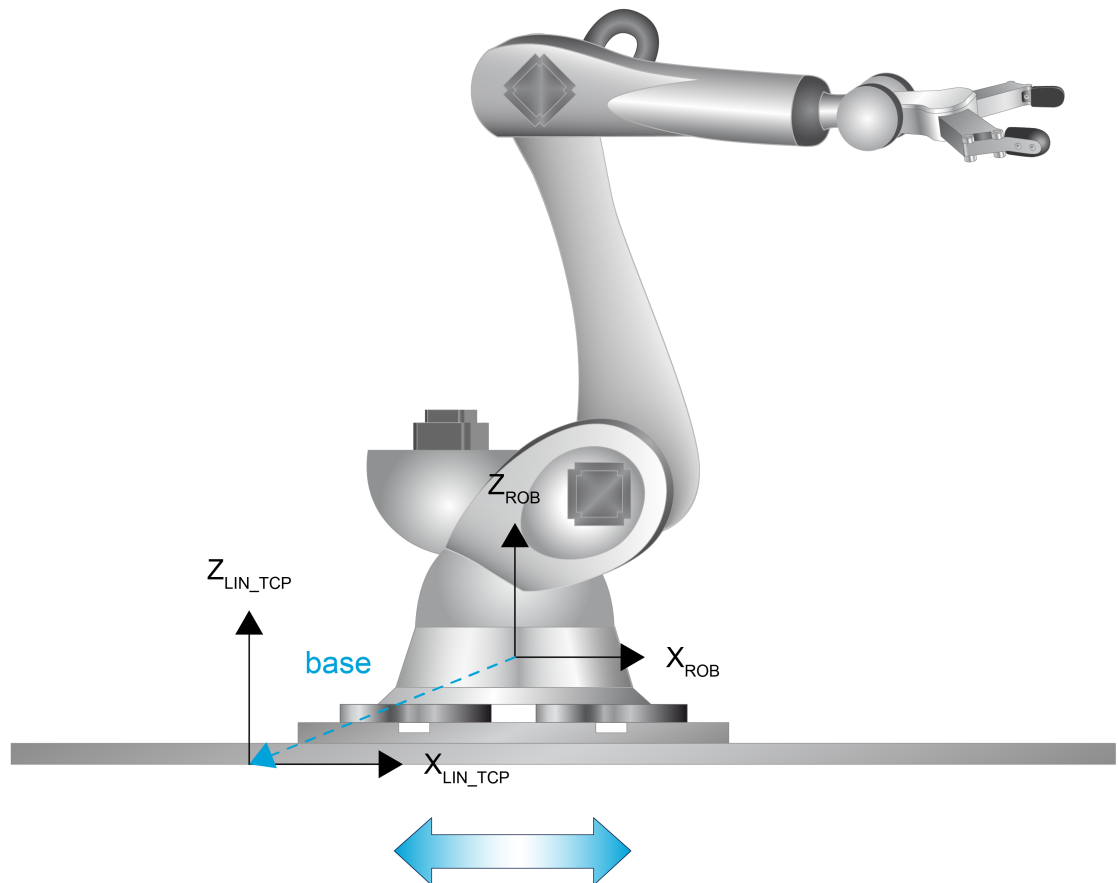


Fig. 2: Kinematic chain with X and Z basic offsets

For example, due to the height of the slide on which the robot is attached, this offset may be in the Z direction.

Properties

- Reference system: MCS partial kinematic.
- Six coordinates can be specified: X, Y, Z, A, B and C offsets.
- The rotation sequence of the angles corresponds to the rotation sequences of each kinematic.

```
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

Special features of Universal Kinematic ID 91

The following special features apply to the Universal Kinematic:

- If it is part of a coupling kinematic, it can be configured with less than three linear axes.
- A configuration with zero linear axes is also possible.
- The parameter zero_orientation (P-CHAN-00285) indicates the direction in which the basic coordinate system of the following kinematic is oriented in the kinematic chain.

3.2 Configuring groups

Every group represents an independent kinematic chain. Therefore, as many groups must be configured in a coupling kinematic as there are kinematic chains. The parameters required to configure a group are as follows:

Channel parameters	Description
P-CHAN-00447 [► 33]	ID of the group (optional)
P-CHAN-00448 [► 34]	The workpiece of the group is held by the group specified here
P-CHAN-00449 [► 35]	Kinematic chain of the group [► 15]
P-CHAN-00450 [► 36]	Sequence for the solution algorithm [► 17]

It is optional to configure a group ID (P-CHAN-00447).

It is also optional to specify a group which holds the workpiece (P-CHAN-00448). It need only be configured if a cooperation between the groups is required, see “Configuration of a workpiece CS [► 13]”.



Attention

If P-CHAN-00448 is configured and the ID specified fails to match any ID in another group, error 292066 is output.

3.2.1

Configuring a workpiece CS

The parameter P-CHAN-00448 [► 34] cooperates one group with another. The group then follows the movements of the group specified in P-CHAN-00448 [► 34]. If the group has a TCP, the PCS shifts the TCP to the effector system of the specified group.

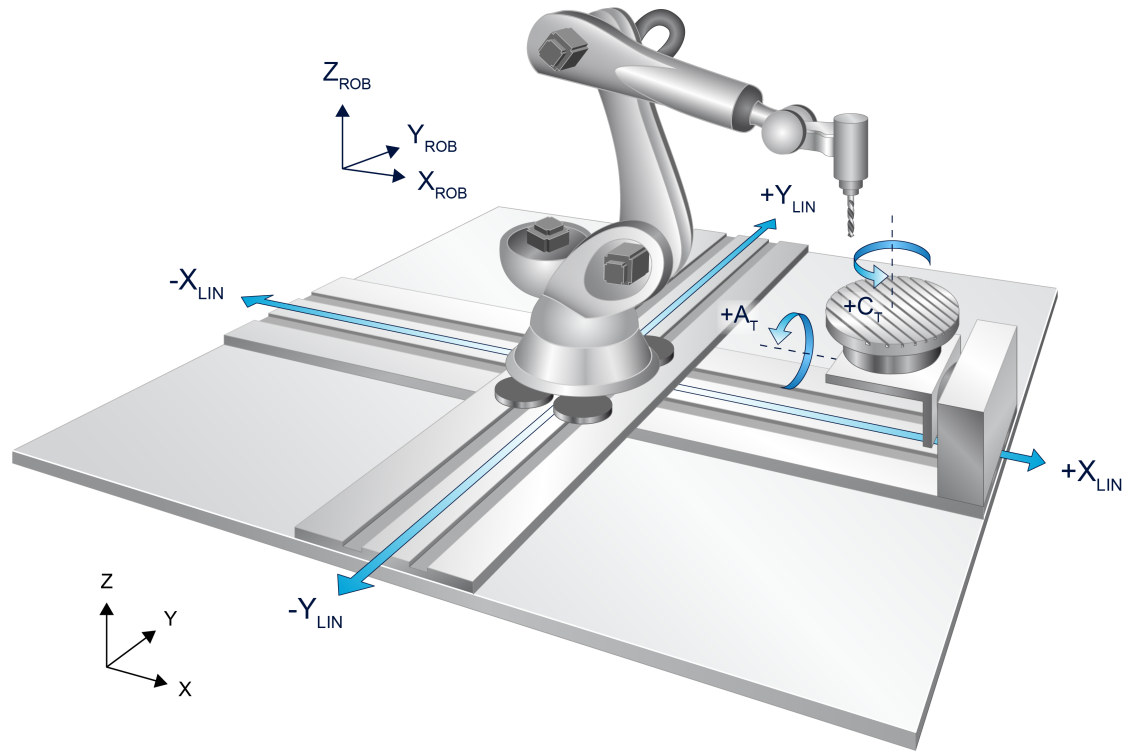


Fig. 3: Robot on an XY linear unit on a rotary/swivel table



Example

A robot works at a rotary/swivel table

The robot is programmed to follow the movement of the rotary/swivel table. This example contains two independent kinematic chains:

- Rotary/swivel table
- Robot

For this reason, two groups must be configured.

To set up a relationship between the two chains, the table group can be defined as workpiece CS of the robot group. The consequence of this is that the robot follows the movement of the table.



Example

The PCS of the group with the active workpiece CS is shifted to the moved workpiece.

In the example below, the PCS of the ID100 group shifts to the workpiece moved by the ID200 group.

```
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
```

The complete parameterisation can be found in the Robot on an XZ linear unit at a rotary/swivel table [► 23].

3.3

Configuring the kinematic chain of a group

Configuration is by means of the parameter chain[i] (P-CHAN-00449 [► 35]).



Attention

Missing kinematic ID

If a kinematic ID specified in P-CHAN-00449 cannot be found in the list of all transformations, error ID 292064 is output.

The kinematic chain is configured according to its physical design:

- chain[0]: Kinematic included by no other kinematic
- chain[i]: Kinematic included by the kinematic chain [i-1]
- Highest index: Kinematic included by no other kinematic

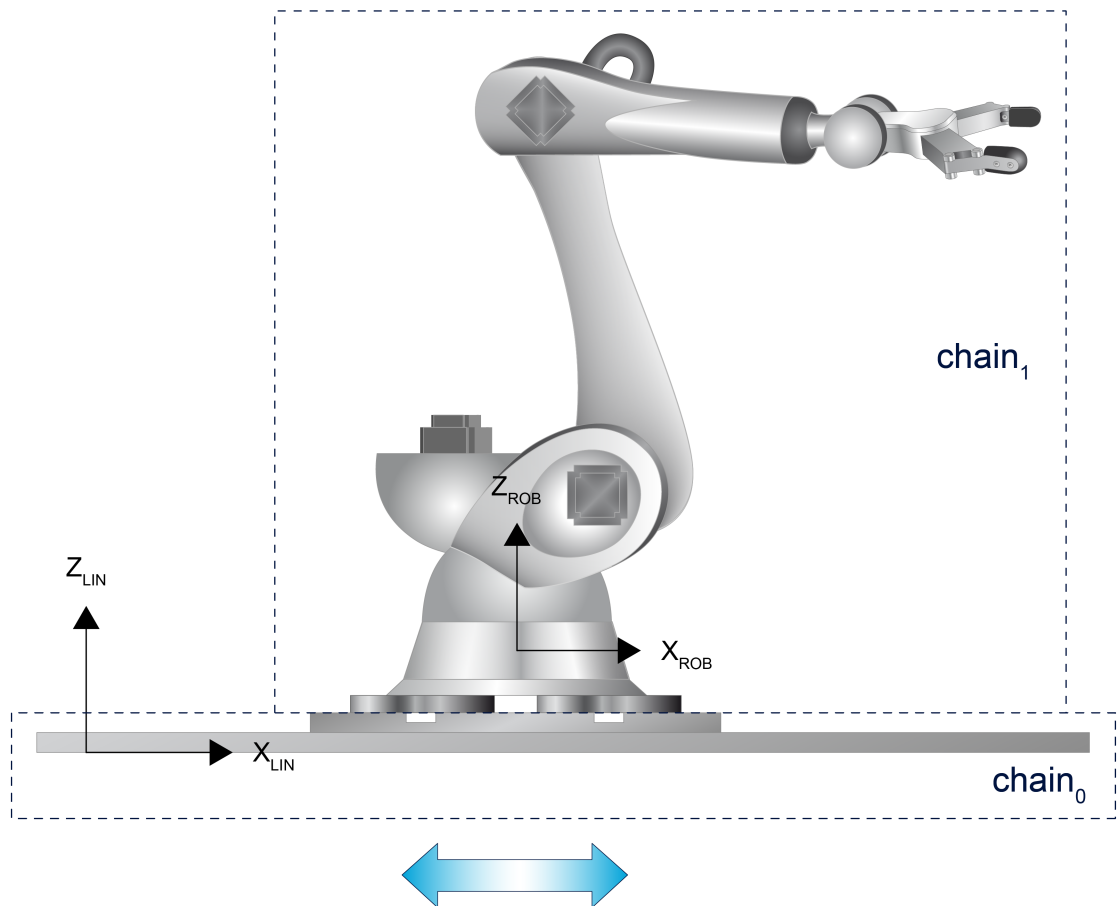


Fig. 4: Group design



Example

A robot stands on a linear axis

```

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 Configuring the priority for the solution algorithm

The partial kinematics of a kinematic chain can be directly programmed by the configured axis identifiers. The TCP can also be programmed. The movement of the TCP is then distributed to the partial kinematics. Each partial kinematic is assigned part of the movement which it can travel due to its Cartesian degrees of freedom.



Attention

Non-executable movements are possible

The boundaries of the workspace of the partial kinematics are not considered. A movement can be assigned to a partial kinematic which cannot execute it for design constraints. The resulting errors are dependent on the partial kinematic itself.

The sequence is configured by means of the parameter `move_prio[i]` (P-CHAN-00450 [► 36]). If the parameter is not configured for a group, the parameter values correspond to those of the kinematic chain (P-CHAN-00449 [► 35]): `move_prio[i] = chain[i]`.



Attention

Incomplete configuration

Either an entry must exist in the priority list for every entry in the kinematic chain or no entry may exist at all in the priority list. This condition applies to every single group. If this is not the case, the error ID 292067 is output when the transformation is selected.



Example

A robot stands on a linear axis

With a programmed TCP, the movement is first assigned to the linear axis. If the programmed motion contains degrees of freedom that the linear axis cannot execute, these motions are transferred to the robot.

```
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    (Linear axis first executes)
                                   (the TCP movement)
trafo[2].group[0].move_prio[1] 1    (Robot executes)
                                   (the rest of the movement)
```

3.5 Configuring axes

The axis configuration of the partial kinematics is described in detail in the subsection Configuring the partial kinematics [► 10].



Attention

Besides the axes which are required for the partial kinematics, six additional simulation axes must be configured for the TCP. They must lie on the axis indices 0 - 5 of the channel.



Notice

Only one TCP is allowed. It must always refer to the first group (Index 0). The kinematics of the other groups must be directly programmed.

The following parameters must be configured for all six TCP axes:

Parameter	Parameter number	Value
kenngr.antr_typ	P-AXIS-00020 [► 32]	4
kenngr.messachse	P-AXIS-00118 [► 32]	0



Notice

Measuring with independent axes

TCP axes latch no measurement values since they are not physical axes. As a result, they need not be configured as measurement axes.

However, a measurement travel can still be executed with the TCP axes when a transformation is active. The Cartesian measurement values of the TCP are then calculated from the measurement values of the individual partial kinematics.

The sequence of the axes in the channel must be arranged according to the scheme below:

1. Starting with group index 0, the axes for the first kinematic in the kinematic chain (chain[0]) are sorted according to the TCP axes in the channel.
2. This is followed by the axes of the next kinematic (chain[1]) etc.
3. The same principle is used for the other groups (Index 1 to n).



Example

Configuring the axes of a 6-axis robot on linear axis

Configuring the axes of a 6-axis robot standing on an X linear axis and working at a rotary/swivel table (2 rotary axes).

```
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
```

The axis sequence can be defined as follows:

- Axis index 0-5: TCP axes
- Axis index 6: Axis of the ID 2 kinematic (X linear axis)
- Axis index 7-12: Axes of the ID 1 kinematic (6-axis robot)
- Axis index 13-14: Axes of the ID 3 kinematic (rotary/swivel table)



Attention

The axis sequence must be maintained according to the scheme described above. Any resulting errors are dependent on which axes are incorrectly assigned to a different kinematic.

3.6

Tool data

In this case, the tool is changed to the partial kinematic with the highest index of the first group.
The kinematic parameters configured in the tool are added to the kinematic parameters of this partial kinematic.

3.7 Modification in the NC program

The following parameters can be configured by V.G.KIN variables in the NC program.

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]

If V.G. variables are programmed when a transformation is active, the new kinematic parameters only become active when the transformation is re-activated.

The command

#KIN DATA [**LOCKDOF** | **UNLOCKDOF**] is provided to change locked degrees of freedom when a transformation is active.

3.8 Configuration examples

3.8.1 Robot on an XY linear unit



Example

Configuration with a group and several partial kinematics

This configuration requires a group with the kinematic types 45 and 91. The linear unit is produced by Universal Kinematic 91 with two axes.

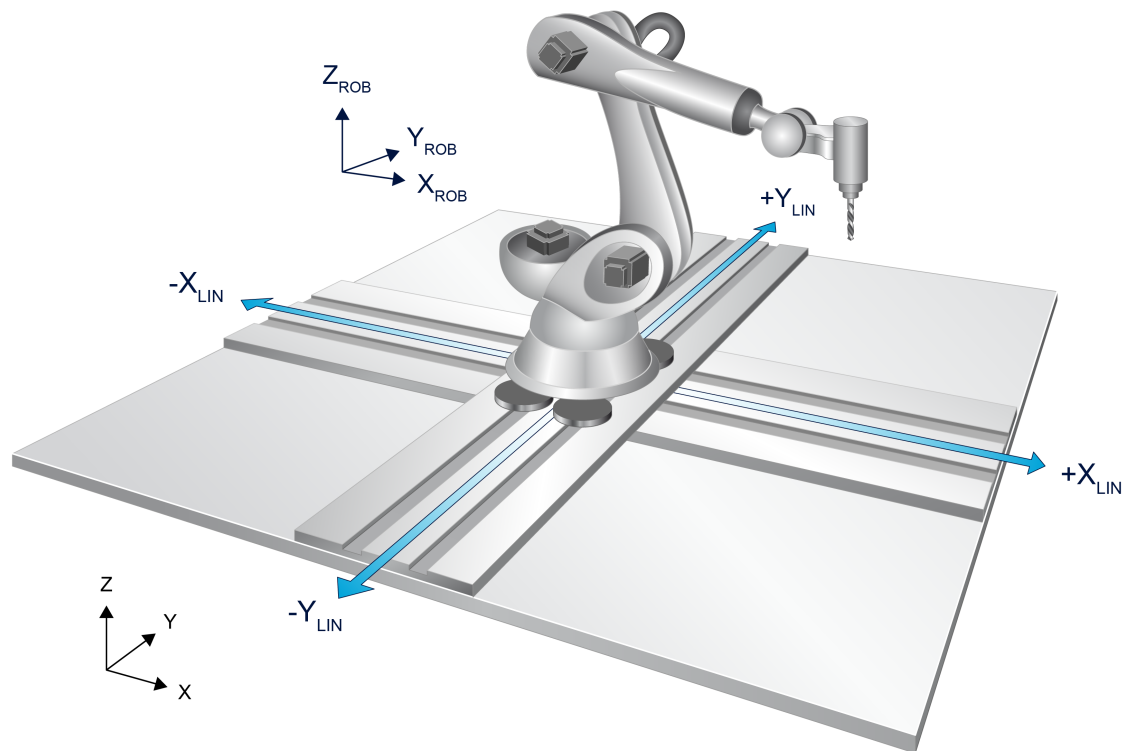


Fig. 5: Robot on an XY linear unit

```
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 Robot on an XZ linear unit at a rotary/swivel table



Example

Configuration with several groups linked to each other

Two groups are required for this configuration. The first group consists of the linear unit and the robot. Configuring this group is analogous to the previous example. The second group consists of the rotary/swivel table which is configured with Universal Kinematic 91 from two rotary axes.

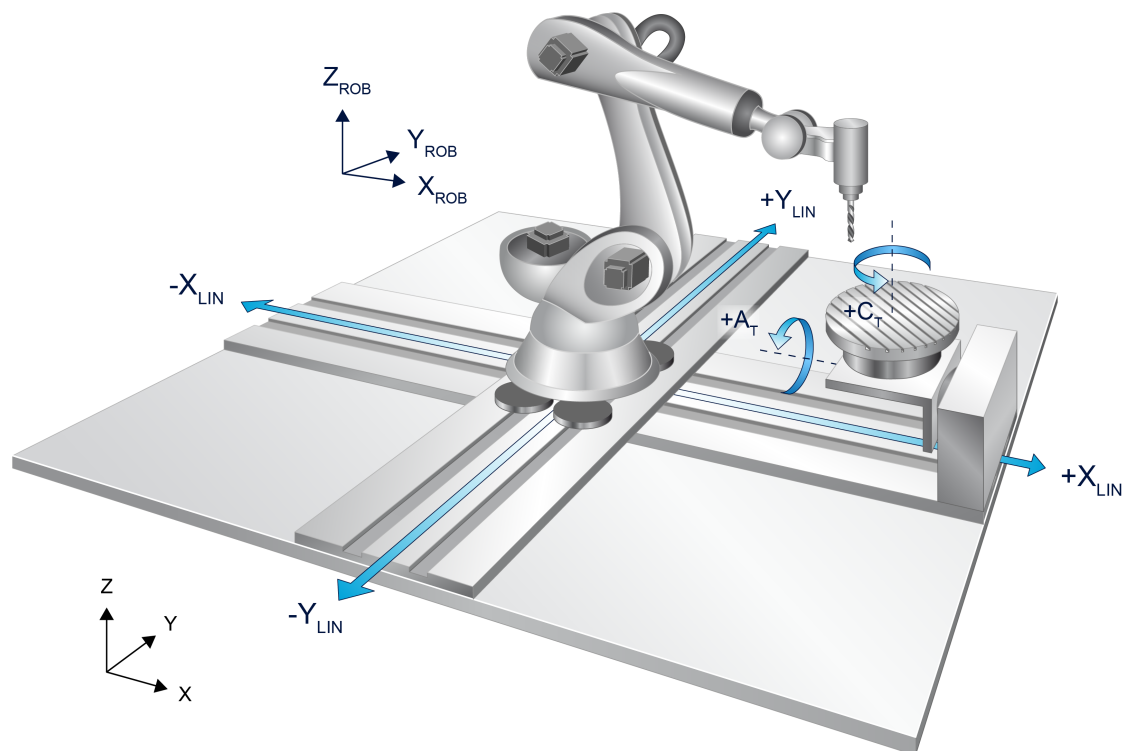


Fig. 6: Robot on an XY linear unit on a rotary/swivel table

Since the configuration comprising XY linear unit and robot corresponds exactly to the previous example [22], the configuration of the elements concerned will not be described here.

```

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 Programming



Notice

The full CNC functionality is only available with TCP programming. Only linear blocks are possible when programming partial kinematics.

All the functions of the NC kernel are only available for programming the TCP and therefore the first group. Only linear blocks can be programmed for all partial kinematics. It is not possible to use partial kinematics for other NC function, e.g. circular interpolation or for machining coordinate systems.

The programming coordinate system (PCS) of the partial kinematics cannot be changed and is located at the machine origin of each coordinate.

In addition to existing methods, the PCS of the TCP can also be shifted by specifying the tool CS (P-CHAN-00448 [▶ 34]) in the moved effector system of a different group.

The coupling kinematics below consists of

- a XY linear unit
- a 6-axis robot mounted on it
- The linear unit and the robot are aligned in the X and Y directions



Example

Configuration example of a robot on an XY linear unit

The configuration below will be used in the programming examples. For reasons of clarity, the partial kinematics and axes are not fully configured here.

```
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
...
```

The axis identifiers are:

- XY linear unit
 - X_LIN, Y_LIN
- Robot
 - 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

Programming RTCP kinematics

Initial situation: The couple kinematic consists of a 5-axis RTCP kinematic and additional partial kinematics.

The orientation angle mode (P-CHAN-00112) must be set so that the rotation sequence is identical to the sequence of rotary axes in the RTCP kinematic. Only the angles that would be programmed for the RTCP kinematic should be programmed in the TCP.



Example

RTCP kinematics – axes to be programmed

Kinematics	Programming mode	TCP axes to be programmed
CA	CAB (3)	A, C
AC	ABC (5)	A, C
BA	CBA_STD (4)	B, C
AB	ABC (5) or CAB (3)	A, B

4.1

Direct programming of partial kinematics

Partial kinematics can only be programmed by linear blocks. However, the TCP must be programmed for extended movements such as circles.



Programing Example

Direct programming of robots and linear axes

```

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

```

At program end, the coordinates are:

	X	Y
Linear unit	500	0
Robot	1000	300
TCP	1500	300

4.2 Programming the TCP



Programing Example

Linear and rotary movement of the TCP

```
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
```

The movement is distributed as follows:

	X	Y	Z	A	B	C
Linear unit	1000	500				
Robot	0	0	50	0	0	10
TCP	1000	500	50	0	0	10

The linear unit can only travel movements in the X and Y directions due to its degrees of freedom. The movement in Z and C directions programmed in N030 are transferred to the robot.

4.3 Mixed programming

Programming TCP coordinates and coordinates of partial kinematics in the same NC command.



Programming Example

Linear and rotary movement of the TCP and fixed programming of the linear unit.

Additional programming of linear axes in block 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
```

The movement is distributed as follows:

	X	Y	Z	A	B	C
Linear unit	800	0				
Robot	200	500	50	0	0	10
TCP	1000	500	50	0	0	10

The robot must travel the difference between the TCP position and the position of the linear axes in X and Y directions.



Programming Example

Circular interpolation in the TCP and fixed programming of individual coordinates

```
N010 #KIN ID [3]
N020 #TRAFO ON
; Approach start position of linear unit and robot
N030 G00 G90 X_LIN=0 Y_LIN=0
N040 G00 G90 X_ROB=500 Y_ROB=0 Z_ROB=1500
; Circular interpolation
N050 G02 G91 F5000 X_TCP=0 Y_TCP=-500 R250 X_LIN=0 Y_ROB=0
N060 #TRAFO OFF
```

The movement is distributed as follows:

	X	Y	Z	A	B	C
Linear unit	0	-500				
Robot	500	0	1500	0	0	10
TCP	500	-500	1500	0	0	10

In block N050, the Y linear axis assumes the movement in the Y direction. Simultaneously, the robot executes the movement in the X direction. The programmed partial kinematics then travel only linear movements.

4.4 Locking degrees of freedom

Programming TCP coordinates and locking degrees of freedom of partial kinematics



Programing Example

Linear movement of the TCP, locking the Y axis of the linear unit

```
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
```

The movement is distributed as follows:

	X	Y	Z	A	B	C
Linear unit	800	0				
Robot	200	500	50	0	0	0
TCP	1000	500	50	0	0	0

Since the Y axis of the linear unit (Y_LIN) in block N030 is excluded from distributing the TCP movement, the robot travels the programmed Y movement of the TCP.

4.5 Available V.G variables

In addition, the following variables listed are provided to access the data of the Universal Kinematic.

V.G.KIN[j].*	Meaning	Data type	Unit for input/output	R/W*
BASE[k]	Basic offset of a kinematic (P-CHAN-00446 [► 33]) <k>:= 0 - 5	Real	[0.1 µm, 10 ⁻⁴ °]	R/W
GROUP[k].NAME	Name of a group (P-CHAN-00447 [► 33]) max. 16 characters <k>:= 0, 1	Integer	-	R/W
GROUP[k].WORK-PIECE_CS	Group bears the workpiece (P-CHAN-00448 [► 34]) max. 16 characters <k>:= 0, 1	Integer	-	R/W
GROUP[k].CHAIN[l]	Definition of the kinematic chain in the group (P-CHAN-00449 [► 35]) max. 16 characters <k>:= 0, 1 <l>:= 0, 1, 2	Integer	-	R/W
GROUP[k].MOVE_PRIO[l]	Definition of the sequence for distributing the TCP movement to partial kinematics((P-CHAN-00450 [► 36]) max. 16 characters <k>:= 0, 1 <l>:= 0, 1, 2	Integer	-	R/W
LOCK_DOF[k]	Locked Cartesian degrees of freedom which cannot be moved by the solution algorithm. (P-CHAN-00458 [► 38]) <k>:= 0 -15 Example: Lock the axis with index 4 V.G.KIN[210].LOCK_DOF[4] = 1	Boolean	-	R/W

5 Parameter

5.1 Overview

Axis parameter

ID	Parameter	Description
P-AXIS-00020	kenngr.antr_typ	Drive type
P-AXIS-00118	kenngr.messachse	Define axis as measurement axis

Channel parameters

ID	Parameter	Description
P-CHAN-00446	base[i]	Basic offset of a kinematic
P-CHAN-00447	id	ID of a group
P-CHAN-00448	workpiece_cs	This group holds the workpiece (only for cooperations)
P-CHAN-00449	chain[i]	Definition of the kinematic chain in the group
P-CHAN-00450	move_prio[i]	Definition of the sequence for distributing the TCP movement to partial kinematics
P-CHAN-00452	default_name_of_kin_step[i]	Name of the default transformation for multi-step transformations
P-CHAN-00454	kinematik_name	Name of the default transformation
P-CHAN-00458	lock_dof[i]	Locked Cartesian degrees of freedom which cannot be moved by the solution algorithm.

5.2 Description

5.2.1 Axis parameters

P-AXIS-00020	Drive type	
Description	This parameter defines the drive type for each axis. A choice of drive types can be made for each axis:	
Parameter	kenngr.antr_typ	
Data type	UNS16	
Data range	1 : Conventional drive interface 2 : SERCOS drive interface 3 : PROFIDRIVE MC drive interface 4 : Drive simulation 5 : Beckhoff Lightbus drive interface 6 : Drive interface +-10V over field bus (terminal) 7 : RT Ethernet drive interface 8 : CANopen drive interface 16 : Virtual axis 32 : CAN bus	
Axis types	T, R, S	
Dimension	T: ----	R,S: ----
Default value	0x0004	
Drive types	----	
Remarks	This entry is not adopted when the axis parameter list is updated. Updates only become effective when the controller is rebooted. Axis parameter block data can be divided into drive type dependent and drive type independent parameters.	

P-AXIS-00118	Define axis as measurement axis	
Description	The parameter must be assigned to TRUE for all axes that participate in a measurement traverse.	
Parameter	kenngr.messachse	
Data type	BOOLEAN	
Data range	0/1	
Axis types	T, R	
Dimension	T: ----	R: ----
Default value	0	
Drive types	----	
Remarks		

5.2.2 Channel parameters

P-CHAN-00446	Definition of the Cartesian basic offsets for stacked kinematics.
Description	The basic offsets are active for stacked kinematics and they describe the Cartesian offsets between the MCS of a kinematic and the TCP of the kinematic stacked below it. A robot is situated on a slide on a linear axis. Due to the presence of the slide there is a 10 cm offset in z direction between the linear axis and the base of the robot. Enter this offset for the robot here: <pre>trafo[0].id 45 trafo[0].base[2] -1000000</pre>
Parameter	trafo[j].base[k] where k=0 ... 5 kin_step[i].trafo[j].base[k] (multistep transformations)
Data type	REAL64
Data range	MIN(REAL64) ... MAX(REAL64)
Dimension	0.1µm for linear offsets, 0.0001° for angles
Default value	0.0
Remarks	The basic offsets are only considered when the coupling kinematic is active.

P-CHAN-00447	Definition of a group ID of a coupling kinematic
Description	A coupling kinematic group includes a number of stacked partial kinematics and is uniquely identifiable by its ID.
Parameter	trafo[j].group[k].id kin_step[i].trafo[j].group[k].id (multi-step transformations)
Data type	UNS16
Data range	0 ... MAX(UNS16)
Dimension	----
Default value	0
Remarks	As of Build V3.1.3080.09 A group ID may not be assigned more than once within a coupling kinematic.

P-CHAN-00448	Definition of a workpiece CS for a group of a couple kinematic.
Description	If a tool CS is specified (group ID of another group of the coupling kinematic), all TCP co-ordinates of this group are interpreted in the TCP system of the group specified as the workpiece CS. This group therefore follows the movements of the workpiece CS group.
Parameter	trafo[j].group[k].workpiece_cs kin_step[i].trafo[j].group[k].workpiece_cs (multi-step transformations)
Data type	UNS16
Data range	0 ... MAX(UNS16)
Dimension	----
Default value	0
Remarks	Configuration example: <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> After the kinematic (ID 210) is selected, group 200 follows the movements of group 100 since it is configured as the tool coordinate system of 200. As of Build V3.1.3080.09

P-CHAN-00449	Definition of a kinematic chain for a group of a coupling kinematic
Description	Description of the sequence in which the partial kinematics within a coupling kinematic are stacked on top of each other.. Index m = 0 points to the lowest kinematic.
Parameter	trafo[j].group[k].chain[m] kin_step[i].trafo[j].group[k].chain[l] (multi-step transformations)
Data type	UNS16
Data range	0 ... MAX(UNS16)
Dimension	----
Default value	0
Remarks	Each kinematic used here must be configured as an autonomous kinematic with a corresponding ID. Configuration example: Robot on a linear axis <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> As of Build V3.1.3080.09

P-CHAN-00450	Priority which the couple kinematic uses to split the TCP movement among the partial kinematics.
Description	The algorithm which distributes the programmed TCP coordinates among the partial kinematics precedes the priority defined here: Beginning with the index $m = 0$ the largest possible part of the movement is to be traversed by this kinematic. The resulting difference to the programmed TCP is handed over to the next kinematic in the list. This step is repeated until the TCP coordinates are reached, or until the last entry within the sequence defined here is reached.
Parameter	<code>trafo[j].group[k].move_prio[m]</code> where $m = 0 \dots \text{Length of the kinematic chain}$ <code>kin_step[i].trafo[j].group[k].move_prio[m]</code> (multi-step transformations)
Data type	UNS16
Data range	0 ... MAX(UNS16)
Dimension	----
Default value	0
Remarks	Each of the names used here must be present within the kinematic chain. Configuration example: Robot on a linear axis <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> In this example, the entire TCP movement is first handed over to the robot. What the robot cannot clear is then adopted by the linear axis. As of Build V3.1.3080.09

P-CHAN-00452	Select default kinematic names for multi-step transformations
Description	The name serves as a unique identifier of the kinematic in case a kinematic ID is configured multiple times. A default kinematic name can be defined for each transformation step. This is valid after the controller is started. The default kinematics must be configured in each of the transformation steps; otherwise an error message is output when the transformations (#TRAFO...) are selected.
Parameter	default_name_of_kin_step[i] where i = 0, 1
Data type	STRING
Data range	Maximum of 16 characters
Dimension	----
Default value	-
Remarks	Parameterisation example: After controller start-up, the transformation with ID2 and the name"test" is valid for the first transformation step (.step[0]). <i>default_id_of_kin_step[0] 2</i> <i>default_name_of_kin_step[0] test</i>

P-CHAN-00454	Name of the default kinematic after start-up
Description	If the kinematic ID is configured multiple times, this parameter defines the unique name of the default kinematic after start-up.
Parameter	kinematik_name
Data type	STRING
Data range	Maximum of 16 characters
Dimension	----
Default value	-
Remarks	Parameterisation example: The valid default kinematic after start-up has the ID 2 with the name "test". <i>kinematik_id 2</i> <i>kinematik_name test</i>

P-CHAN-00458	Lock the Cartesian degrees of freedom of the couple kinematic
Description	This parameter influences the behaviour of the coupling kinematic with a programmed TCP. While the TCP movement is distributed among the individual partial kinematics, the locked axes are not moved. Index k describes the axis index to be locked. A parameter value > 0 marks the axis with axis index k as locked.
Parameter	trafo[i].lock_dof[k] kin_step[i].trafo[j].lock_dof[k] (multistep transformations)
Data type	BOOLEAN
Data range	0 / 1
Dimension	----
Default value	0
Remarks	A locked axis may continue to be programmed directly by its axis identifier. Configuration example: <pre>trafo[0].id 210 trafo[0].lock_dof[7] 1 (The axis with Index 7 is not moved while (TCP movement is distributed</pre>

6 Appendix

6.1 Suggestions, corrections and the latest documentation

Did you find any errors? Do you have any suggestions or constructive criticism? Then please contact us at documentation@isg-stuttgart.de.

The latest documentation is posted in our Online Help (DE/EN):



QR code link: <https://www.isg-stuttgart.de/documentation-kernel/>

The link above forwards you to:

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



Notice

Change options for favourite links in your browser;

Technical changes to the website layout concerning folder paths or a change in the HTML framework and therefore the link structure cannot be excluded.

We recommend you to save the above "QR code link" as your primary favourite link.

PDFs for 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>

Email: documentation@isg-stuttgart.de

Keyword index

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
All rights reserved
www.isg-stuttgart.de
support@isg-stuttgart.de

