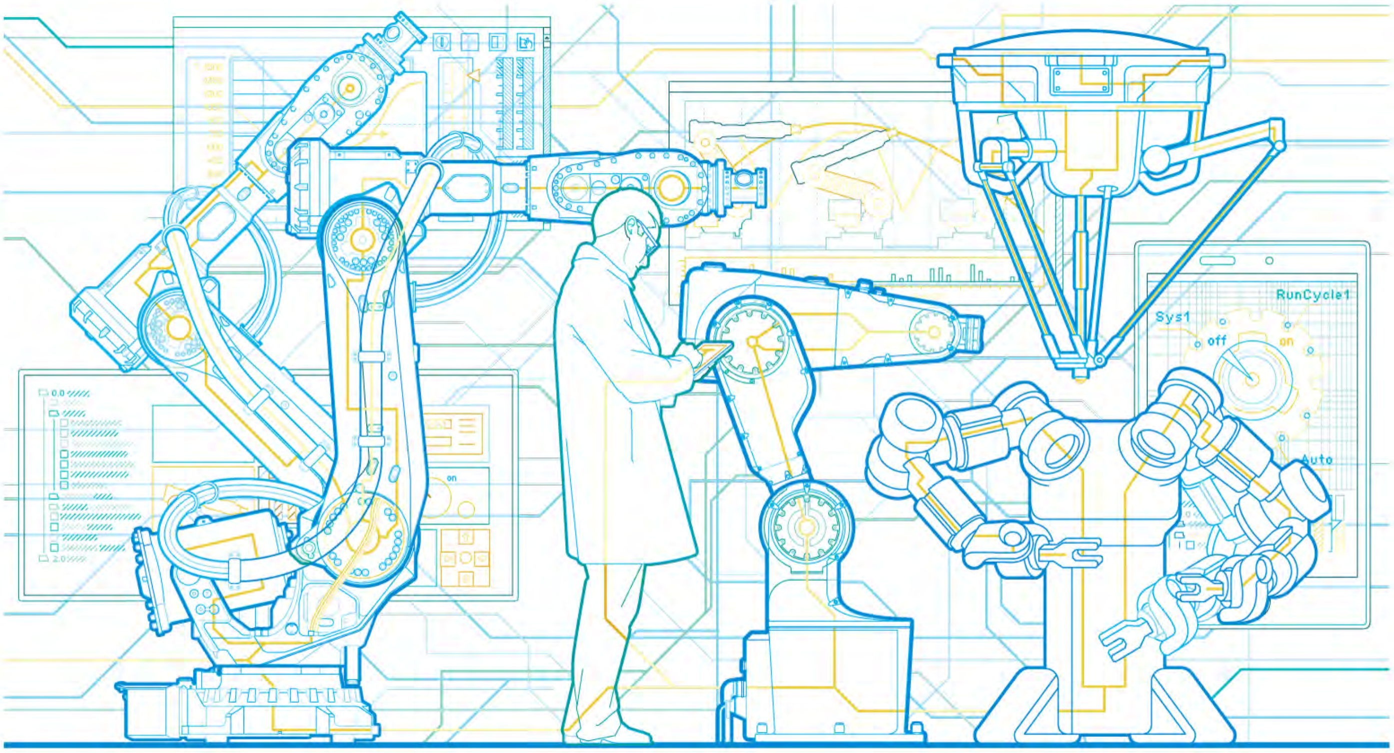


Application manual

Additional axes and stand alone controller

ABB

Trace back information:
Workspace R17-1 version a2
Checked in 2017-03-13
Skribenta version 5.1.011

Application manual

Additional axes and stand alone controller

RobotWare 6.05

Document ID: 3HAC051016-001

Revision: C

The information in this manual is subject to change without notice and should not be construed as a commitment by ABB. ABB assumes no responsibility for any errors that may appear in this manual.

Except as may be expressly stated anywhere in this manual, nothing herein shall be construed as any kind of guarantee or warranty by ABB for losses, damages to persons or property, fitness for a specific purpose or the like.

In no event shall ABB be liable for incidental or consequential damages arising from use of this manual and products described herein.

This manual and parts thereof must not be reproduced or copied without ABB's written permission.

Keep for future reference.

Additional copies of this manual may be obtained from ABB.

Original instructions.

ABB AB, Robotics
Robotics and Motion
Se-721 68 Västerås
Sweden

Table of contents

This page is intentionally left blank

Overview of this manual

About this manual

This manual details the setup of additional axes and non-ABB robots.

Usage

This manual can be used as a brief description of how to install, configure and tune additional axes and non-ABB robots. It also provides information about related system parameters. Detailed information regarding system parameters, RAPID instructions and so on can be found in the respective reference manual.

Who should read this manual?

This manual is primarily intended for advanced users and integrators.

Prerequisites

The reader should...

- be familiar with industrial robots and their terminology
- be familiar with controller configuration and setup
- be familiar with the mechanical and dynamic properties of the controlled mechanism.

References

Reference	Document ID
Application manual - Controller software IRC5	*3HAC050798-001*
Application manual - Servo gun tuning	*3HAC026820-001*
Operating manual - RobotStudio	*3HAC032104-001*
Operating manual - IRC5 with FlexPendant	*3HAC050941-001*
Technical reference manual - RAPID Instructions, Functions and Data types	*3HAC050917-001*
Technical reference manual - System parameters	*3HAC050948-001*
Product manual - IRC5	*3HAC047136-001*
Product manual - Motor Units and Gear Units	*3HAC040148-001*
Product specification - Controller IRC5 with FlexPendant	*3HAC041344-001*
Product specification - Motor Units and Gear Units	*3HAC040147-001*

Revisions

Revision	Description
-	Released with RobotWare 6.0.
A	Released with RobotWare 6.01. Replaced picture in section *Commutate the motor on page 103*, just to increase clarity. Added a note in the section *Defining brake relays on page 80*.

Continues on next page

Revision	Description
B	Released with RobotWare 6.02. Updated the path to the template files, see *Template files on page 22*. Notch filter removed.
C	Released with RobotWare 6.05 The allowed values are updated for parameters: *K Soft Max Factor*, *K Soft Min Factor* and *Kp/Kv Ratio Factor*. Minor corrections.

Product documentation, IRC5

Categories for user documentation from ABB Robotics

The user documentation from ABB Robotics is divided into a number of categories. This listing is based on the type of information in the documents, regardless of whether the products are standard or optional.

All documents listed can be ordered from ABB on a DVD. The documents listed are valid for IRC5 robot systems.

Product manuals

Manipulators, controllers, DressPack/SpotPack, and most other hardware is delivered with a **Product manual** that generally contains:

- Safety information.
- Installation and commissioning (descriptions of mechanical installation or electrical connections).
- Maintenance (descriptions of all required preventive maintenance procedures including intervals and expected life time of parts).
- Repair (descriptions of all recommended repair procedures including spare parts).
- Calibration.
- Decommissioning.
- Reference information (safety standards, unit conversions, screw joints, lists of tools).
- Spare parts list with exploded views (or references to separate spare parts lists).
- Circuit diagrams (or references to circuit diagrams).

Technical reference manuals

The technical reference manuals describe reference information for robotics products.

- *Technical reference manual - Lubrication in gearboxes*: Description of types and volumes of lubrication for the manipulator gearboxes.
- *Technical reference manual - RAPID overview*: An overview of the RAPID programming language.
- *Technical reference manual - RAPID Instructions, Functions and Data types*: Description and syntax for all RAPID instructions, functions, and data types.
- *Technical reference manual - RAPID kernel*: A formal description of the RAPID programming language.
- *Technical reference manual - System parameters*: Description of system parameters and configuration workflows.

Continues on next page

Application manuals

Specific applications (for example software or hardware options) are described in **Application manuals**. An application manual can describe one or several applications.

An application manual generally contains information about:

- The purpose of the application (what it does and when it is useful).
- What is included (for example cables, I/O boards, RAPID instructions, system parameters, DVD with PC software).
- How to install included or required hardware.
- How to use the application.
- Examples of how to use the application.

Operating manuals

The operating manuals describe hands-on handling of the products. The manuals are aimed at those having first-hand operational contact with the product, that is production cell operators, programmers, and trouble shooters.

The group of manuals includes (among others):

- *Operating manual - Emergency safety information*
- *Operating manual - General safety information*
- *Operating manual - Getting started, IRC5 and RobotStudio*
- *Operating manual - IRC5 Integrator's guide*
- *Operating manual - IRC5 with FlexPendant*
- *Operating manual - RobotStudio*
- *Operating manual - Trouble shooting IRC5*

Safety

Safety of personnel

A robot is heavy and extremely powerful regardless of its speed. A pause or long stop in movement can be followed by a fast hazardous movement. Even if a pattern of movement is predicted, a change in operation can be triggered by an external signal resulting in an unexpected movement.

Therefore, it is important that all safety regulations are followed when entering safeguarded space.

Safety regulations

Before beginning work with the robot, make sure you are familiar with the safety regulations described in the manual *Operating manual - General safety information*.

This page is intentionally left blank

1 Introduction

1.1 Overview

Purpose

The additional axes option is used when the robot controller needs to control additional axes besides the robot axes. These axes are synchronized and, if desired, coordinated with the movement of the robot, which results in high speed and high accuracy.

Stand alone controller is an ABB controller delivered without an ABB robot. The purpose is to use it to control non-ABB equipment.

When the controller is used in a robot system with external axes or a non-ABB manipulator, the system requires configuration and tuning as detailed in this manual. This manual can also be useful when such a system needs to be upgraded.

As external axes and non-ABB robots consume more power the drive system needs a more powerful transformer, rectifier and capacitor. In addition, suitable drive units must be installed in the controller. The hardware setup must also be configured with software to make the system functional.

Basic approach

This is the basic approach for the setup of additional axes or a stand alone controller.

- Installation
- Configuration
- Tuning

For a detailed description of how this is done, see the respective section.

For more information on the hardware components see *Hardware on page 165*.

⚠ **WARNING**

The manual mode peripheral speed of the external axis must be restricted to 250mm/s for personal safety reasons. The speed is supervised at three different levels, which means that three system parameters need to be set up. For more information see *Limit peripheral speed of external axis on page 65*.

1.2 Definitions

Robot

A robot is a mechanical unit with a tool center point (TCP). A robot can be programmed both in Cartesian coordinates (x, y and z) of the TCP and in tool orientation.

Non-MultiMove system

A non-MultiMove system can have

- only one motion task
- only one robot
- up to 6 additional axes (which can be grouped in an arbitrary number of mechanical units)
- up to 12 axes in total (located in one or two drive modules)

Tip

In a non-MultiMove system, semi-independent programming of individual mechanical units or axes can be achieved through the option *Independent Axes*. However, MultiMove is normally preferred when independent programming is desired.

MultiMove system

A MultiMove system can have

- up to 6 motion tasks (each task has the same limitations as in a non-MultiMove system)
- up to 4 robots
- up to 4 drive modules (i.e. up to 36 axes including the robot axes)

Additional axes

The robot controller can control additional axes besides the robot axes. These mechanical units can be used for Non-MultiMove and MultiMove systems alike. They can be jogged and coordinated with the movements of the robot. The system may have a single additional axis, e.g. a motor, or a set of additional axes such as a two axis positioner.

Stand alone controller

Stand alone controller means an ABB controller delivered without an ABB robot. The stand alone controller can be used to control non-ABB equipment, usually TCP robots. It can be used for Non-MultiMove and MultiMove systems alike. MultiMove makes it possible to configure and run multiple mechanical units on the same drive module.

1.3 General guidelines and limitations

Use integer gear ratio

The transmission gear ratio between motor and arm of a continuously rotating axis shall be an integer in order not to cause calibration problems when updating revolution counters.

When the revolution counter is updated, the number of motor revolutions is reset. In order for the zero position of the motor to coincide with the zero position of the arm, independent of number of revolutions on the arm side, the gear ratio needs to be an integer (not a decimal number).

Example: Gear ratio = 1:81 (not 1:81.73).

This problem will only be visible when updating revolution counters with the arm side rotated n turns from the original zero position. I.e. an axis with mechanical stops will not have this problem.

This page is intentionally left blank

2 Getting started

2.1 Get started with additional axes, servo guns and non-ABB robots

Overview

This section describes the steps to get started with:

- additional axes
- a servo gun
- non-ABB robots

Step by step

	Action	See
1	Mount the hardware, such as motor unit, connection box cables and connectors.	
2	For additional axes and servo guns there are various template files available depending on the setup of the hardware. If the user does not already have a specific template file, see information on what file to use.	*Template files on page 22*.
3	For a non-ABB robot find the Kinematic model to be used.	*Kinematic models on page 28*.
4	Install the robotware software and create a system using RobotStudio.	*Creating a stand alone controller system on page 63*.
5	Download the system to the robot controller.	
6	Use RobotStudio or the FlexPendant for a basic configuration of system parameters.	*Limit peripheral speed of external axis on page 65*. *Minimal configuration of general additional axes on page 67* or *Minimal configuration of servo gun on page 69* or *Minimal configuration of non-ABB robots on page 72*.
7	Check if any advanced setting needs to be done.	*Advanced settings on page 77*.
8	When configuration is done the system needs to be fine calibrated and tuned.	*Commutate the motor on page 103*. *Tuning on page 107*.

This page is intentionally left blank

3 Installation

3.1 Additional axes and servo guns

3.1.1 Standard additional axis

Overview

Normally all necessary configuration parameters regarding drive unit, rectifiers and transformers are pre-loaded at ABB, and do not need to be re-installed. For more information on how to add options to the system using the Installation Manager, see *Operating manual - RobotStudio*.

Peripheral equipment

If the supplier of track motion or other peripheral equipment, supplies configuration files. These files should be used instead of the standard files. For more information, see the documentation provided by the supplier.

3.1.2 Template files

Overview

This section details the template files for respective hardware. Normally you only need to change the motor data in these files. For more information on how to change these files, see *Operating manual - RobotStudio*.

The template files are located in the following directory in the RobotWare installation: *...\RobotPackages\RobotWare_RPK_<version>\utility\AdditionalAxis*.

Note

Navigate to the RobotWare installation folder from the RobotStudio **Add-Ins** tab, by right-clicking on the installed RobotWare version in the **Add-Ins** browser and selecting **Open Package Folder**.

Motors

There are template files used to connect the motors to the drive system and measurement system.

Listed below are files for motors connected to drive module 1. They are located in: *...\utility\AdditionalAxis\General\DM1*.

Similar template files exist for drive modules 2-4. These files are adjusted for additional axes on the same drive unit as a robot.

File name	Measurement link	Board position	Measurement node
M7L1B1_DM1.cfg	1	1	7
M7L1B2_DM1.cfg	1	2	7
M7L2B1_DM1.cfg	2	1	1
M8L2B1_DM1.cfg	2	1	2
M9L2B1_DM1.cfg	2	1	3

There are also common template files for a general purpose. These files exist for drive module 1-4 (but are rarely used for drive module 1 since axes 1-6 for drive module 1 are usually used by the robot). Listed below are these files for drive module 2. They are located in: *...\utility\AdditionalAxis\IRC_U\DM2*.

File name	Measurement link	Board position	Measurement node
M1_DM2	1	1	1
M2_DM2	1	1	2
M3_DM2	1	1	3
M4_DM2	1	1	4
M5_DM2	1	1	5
M6_DM2	1	1	6

Template files for defining general kinematics can be found in: *...\utility\AdditionalAxis\GeneralKinematics\DM1*.

Continues on next page

These files exist for drive module 1-4. The measurement link, board position and measurement node are all configured as for a regular robot. The manipulator XZB(X) is chosen as it is composed of two linear and one rotating axis thus highlighting the import settings described in section *Defining parameters for general kinematics on page 91*.

Listed below are the configuration files for drive module 1. Primarily the "_11"-files should be used. The other files exist to support the case when the XZB(X) manipulator is connected together with for example another XZB(X) manipulator on the same drive module.

Load all four "_11"-files (or all "_12"-files if it is the second XZB(X), "_13" for the third and "_14" for the fourth) and then restart the controller.

File name
EXT_XZB(X)_TEMPLATE_11.cfg
INT_XZB(X)_TEMPLATE_11.cfg
SEC_XZB(X)_TEMPLATE_11.cfg
UNCALIB_11.cfg

Servo Gun / Track motion

The template files for Servo Gun and Track motion are all prepared for drive module 1-4. The files contain default data for Servo Gun and Track motion. Motor data etc. for selected motor must be changed. Listed below are the template files for drive module 1.

Servo Gun template files located in: *...\utility\AdditionalAxis\ServoGun*.

File name	Measurement link	Board position	Measurement node
M7L1B1S_DM1.cfg	1	1	7
M7L1B2S_DM1.cfg	1	2	7
M8L2B1S_DM1.cfg	2	1	2

Track motion template files located in: *...\utility\AdditionalAxis\Track*.

File name	Measurement link	Board position	Measurement node
M7L1B1T_DM1.cfg	1	1	7
M7L1B2T_DM1.cfg	1	2	7
M8L2B1T_DM1.cfg	2	1	2

Recommended combinations

The following combination of configuration files for motor 7, 8 and 9 are the recommended combinations in one drive module.

Motor 7	Motor 8	Motor 9
M7L1B1_DM1.cfg	M8L2B1_DM1.cfg	M9L2B1_DM1.cfg
M7L1B2_DM1.cfg	M8L2B1_DM1.cfg	M9L2B1_DM1.cfg
M7L2B1_DM1.cfg	M8L2B1_DM1.cfg	M9L2B1_DM1.cfg

Continues on next page

Note

See examples in *Serial Measurement Link examples on page 174*.

3.1.3 Serial measurement system configuration

Overview

The following section details how to configure the measurement link.

Measurement Channel

The Measurement Channel parameters can easily be changed via RobotStudio or the FlexPendant. Select the configuration topic *Motion* and the type *Measurement Channel*. Another alternative is to edit the parameters in the file `MOC.cfg` and load this file to the controller. For information about how to load a cfg file, see *Operating manual - RobotStudio*.

	Action	Info/Illustration
1	Select the serial measurement link by changing the value of the parameter *Measurement Link*.	selectable values: 1 or 2
2	Select the SMB placement by changing the value of the parameter *Board Position*.	selectable values: 1 or 2
3	Select the measurement node by changing the value of the parameter *Measurement Node*.	selectable values: 1 to 7

Note

Each node (1 to 7) must not be used more than once on each serial measurement link.

3.2 Non ABB robots

3.2.1 Introduction

Overview

This section details how to create and install a stand alone controller system, i.e. a system to be used with non-ABB robots. The basic steps to do this are as follows:

- Find the correct drive unit configuration.
- Find the appropriate kinematic model.
- Install RobotWare and the stand alone controller software on your PC.
- Create a stand alone controller system with the selected kinematic model.
- Download the system to the robot controller.

This section also details how to modify and distribute a stand alone package for easy installation and startup at a customer.

3.2.2 Drive module for non-ABB robots

Drive unit configuration

The table shows the different drive units available for non-ABB robots.

No of axes	Corresponding robot	Drive units
6	IRB 140, 1410, 1600	MDU-430A
4	IRB 260, 360	MDU-430A
6	IRB 2400, 2600, 4400, 4600, 66xx, 7600	MDU-790A
4	IRB 460, 660, 760 IRB 4400, 66xx (with only 4 active drives)	MDU-790A

For definitions of drive units and power stages see *Drive units on page 171*.

3.2.3 Kinematic models

3.2.3.1 Introduction

Overview

This section describes the different built-in kinematic models available in the controller. It serves as a guideline for choosing the appropriate model for the current robot system.

Model groups

The table below describes the different groups of kinematic models.

Notation:	Indicates:
Single Axes	one axis
Area	three to six axes
Linear	two to five axes
TopLoader	four to six axes
Doppin Feeder	two or three axes

Model notation

The specific kinematic models within a model group are designated with a combination of capital letters. The table below details the meaning of this notation.

Notation:	Indicates:
X, Y, Z	linear motion
A, B, C, D, E, F	rotational movement
A(X)	rotational movement around X-axis
A(Y)	rotational movement around Y-axis
A(Z)	rotational movement around Z-axis

Note

The base frame is orientated so that the linear motions are parallel to the directions of the base frame axes X, Y and Z.

Related information

Useful information:

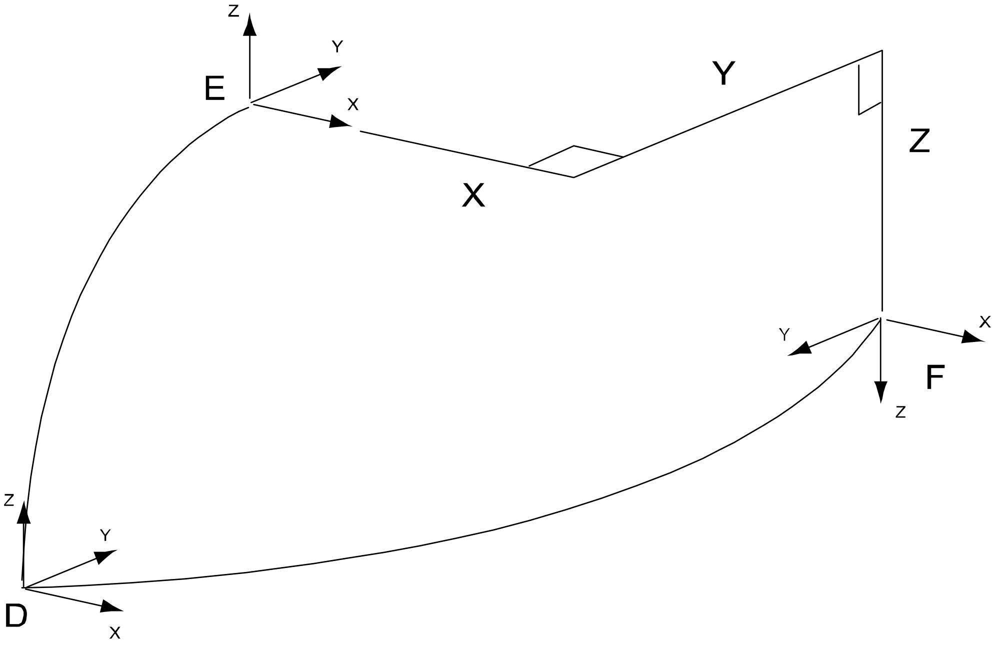
- Read about base and world coordinates in *Operating manual - IRC5 with FlexPendant*, section *Jogging*.

3.2.3.2 Kinematic model XYZ

Description

The kinematic model is based on an area gantry concept, with three linear motions and no rotations.

Illustration



xx0300000614

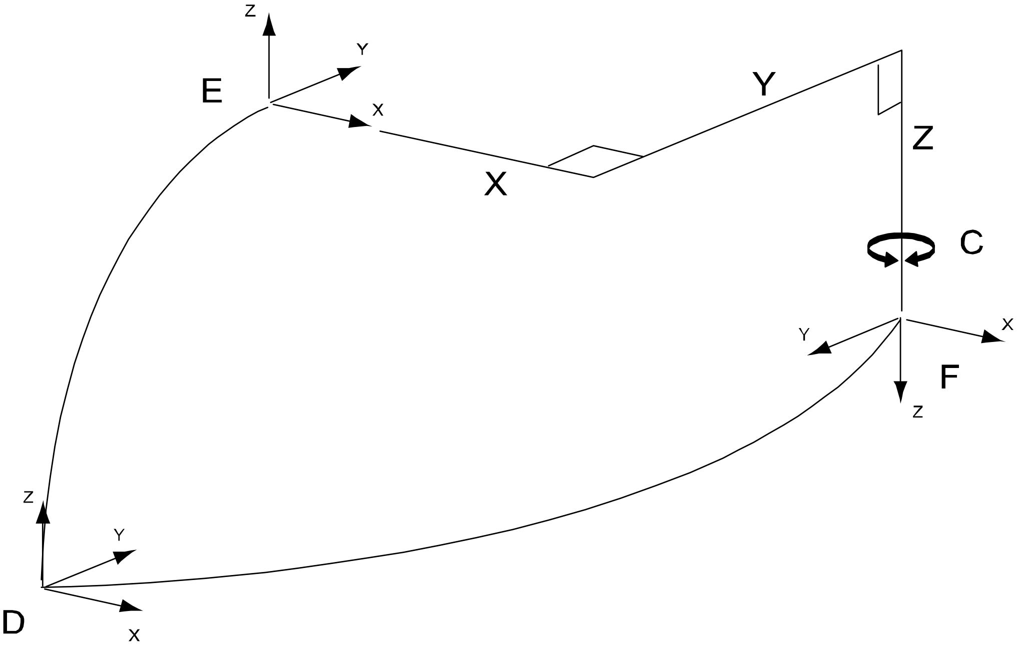
D	World frame
E	Base frame
F	Tool frame
X	X-linear motion
Y	Y-linear motion
Z	Z-linear motion

3.2.3.3 Kinematic model XYZC(Z)

Description

The kinematic model is based on an area gantry concept, with three linear motions and one rotation.

Illustration



xx0300000616

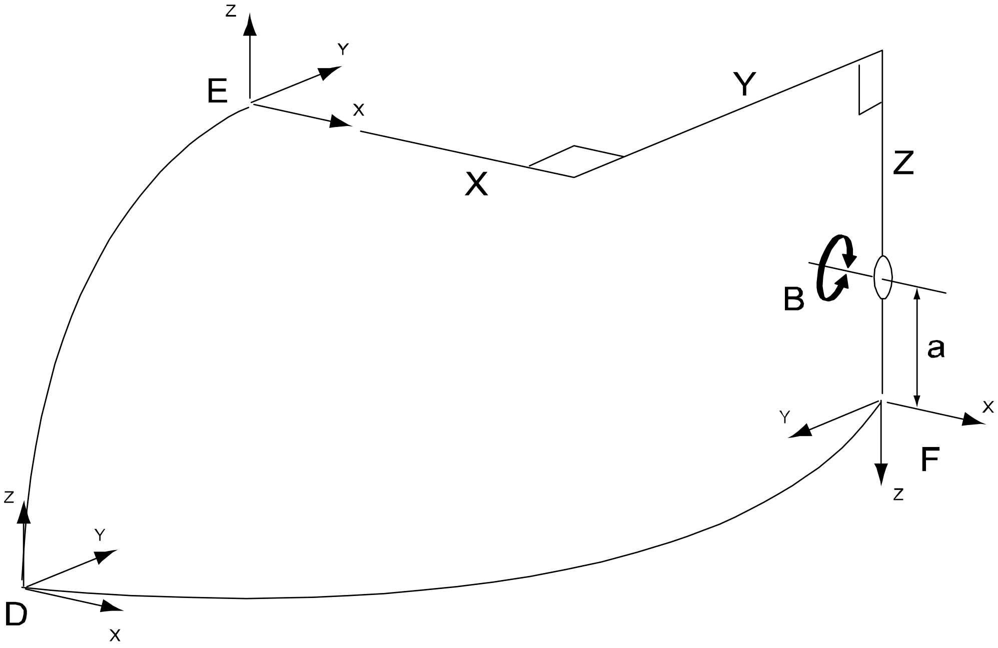
D	World frame
E	Base frame
F	Tool frame
X	X-linear motion
Y	Y-linear motion
Z	Z-linear motion
C	C rotating around Z axis in base frame

3.2.3.4 Kinematic model XYZB(X)

Description

The kinematic model is based on an area gantry concept, with three linear motions and one rotation.

Illustration



xx0300000617

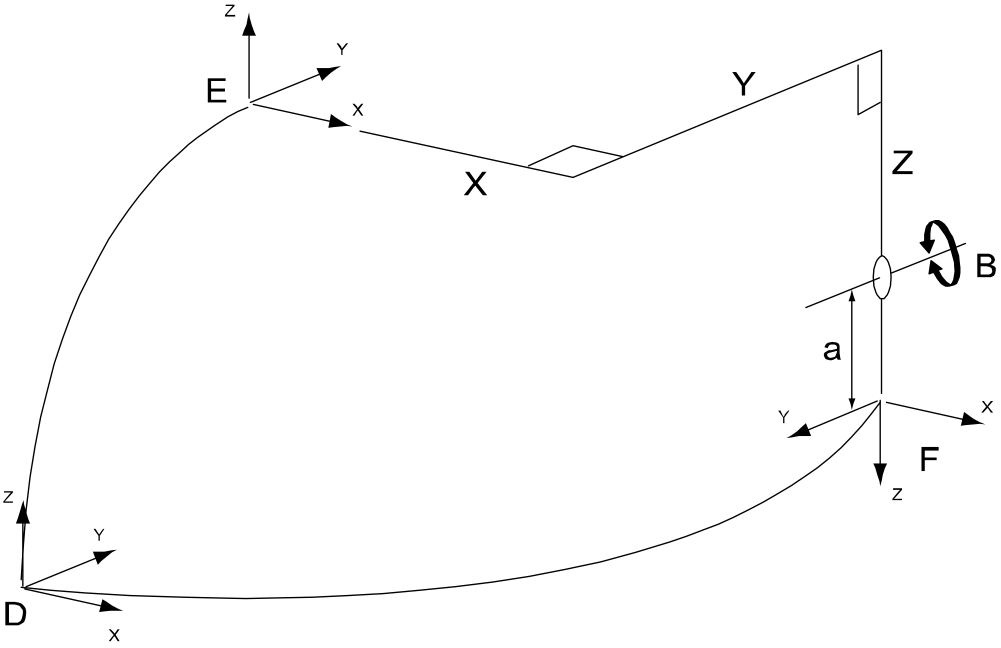
D	World frame
E	Base frame
F	Tool frame
X	X-linear motion
Y	Y-linear motion
Z	Z-linear motion
B	B rotating around X axis in base frame
a	*offset_z* of arm "robx_6”

3.2.3.5 Kinematic model XYZB(Y)

Description

The kinematic model is based on an area gantry concept, with three linear motions and one rotation.

Illustration



xx0300000618

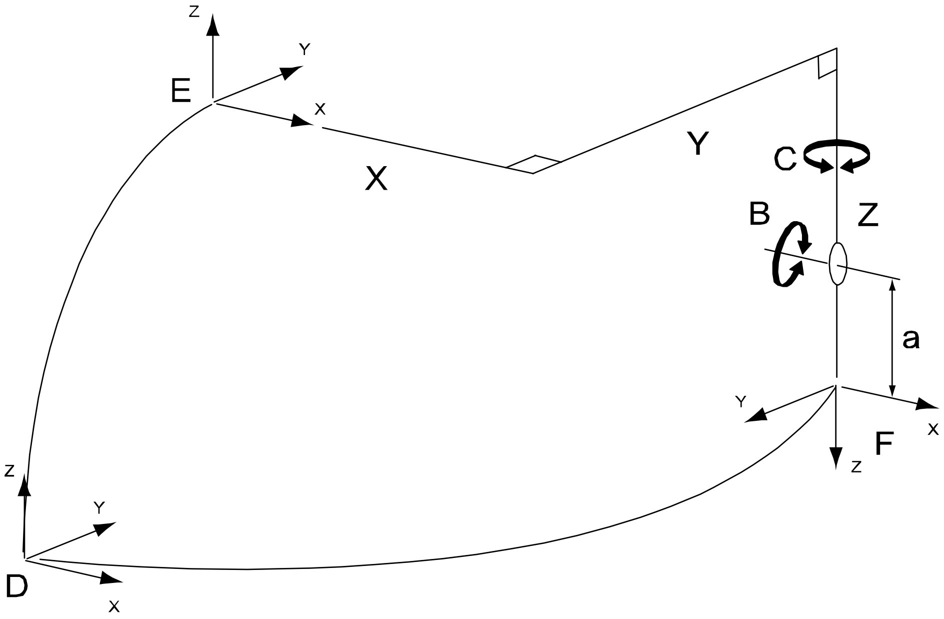
D	World frame
E	Base frame
F	Tool frame
X	X-linear motion
Y	Y-linear motion
Z	Z-linear motion
B	B rotating around Y axis in base frame
a	*offset_z* of arm “robx_6”

3.2.3.6 Kinematic model XYZC(Z)B(X)

Description

The kinematic model is based on an area gantry concept, with three linear motions and two rotations.

Illustration



xx0500002122

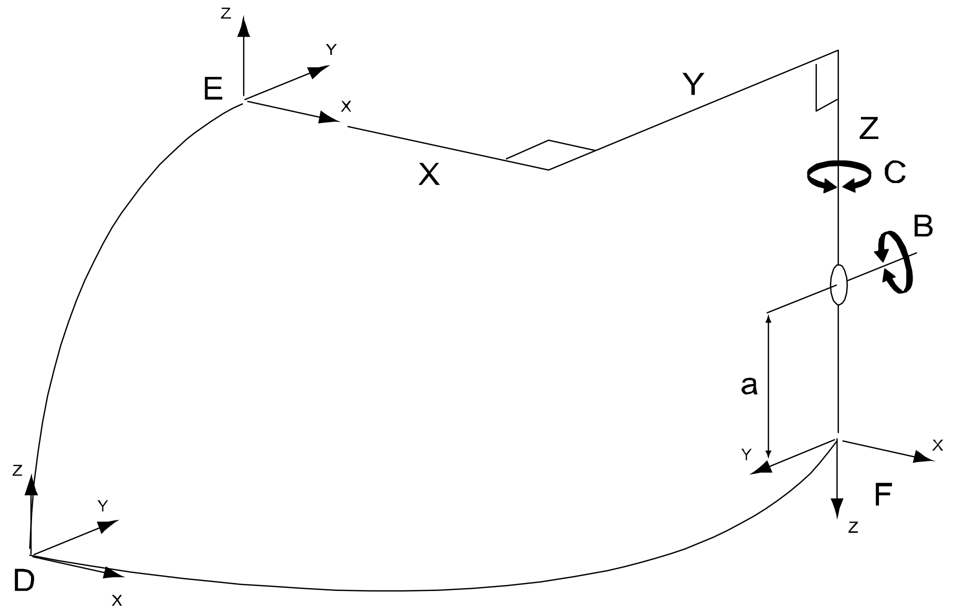
D	World frame
E	Base frame
F	Tool frame
X	X-linear motion
Y	Y-linear motion
Z	Z-linear motion
C	C rotating around Z axis in base frame
B	B rotating around X axis in base frame
a	*offset_z* of arm “robx_6”

3.2.3.7 Kinematic model XYZC(Z)B(Y)

Description

The kinematic model is based on an area gantry concept, with three linear motions and two rotations.

Illustration



xx0500002123

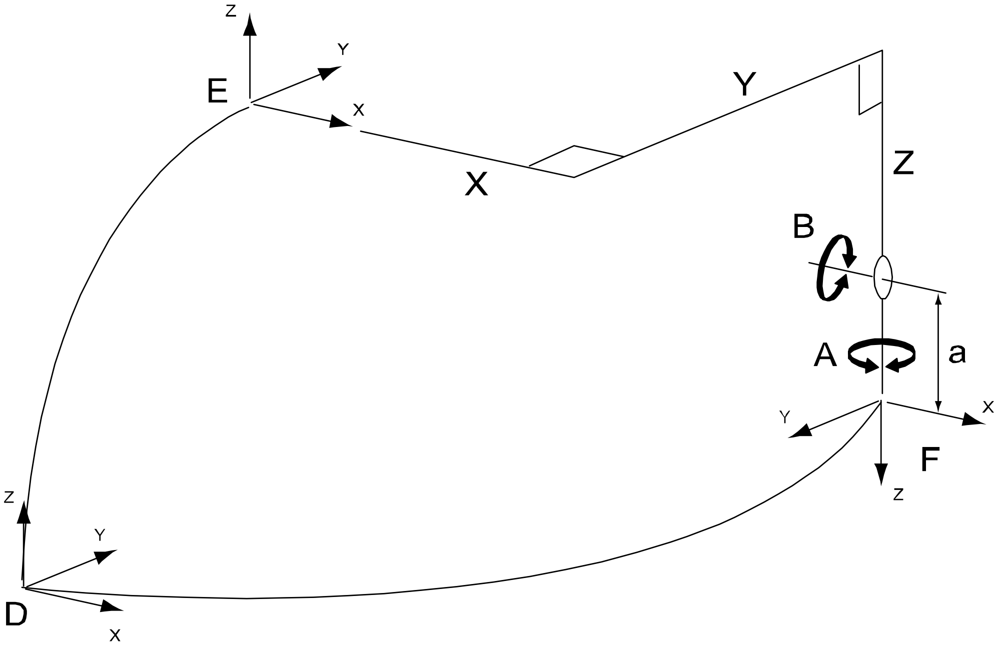
D	World Frame
E	Base Frame
F	Tool Frame
X	X-linear motion
Y	Y-linear motion
Z	Z-linear motion
C	C rotating around Z axis in base frame
B	B rotating around Y axis in base frame
a	*offset_z* of arm "robx_6"

3.2.3.8 Kinematic model XYZB(X)A(Z)

Description

The kinematic model is based on an area gantry concept, with three linear motions and two rotations.

Illustration



xx0300000619

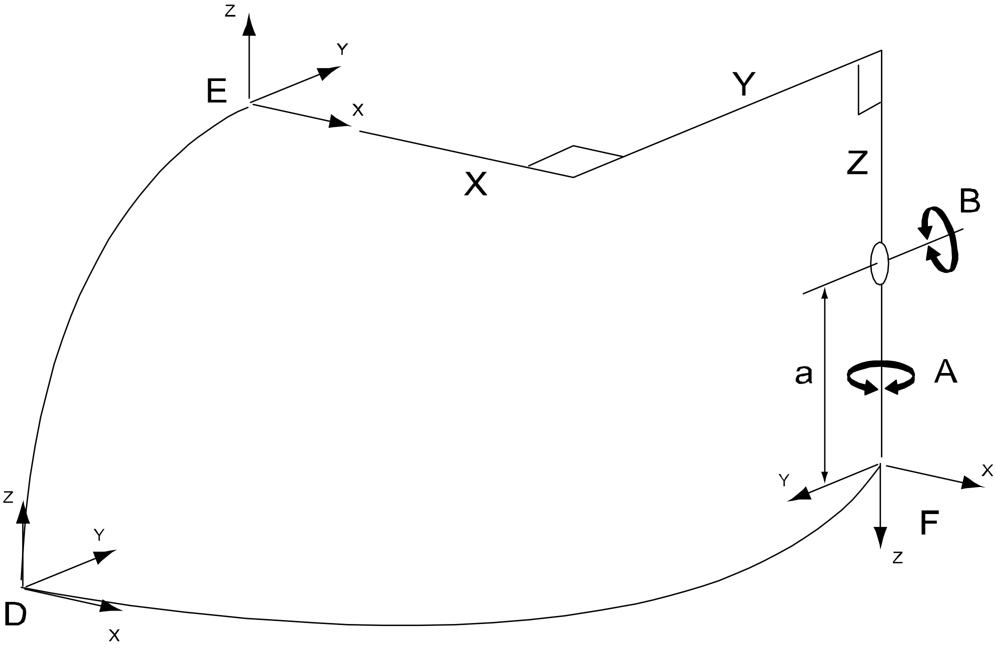
D	World frame
E	Base frame
F	Tool frame
X	X-linear motion
Y	Y-linear motion
Z	Z-linear motion
B	B rotating around X axis in base frame
A	A rotating around Z axis in base frame if B is zero
a	*offset_z* of arm “robx_6”

3.2.3.9 Kinematic model XYZB(Y)A(Z)

Description

The kinematic model is based on an area gantry concept, with three linear motions and two rotations.

Illustration



xx0300000620

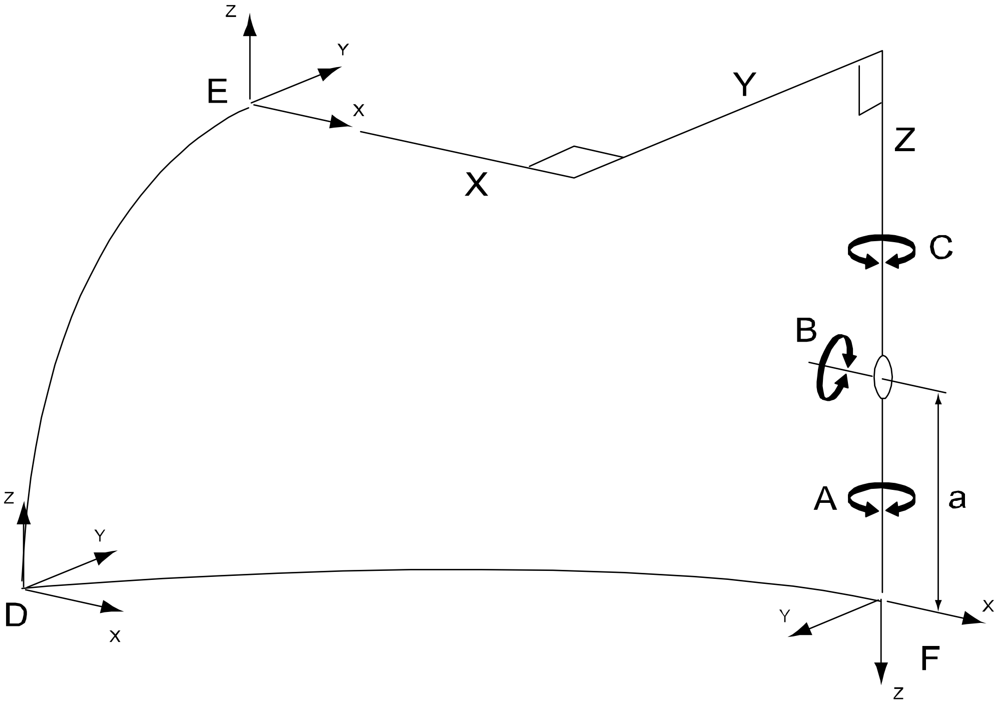
D	World frame
E	Base frame
F	Tool frame
X	X-linear motion
Y	Y-linear motion
Z	Z-linear motion
B	B rotating around Y axis in base frame
A	A rotating around Z axis in base frame if B is zero
a	*offset_z* of arm “robx_6”

3.2.3.10 Kinematic model XYZC(Z)B(X)A(Z)

Description

The kinematic model is based on an area gantry concept, with three linear motions and three rotations.

Illustration



xx0300000621

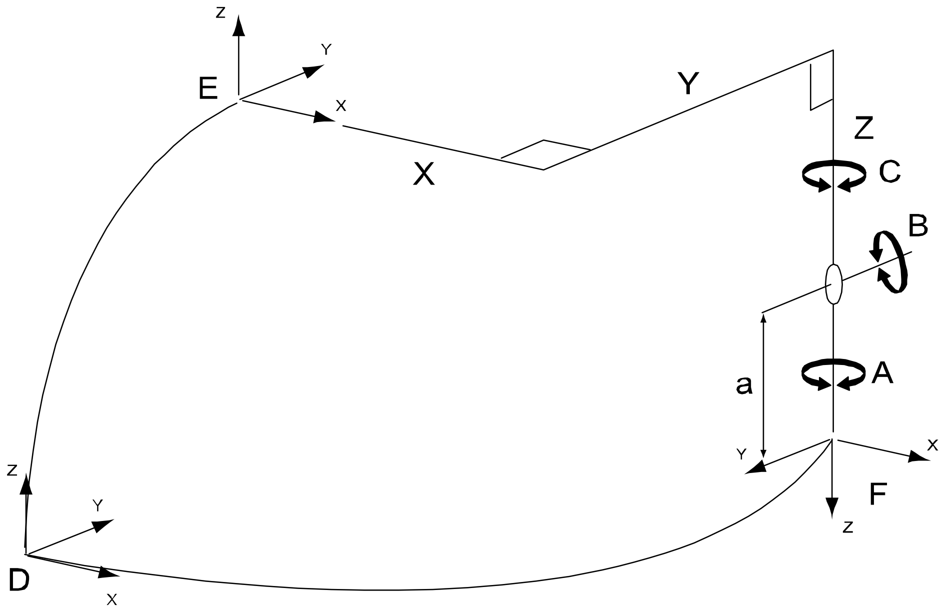
D	World frame
E	Base frame
F	Tool frame
X	X-linear motion
Y	Y-linear motion
Z	Z-linear motion
C	C rotating around Z axis in base frame
B	B rotating around X axis in base frame when C is zero
A	A rotating around Z axis in base frame if B is zero
a	*offset_z* of arm “robx_6”

3.2.3.11 Kinematic model XYZC(Z)B(Y)A(Z)

Description

The kinematic model is based on an area gantry concept, with three linear motions and three rotations.

Illustration



xx0500002211

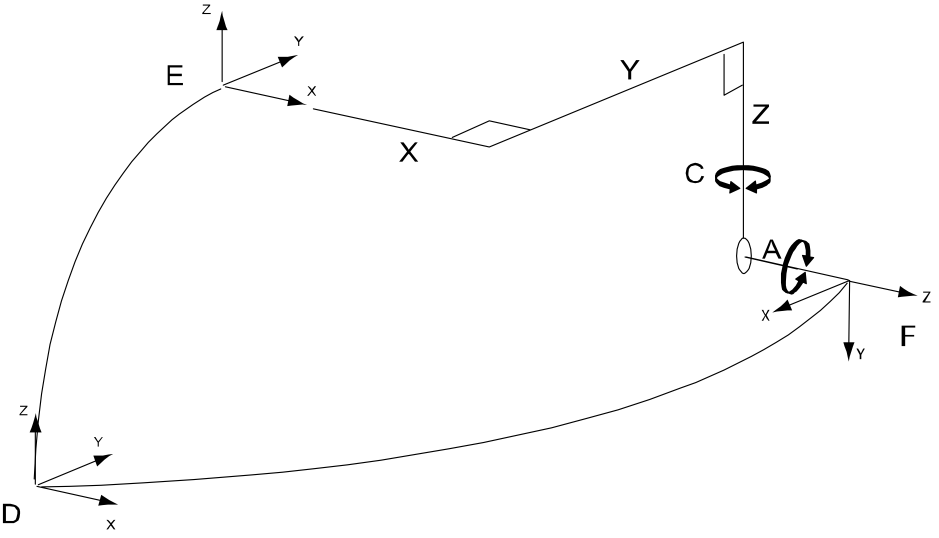
D	World frame
E	Base frame
F	Tool frame
X	X-linear motion
Y	Y-linear motion
Z	Z-linear motion
C	C rotating around Z axis in base frame
B	B rotating around Y axis in base frame when C is zero
A	A rotating around Z axis in base frame if B is zero
a	offset _z of arm ”robx_6”

3.2.3.12 Kinematic model XYZC(Z)A(X)

Description

The kinematic model is based on an area gantry concept, with three linear motions and two rotations.

Illustration



xx0500002202

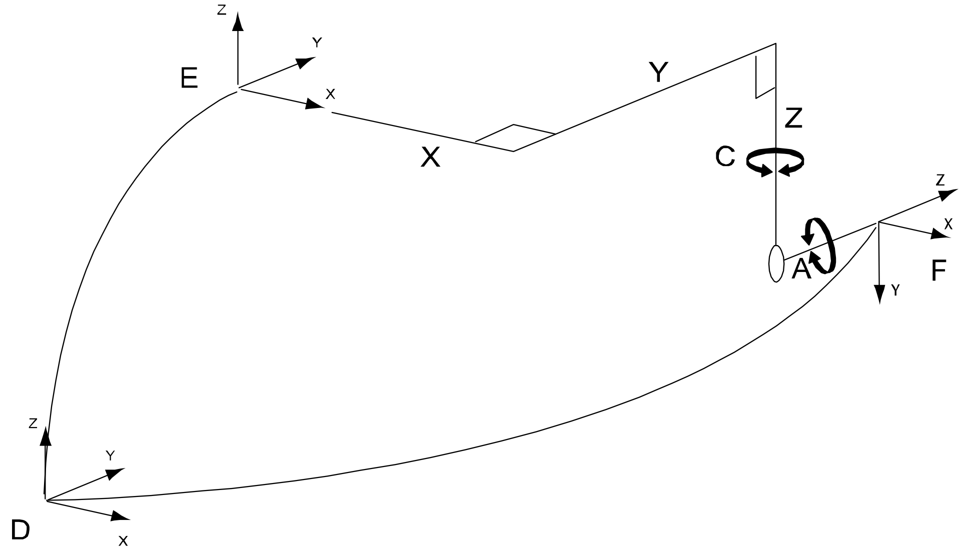
D	World frame
E	Base frame
F	Tool frame
X	X-linear motion
Y	Y-linear motion
Z	Z-linear motion
C	C rotating around Z axis in base frame
A	A rotating around X axis in base frame

3.2.3.13 Kinematic model XYZC(Z)A(Y)

Description

The kinematic model is based on an area gantry concept, with three linear motions and two rotations

Illustration



xx0500002203

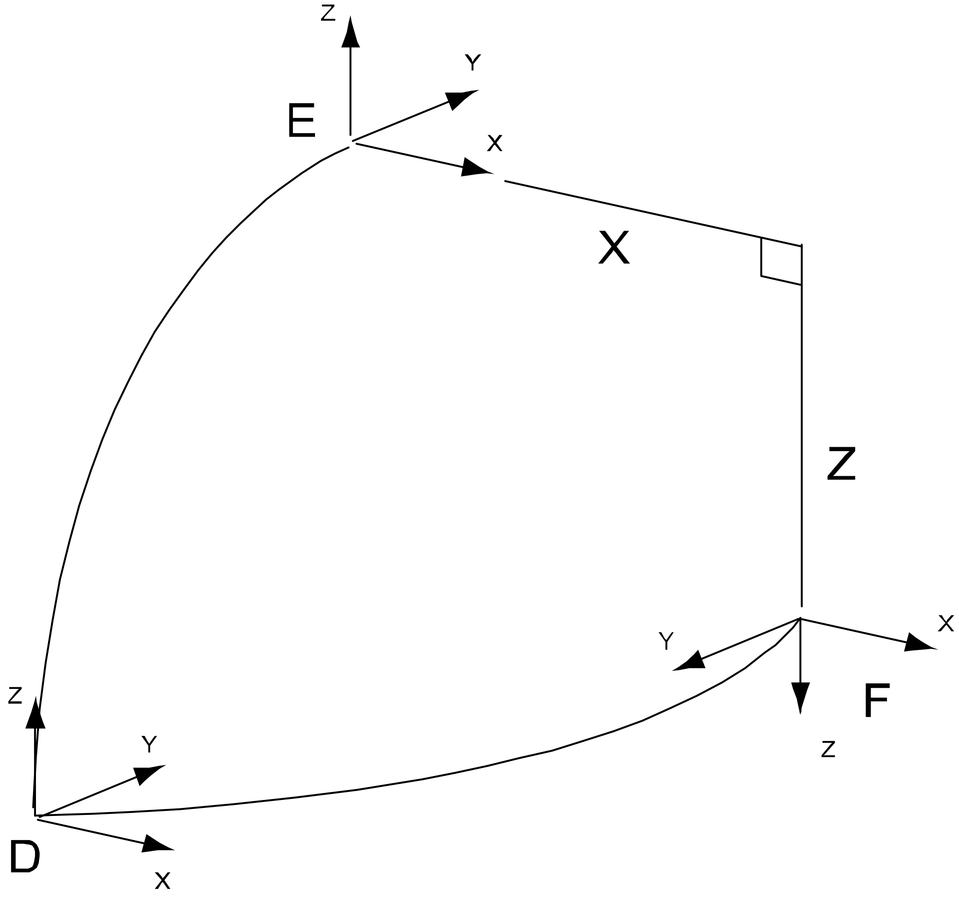
D	World frame
E	Base frame
F	Tool frame
X	X-linear motion
Y	Y-linear motion
Z	Z-linear motion
C	C rotating around Z axis in base frame
A	A rotating around Y axis in base frame

3.2.3.14 Kinematic model XZ

Description

The kinematic model is based on a linear gantry concept, with two linear motions.

Illustration



xx0500002110

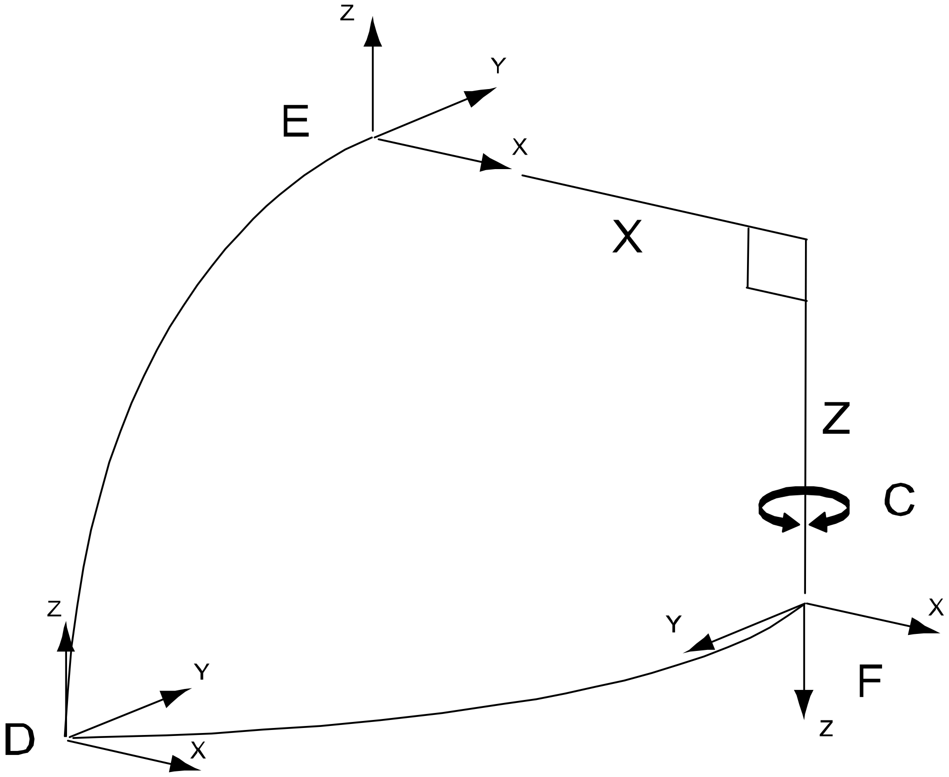
D	World Frame
E	Base Frame
F	Tool Frame
X	X-linear motion
Z	Z-linear motion

3.2.3.15 Kinematic model XZC(Z)

Description

The kinematic model is based on a linear gantry concept, with two linear motions and one rotation.

Illustration



xx0500002115

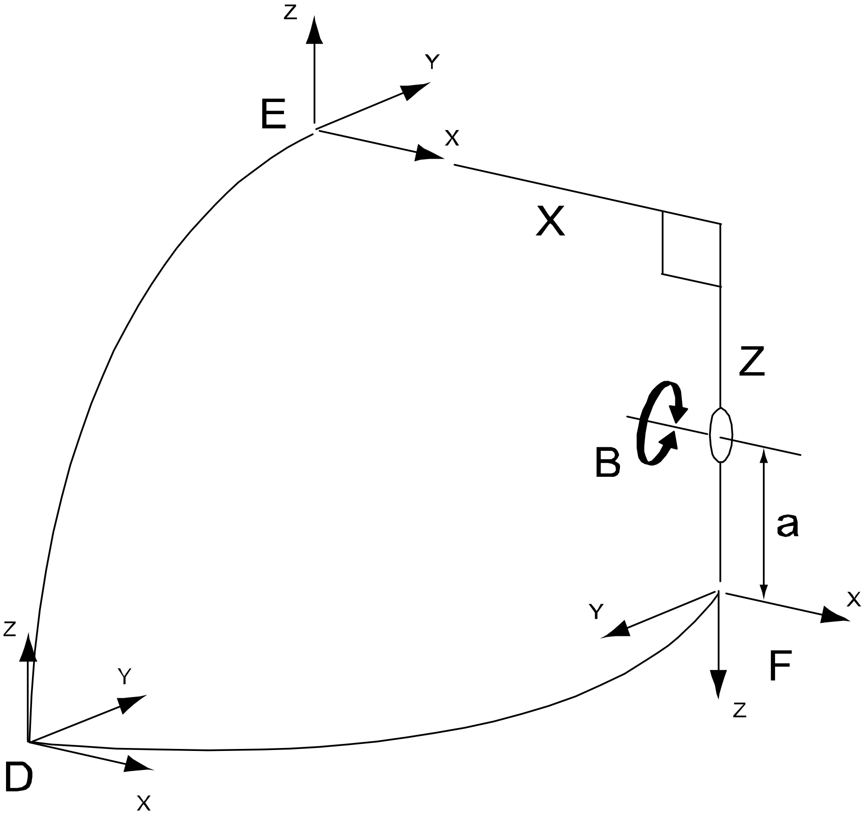
D	World Frame
E	Base Frame
F	Tool Frame
X	X-linear motion
Z	Z-linear motion
C	C rotating around Z axis in base frame

3.2.3.16 Kinematic model XZB(X)

Description

The kinematic model is based on a linear gantry concept, with two linear motions and one rotation.

Illustration



xx0500002111

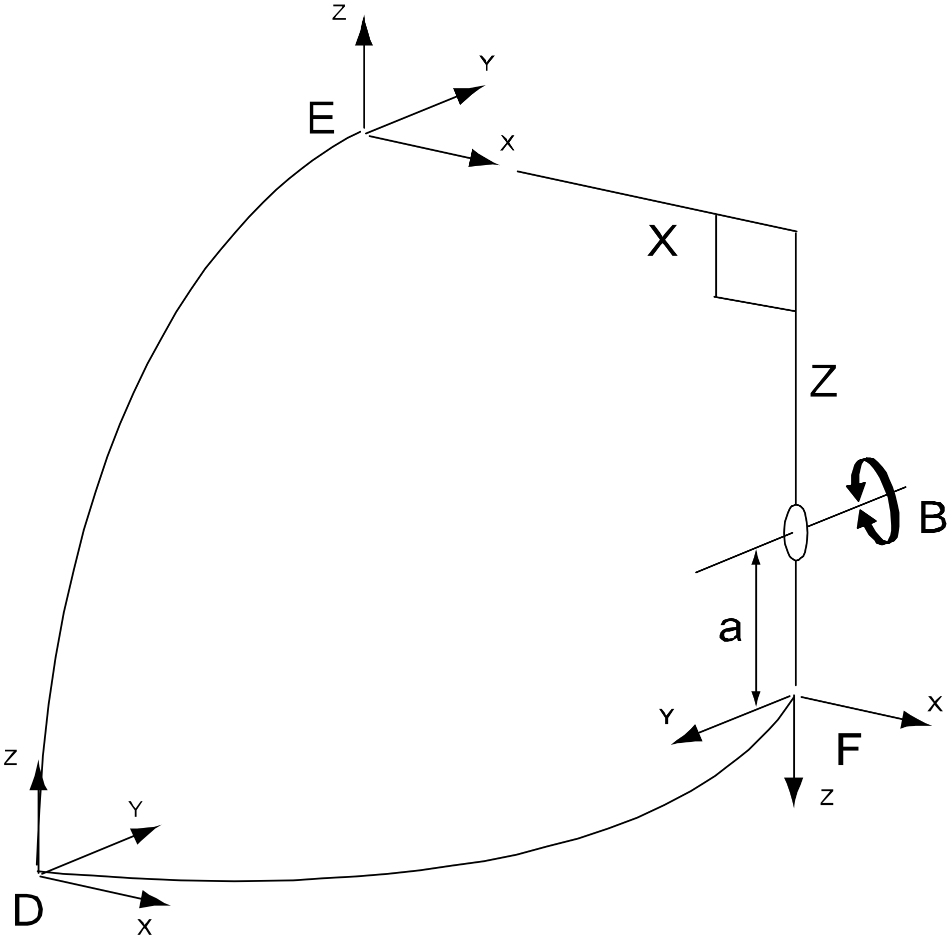
D	World Frame
E	Base Frame
F	Tool Frame
X	X-linear motion
Z	Z-linear motion
B	B rotating around X axis in base frame
a	offset_z of arm “robx_6”

3.2.3.17 Kinematic model XZB(Y)

Description

The kinematic model is based on a linear gantry concept, with two linear motions and one rotation.

Illustration



xx0500002113

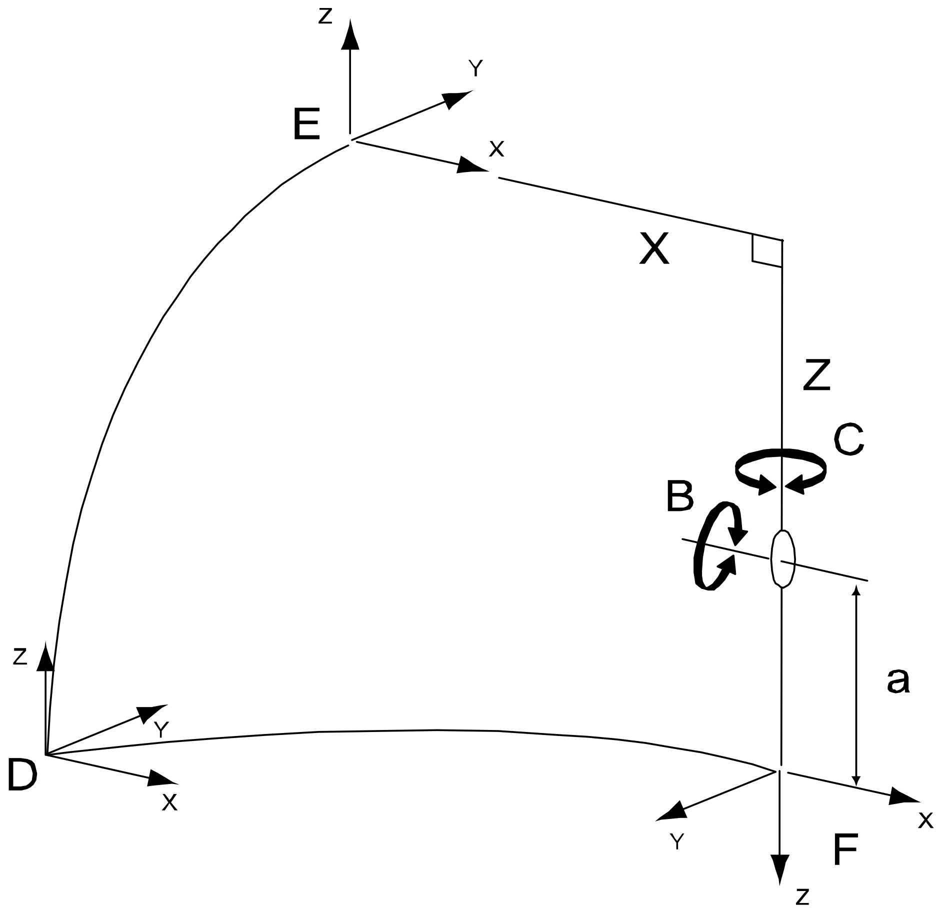
D	World Frame
E	Base Frame
F	Tool Frame
X	X-linear motion
Z	Z-linear motion
B	B rotating around Y axis in base frame
a	offset_z of arm “robx_6”

3.2.3.18 Kinematic model XZC(Z)B(X)

Description

The kinematic model is based on a linear gantry concept, with two linear motions and two rotations.

Illustration



xx0500002116

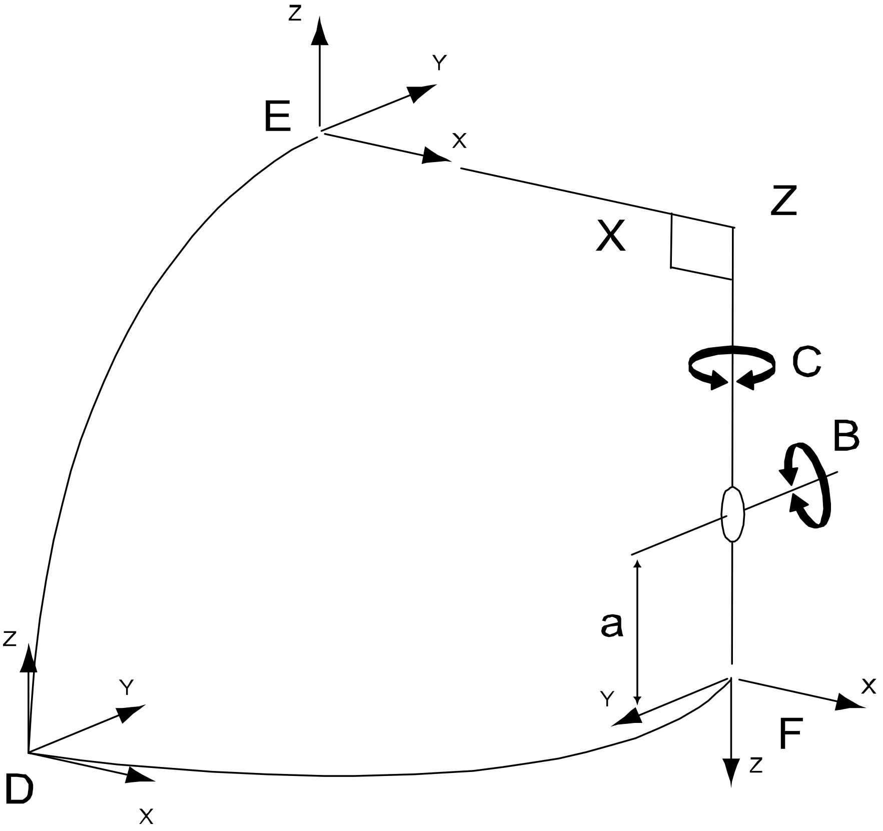
D	World Frame
E	Base Frame
F	Tool Frame
X	X-linear motion
Z	Z-linear motion
C	C rotating around Z axis in base frame
B	B rotating around X axis in base frame
a	offset_z of arm “robx_6”

3.2.3.19 Kinematic model XZC(Z)B(Y)

Description

The kinematic model is based on a linear gantry concept, with two linear motions and two rotations.

Illustration



xx0500002118

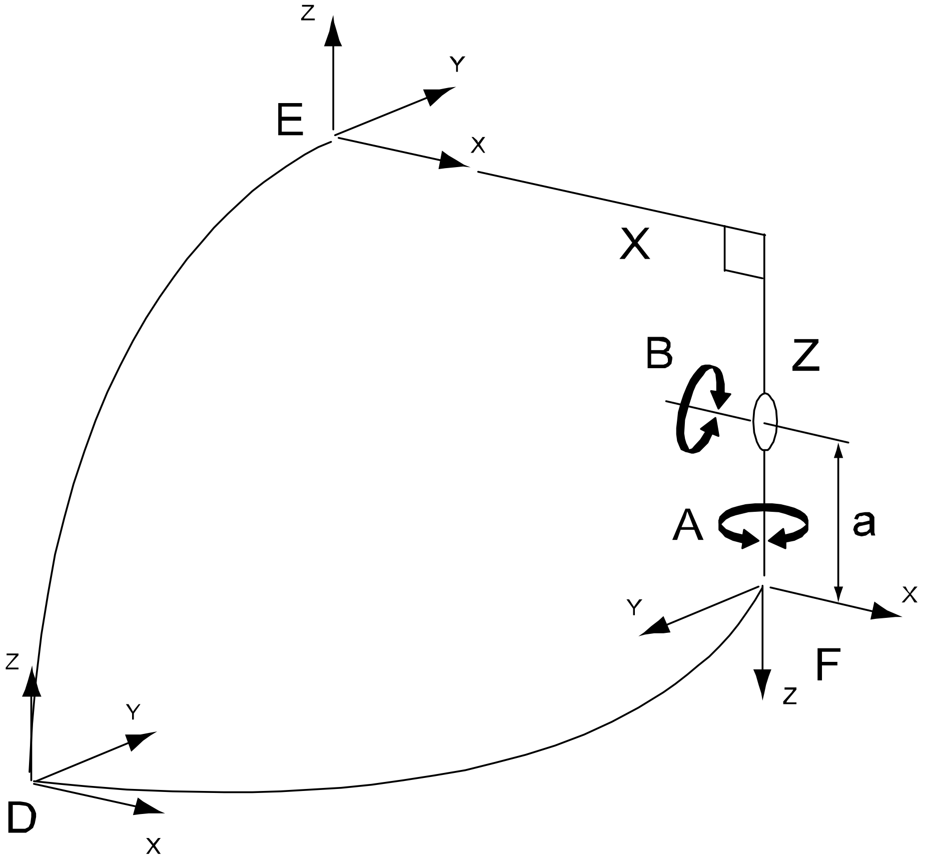
D	World Frame
E	Base Frame
F	Tool Frame
X	X-linear motion
Z	Z-linear motion
C	C rotating around Z axis in base frame
B	B rotating around Y axis in base frame
a	*offset_z* of arm “robx_6”

3.2.3.20 Kinematic model XZB(X)A(Z)

Description

The kinematic model is based on a linear gantry concept, with two linear motions and two rotations.

Illustration



xx0500002112

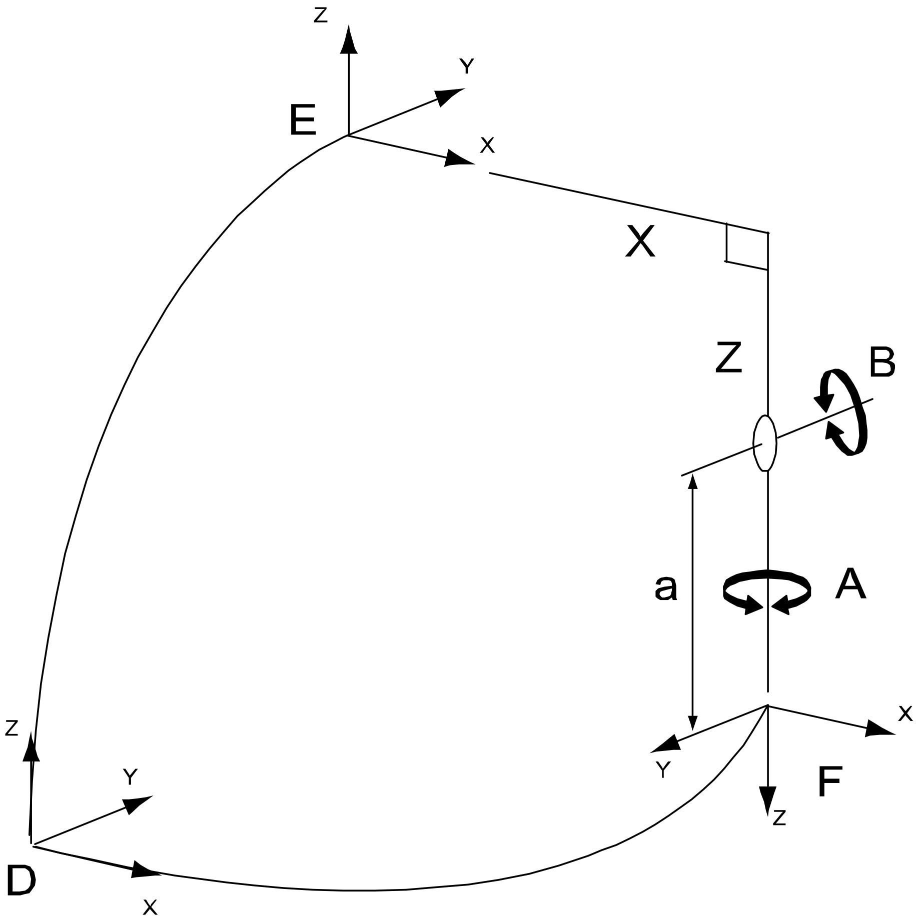
D	World Frame
E	Base Frame
F	Tool Frame
X	X-linerar motion
Z	Z-linear motion
B	B rotating around X axis in base frame
A	A rotating around Z axis in base frame
a	*offset_z* of arm “robx_6”

3.2.3.21 Kinematic model XZB(Y)A(Z)

Description

The kinematic model is based on a linear gantry concept, with two linear motions and two rotations.

Illustration



xx0500002114

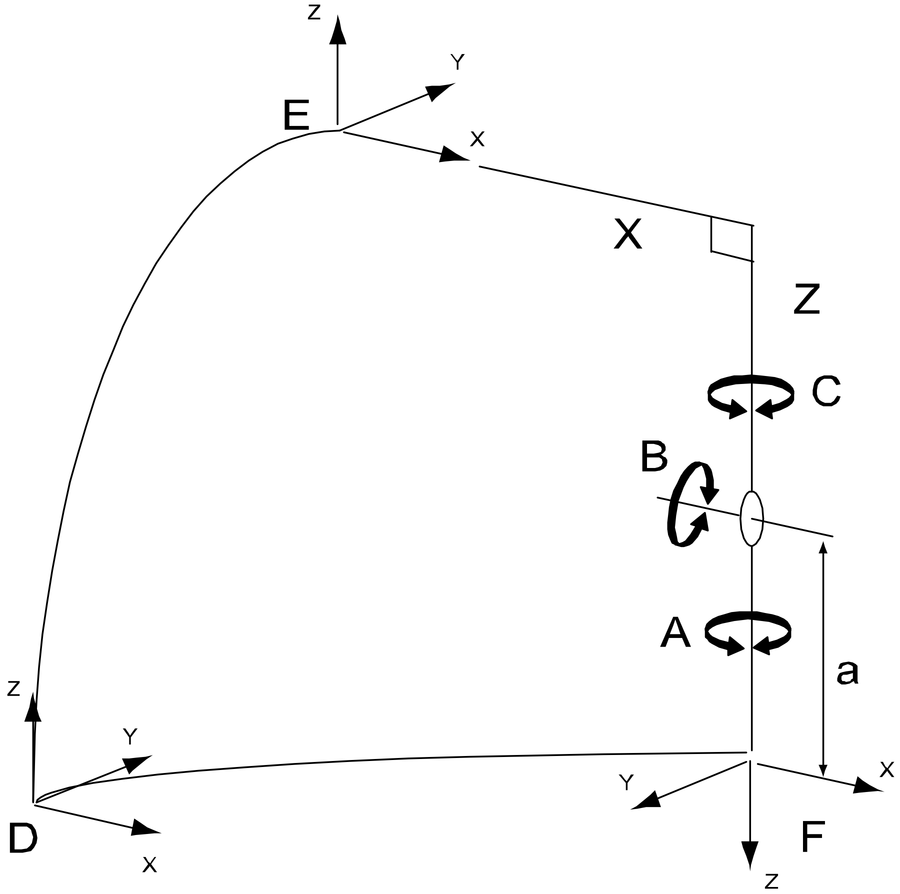
D	World Frame
E	Base Frame
F	Tool Frame
X	X-linear motion
Z	Z-linear motion
B	B rotating around Y axis in base frame
A	A rotating around Z axis in base frame
a	*offset_z* of arm “robx_6”

3.2.3.22 Kinematic model XZC(Z)B(X)A(Z)

Description

The kinematic model is based on a linear gantry concept, with two linear motions and three rotations.

Illustration



xx0500002117

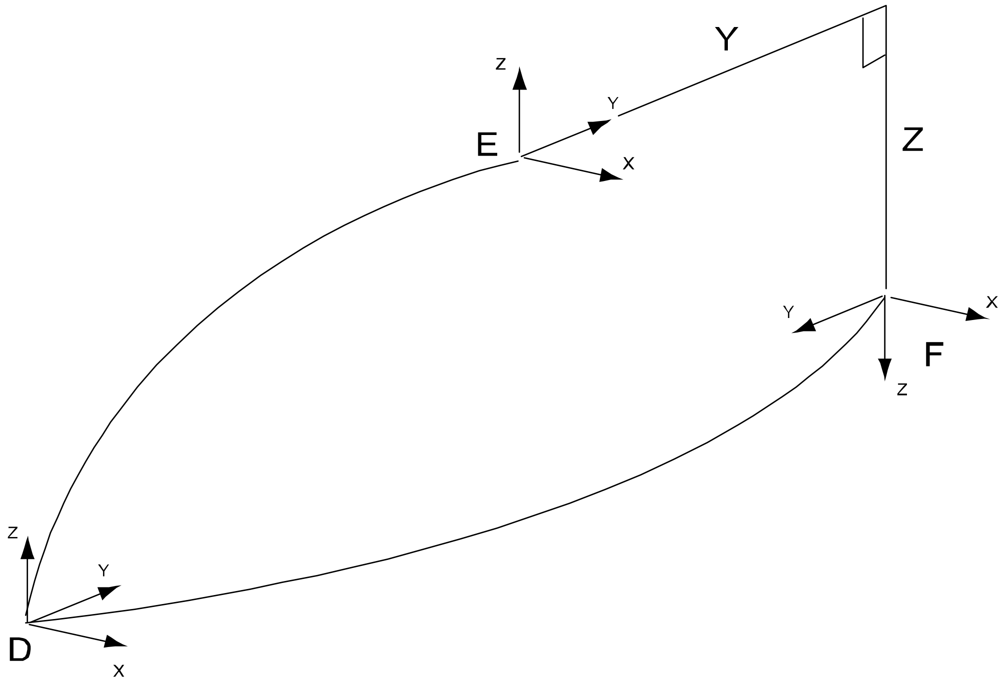
D	World Frame
E	Base Frame
F	Tool Frame
X	X-linear motion
Z	Z-linear motion
C	C rotating around Z axis in base frame
B	B rotating around X axis in base frame if C is zero
A	A rotating around Z axis in base frame
a	*offset_z* of arm “robx_6”

3.2.3.23 Kinematic model YZ

Description

The kinematic model is based on a linear gantry, with two linear motions and no rotation.

Illustration



xx0300000622

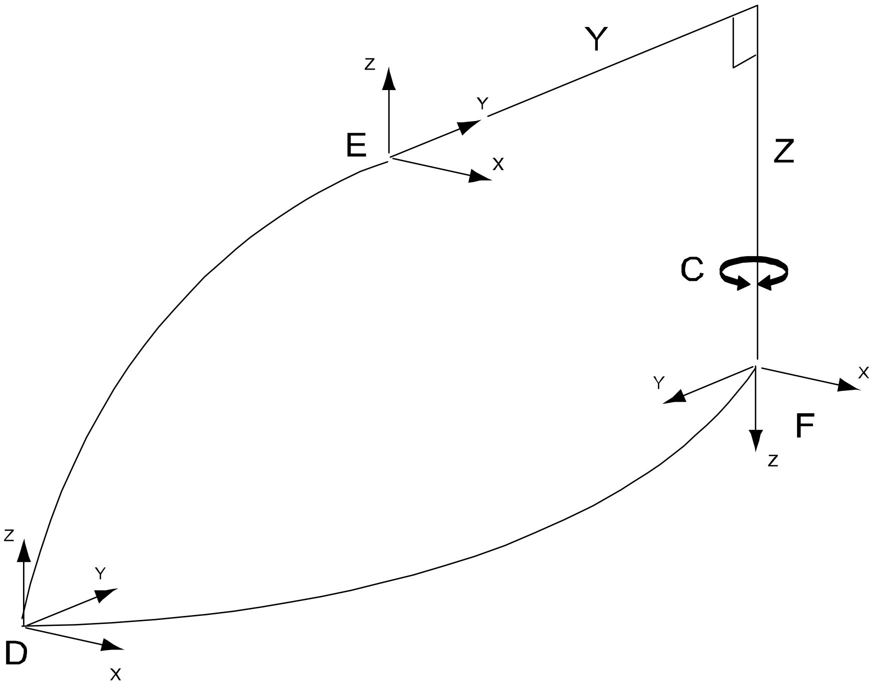
D	World frame
E	Base frame
F	Tool frame
Y	Y-linear motion
Z	Z-linear motion

3.2.3.24 Kinematic model YZC(Z)

Description

The kinematic model is based on a linear gantry, with two linear motions and one rotation.

Illustration



xx0300000623

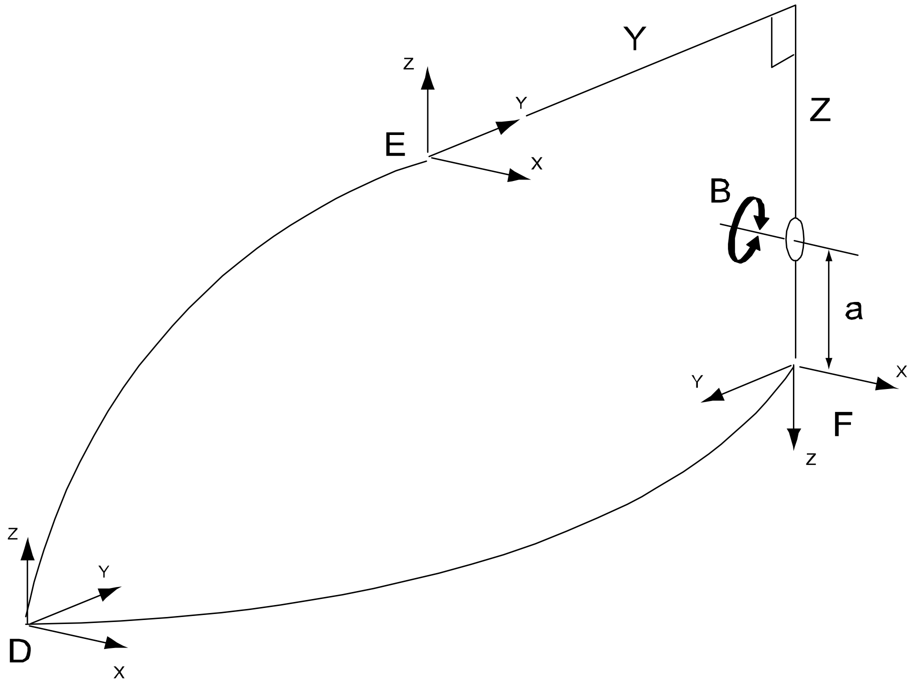
D	World frame
E	Base frame
F	Tool frame
Y	Y-linear motion
Z	Z-linear motion
C	C rotating around Z axis in base frame

3.2.3.25 Kinematic model YZB(X)

Description

The Y_Z_B(X) is a kinematic model, based on a linear gantry, with two linear motions and one rotation.

Illustration



xx0300000624

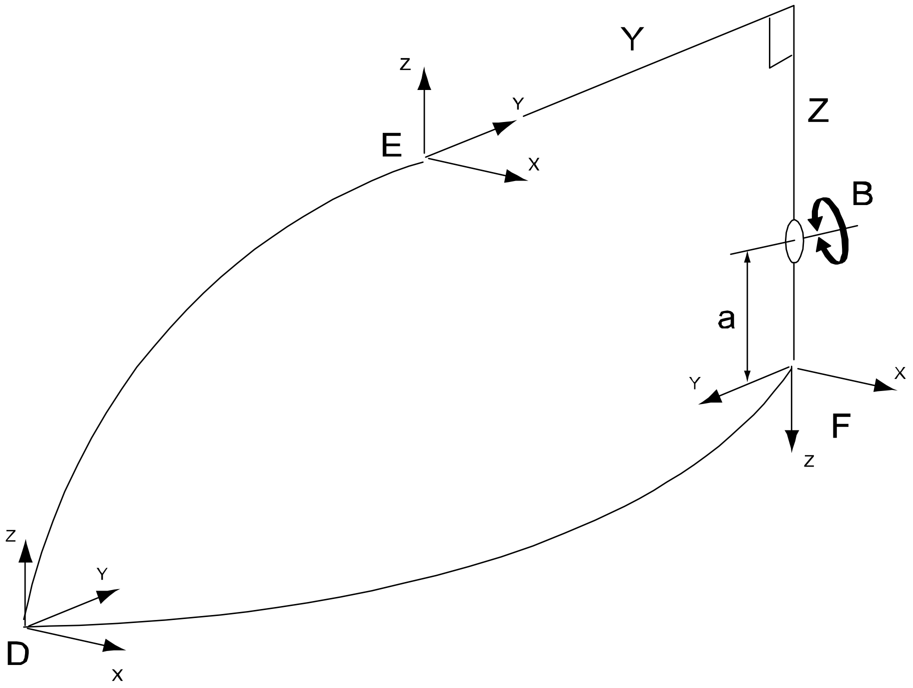
D	World frame
E	Base frame
F	Tool frame
Y	Y-linear motion
Z	Z-linear motion
B	B rotating around X axis in base frame
a	*offset_z* of arm “robx_6”

3.2.3.26 Kinematic model YZB(Y)

Description

The kinematic model is based on a linear gantry, with two linear motions and one rotation.

Illustration



xx0300000625

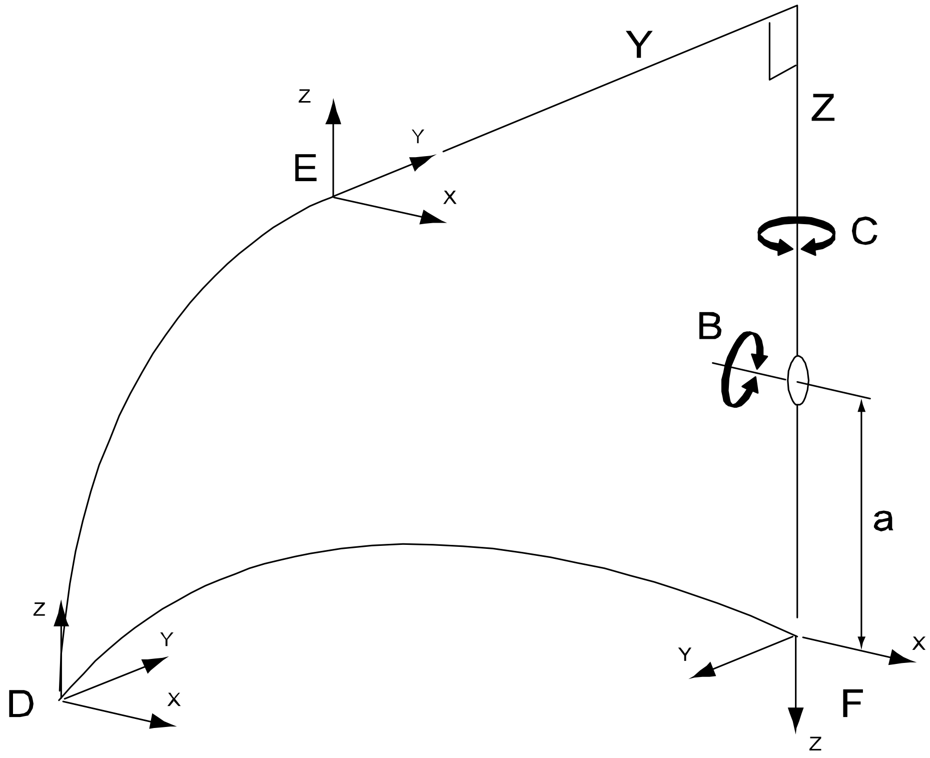
D	World frame
E	Base frame
F	Tool frame
Y	Y-linear motion
Z	Z-linear motion
B	B rotating around Y axis in base frame
a	*offset_z* of arm “robx_6”

3.2.3.27 Kinematic model YZC(Z)B(X)

Description

The kinematic model is based on a linear gantry concept, with two linear motions and two rotations.

Illustration



xx0500002119

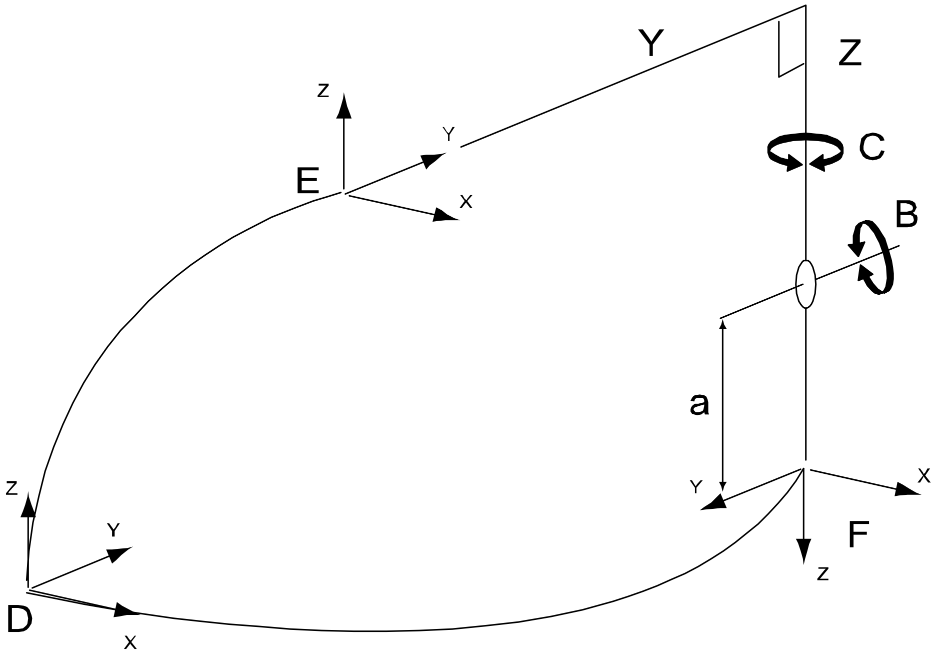
D	World Frame
E	Base Frame
F	Tool Frame
Y	Y-linear motion
Z	Z-linear motion
C	C rotating around Z axis in base frame
B	B rotating around X axis in base frame if C is zero
a	*offset_z* of arm “robx_6”

3.2.3.28 Kinematic model YZC(Z)B(Y)

Description

The kinematic model is based on a linear gantry concept, with two linear motions and two rotations.

Illustration



xx0500002120

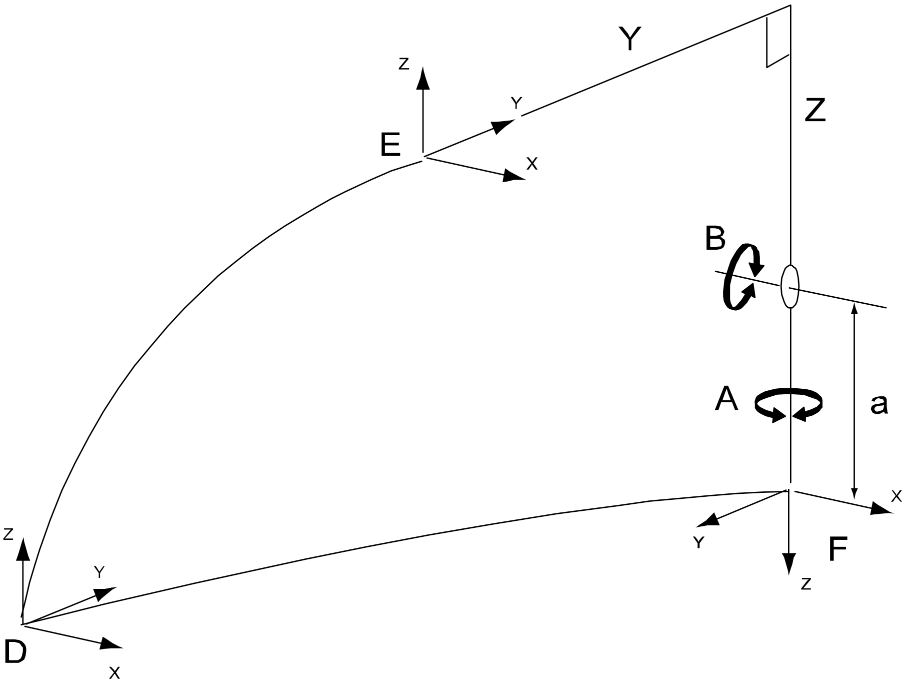
D	World Frame
E	Base Frame
F	Tool Frame
Y	Y-linear motion
Z	Z-linear motion
C	C rotating around Z axis in base frame
B	B rotating around Y axis in base frame
a	offset_z of arm “robx_6”

3.2.3.29 Kinematic model YZB(X)A(Z)

Description

The kinematic model is based on a linear gantry, with two linear motions and two rotations.

Illustration



xx0300000626

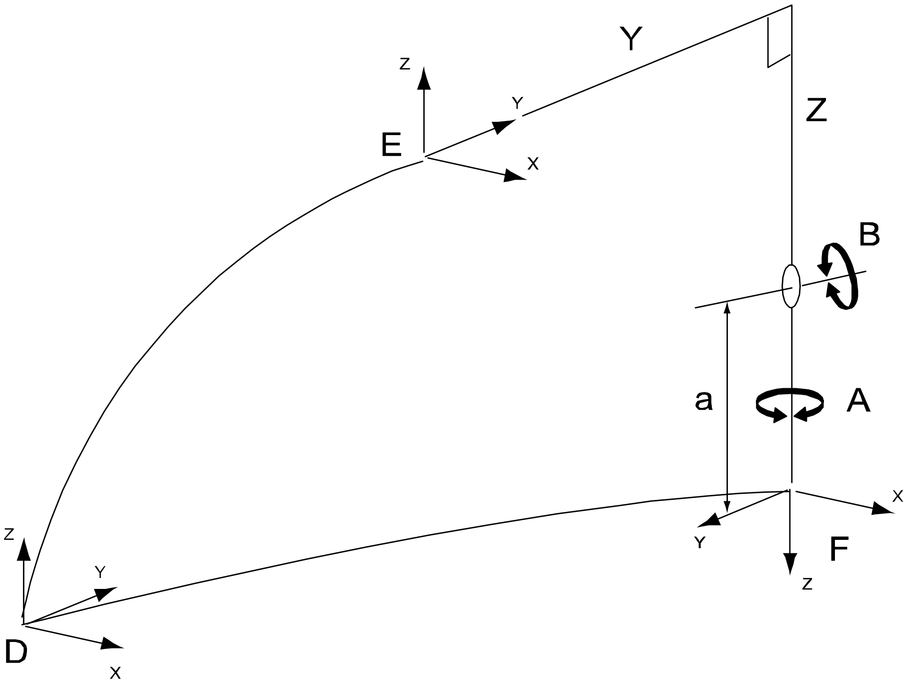
D	World frame
E	Base frame
F	Tool frame
Y	Y-linear motion
Z	Z-linear motion
B	B rotating around X axis in base frame
A	A rotating around Z axis in base frame if B is zero
a	*offset_z* of arm “robx_6”

3.2.3.30 Kinematic model YZB(Y)A(Z)

Description

The kinematic model is based on a linear gantry, with two linear motions and two rotations.

Illustration



xx0300000627

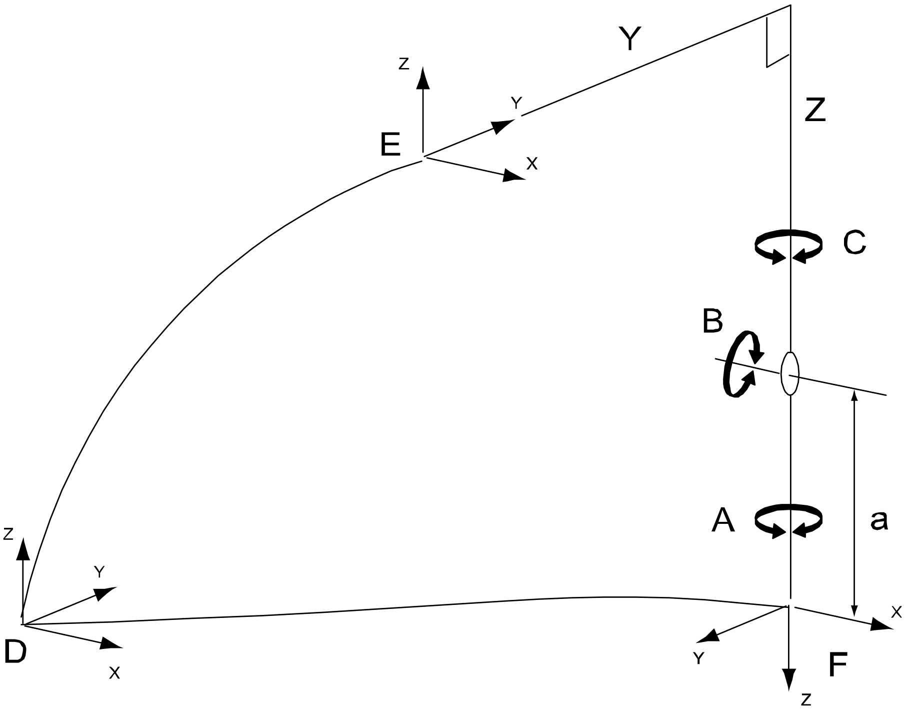
D	World frame
E	Base frame
F	Tool frame
Y	Y-linear motion
Z	Z-linear motion
B	B rotating around Y axis in base frame
A	A rotating around Z axis in base frame if B is zero
a	*offset_z* of arm “robx_6”

3.2.3.31 Kinematic modelYZC(Z)B(X)A(Z)

Description

The kinematic model is based on a linear gantry, with two linear motions and three rotations.

Illustration



xx0300000628

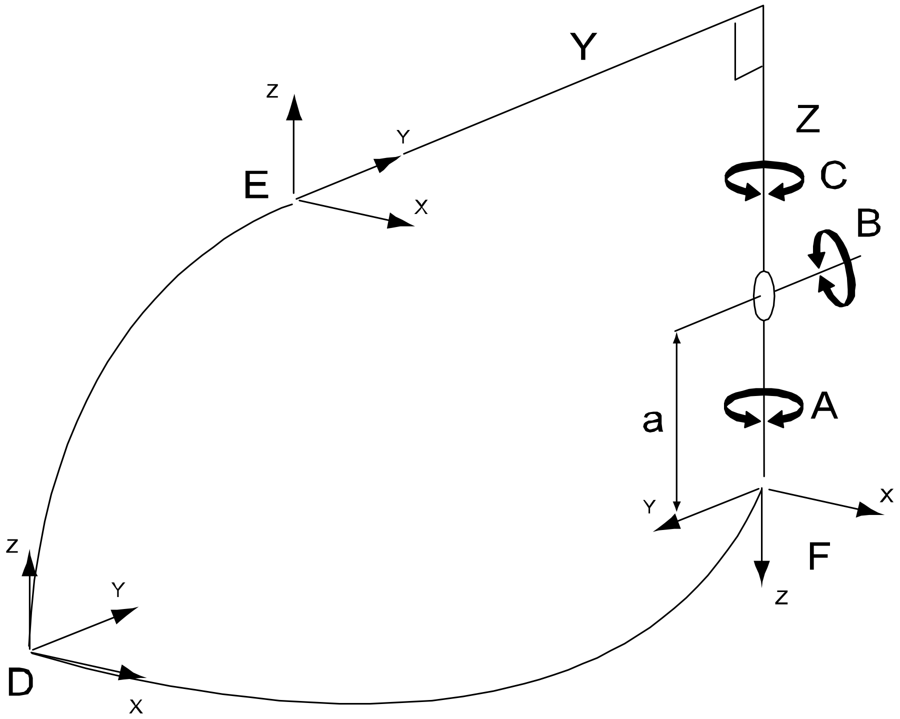
D	World frame
E	Base frame
F	Tool frame
Y	Y-linear motion
Z	Z-linear motion
C	C rotating around Z axis in base frame
B	B rotating around X axis in base frame when C is zero
A	A rotating around Z axis in base frame if B is zero
a	*offset_z* of arm “robx_6”

3.2.3.32 Kinematic model YZC(Z)B(Y)A(Z)

Description

The kinematic model is based on a linear gantry, with two linear motions and three rotations.

Illustration



xx0500002223

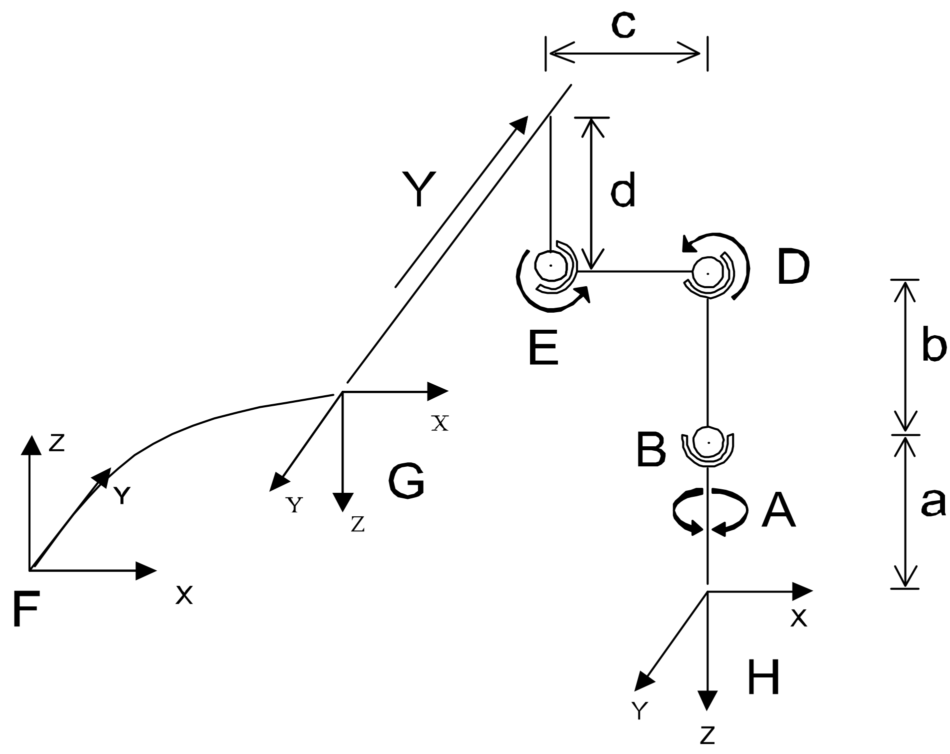
D	World frame
E	Base frame
F	Tool frame
Y	Y-linear motion
Z	Z-linear motion
C	C rotating around Z axis in base frame
B	B rotating around Y axis in base frame when C is zero
A	A rotating around Z axis in base frame if B is zero
a	*offset_z* of arm “robx_6”

3.2.3.33 Kinematic model YE(Y)D(Y)B(Y)A(Z)

Description

The five axes kinematic model is based on a TopLoader concept. It consists of a manipulator with five rotating axes that can move with a linear movement. The home position is shown in the figure below.

Illustration



xx0500002224

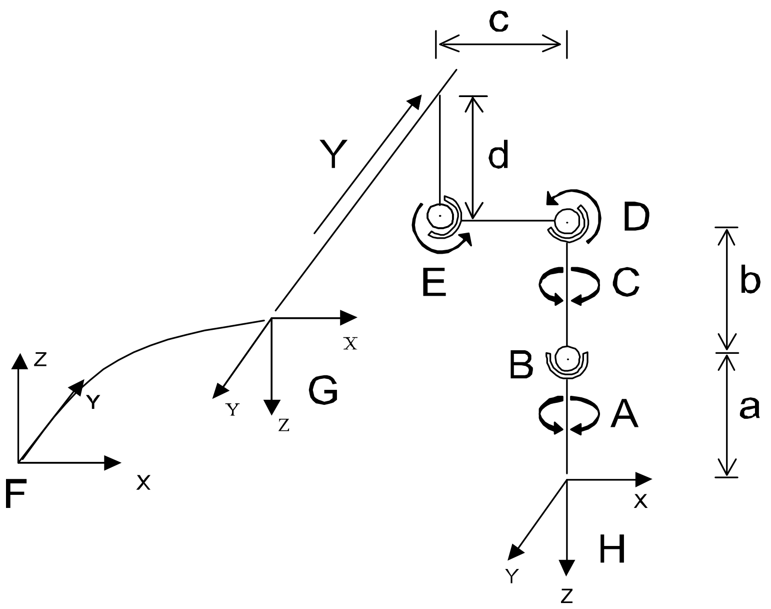
F	World frame
G	Base frame
H	Tool frame
Y	Joint "robx_1" linear along Y axis in base frame
E	Joint "robx_2" rotating around Y axis in base frame
D	"robx_3" rotating around Y axis in base frame
B	Joint "robx_5" rotating around Y axis in base frame if the others are zero
A	Joint "robx_6" rotating around Z axis in base frame if the others are zero
a	*offset_z* of arm "robx_6"
b	*offset_z* of arm "robx_4"
c	*offset_z* of arm "robx_3"
d	*offset_z* of arm "robx_2"

3.2.3.34 Kinematic model YE(Y)D(Y)C(Z)B(Y)A(Z)

Description

The six axes kinematic model is based on a TopLoader concept. It consists of a manipulator with five rotating axes that can move with a linear movement. The home position is shown in the figure below.

Illustration



xx0300000629

F	World frame
G	Base frame
H	Tool frame
Y	Joint "robx_1" linear along Y axis in base frame
E	Joint "robx_2 " rotating around Y axis in base frame
D	"robx_3" rotating around Y axis in base frame
C	Joint "robx_4" rotating around Z axis in base frame if the others are zero
B	Joint "robx_5" rotating around Y axis in base frame if the others are zero
A	Joint "robx_6" rotating around Z axis in base frame if the others are zero
a	*offset_z* of arm “robx_6”
b	*offset_z* of arm “robx_4”
c	*length* of arm “robx_3”
d	*length* of arm “robx_2”

3.2.3.35 Doppin Feeder

Description

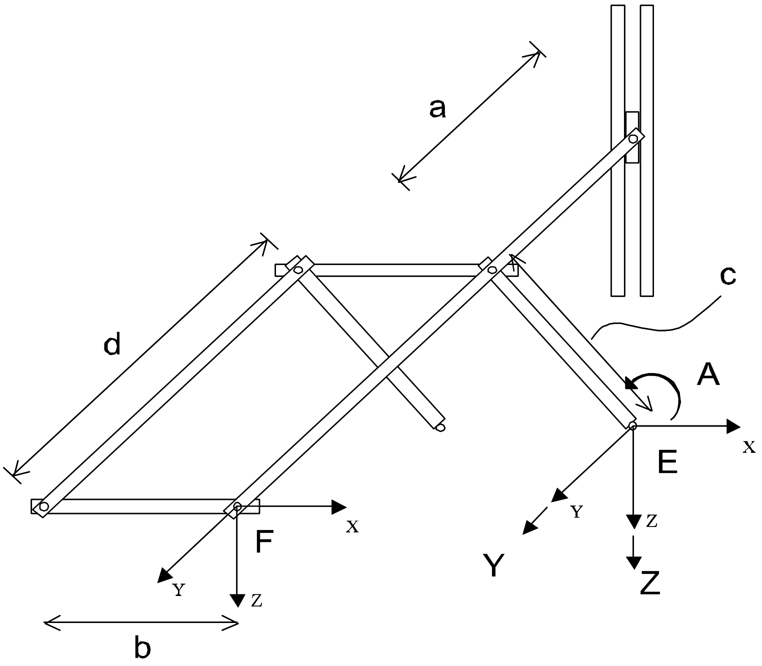
The Doppin Feeder is a two or three axes kinematical model. (doppin_2; 2 axes and doppin_3; 3 axes)

Home position for this model is with the arm "robx_2" pointing vertically upwards.

Illustration

The figure below illustrates the kinematic model for the Doppin Feeder.

Note! The moving revolute joint "robx_2" will result in a non-linear motion of the tool frame if the length of the arm "robx_2" is not equal to the length of the arm "robx_3". The linear motion along "robx_1" and "robx_3" moves the whole mechanism.



xx0300000630

E	Base frame
F	Tool frame
Y	Joint "robx_1" linear along Y axis in base frame
A	Joint "robx_2" rotating around Y axis in base frame
Z	Joint "robx_3" linear along Z axis in base frame
a	*offset_z* of arm "robx_2"
b	*offset_x* of arm "robx_2"
c	*length* of arm "robx_2"
d	*length* of arm "robx_3"

3.2.4 Creating a stand alone controller system

Overview

This section describes how to create a stand alone controller system using the Installation Manager in RobotStudio.

General procedure

Follow these basic steps to create a stand alone controller system. For more information on how to install RobotWare, SAC (stand alone controller), and create a new system see *Operating manual - RobotStudio*.

	Action
1	Install RobotWare, as described in *Operating manual - RobotStudio*.
2	Install the SAC Add-In, as described in *Operating manual - RobotStudio*.
3	Create a stand alone controller system using the Installation Manager in RobotStudio, see *Installation Manager procedure on page 63*.

Installation Manager procedure

General information about creating a new system is available in the **Help** menu in RobotStudio. This section gives information specific for the stand alone controller option.

	Action
1	Open the Installation Manager in RobotStudio.
2	Add the products for RobotWare and SAC.
3	Add the licenses for RobotWare and SAC.
4	The next dialog is used to modify options. Select the **Drive Modules** tab and expand the **SAC** node in the tree view.
5	Select the appropriate kinematic model to be used under the **First Mechanical Unit** node.
6	If the system has several mechanical units, a kinematic model for each one of them should be selected. Continue by selecting kinematic models for **Second Mechanical Unit** etc.
7	The next dialog is used to verify all selections before downloading the system to the IRC5 controller. • Check that the correct drive system has been selected.
8	Click **Apply** to download the system.

Errors at start up

When the system is ready with start-up, inform yourself on system status by studying the event log on the FlexPendant or in RobotStudio.

A system with non-ABB equipment needs configuration to become functional, and it is even quite likely that your system is in system failure state at this point. Ignore any errors until you are ready with the configuration procedure described in section *Minimal configuration of non-ABB robots on page 72*.

If there are remaining errors after configuration is done find out more about error localization in section *Error management on page 131*.

This page is intentionally left blank

4 Configuration

4.1 Basic settings

4.1.1 Limit peripheral speed of external axis

! CAUTION

The manual mode peripheral speed of the external axis must be restricted to 250mm/s for personal safety reasons. The speed is supervised at two different levels, which means that two system parameters need to be set up. All two parameters belong to the type *Supervision Type* in the configuration topic *Motion* and are expressed in ratio of max speed (1 = 100%).

Calculate parameter values

Teach Max Speed Main

Teach Max Speed Main = (0.25 / *Arm Length*) * (*Transmission Gear Ratio* / *Speed Absolute Max*)

where:

- *Transmission Gear Ratio* belongs to the type *Transmission*.
- *Speed Absolute Max* belongs to the type *Stress Duty Cycle* (rad/s).
- *Arm Length* should be measured from the rotational center of the external axis (meter).

If the result of the calculation exceeds 0.94, use 0.94 instead of the calculated value.

Insert the calculated result at the type *Supervision Type: Teach Max Speed Main*.

Teach Max Speed DSP

Calculate and use the largest value of:

- *Teach Max Speed Main* * 1.20
- *Teach Max Speed Main* + (8 / *Speed Absolute Max*)

Insert the calculated result at the type *Supervision Type*: *Teach Max Speed DSP*.

Example

Given parameter values

Transmission Gear Ratio = 120

Speed Absolute Max = 320 rad/s

Arm Length = 0.5 m

Calculations

Teach Max Speed Main = (0.25 /*Arm Length*) * (*Transmission Gear Ratio* / *Speed Absolute Max*) = (0.25 / 0.5) * (120 / 320) = 0.188

Continues on next page

Teach Max Speed Dsp = max{(*Teach Max Speed Main* * 1.20) , (*Teach Max Speed Main* + (8 / *Speed Absolute Max*))} = max{(0.188 * 1.2) , (0.188 + (8 / 320))} = max {0.226 , 0.213} = 0.226

4.1.2 Minimal configuration of general additional axes

Overview

This section describes how to make a minimal configuration of a standard additional axes.

> ⚠ **WARNING**
>
> Incorrect definition of system parameters for brakes or additional axes may cause damage to the robot or personal injury.

Load parameters

Use RobotStudio to perform the following instructions. See *Operating manual - RobotStudio*.

	Action
1	Right click on configuration icon in the system view, and select **Load Parameters**.
2	Select **Load parameters if no duplicates** and click **Open**.
3	Browse to the template files in the RobotWare installation, see *Template files on page 22*. • For general additional axis, browse to the directory: *...\utility\AdditionalAxis\DriveSystem 09\General\DM1* • For track motion, browse to the directory: *...\utility\AdditionalAxis\DriveSystem 09\Track\DM1*
4	Select the configuration file for required axes and click **Open**.
5	Perform a warm start of the system from the FlexPendant or RobotStudio.

Configure parameters

Use RobotStudio or the FlexPendant to perform the following instructions. See *Operating manual - RobotStudio*.

For parameter description, see *System parameters on page 135*.

	Action	Info/Illustration
1	Select the topic *Motion* and type *Mechanical Unit* and define the following in the parameter. ℹ **Note** For a single axis mechanical unit without kinematic model, *Name* and *Use Single 1* in the type *Mechanical Unit* and *Name* in the type *Single* must be the same.	• *Name* • *Activate at Start Up* • *Deactivation Forbidden* • *Use Single 1* • *Allow Move of User Frame*
2	Select the topic *Motion* and type *Single* and specify which *Single Type* to use.	• *Name* • *Use Single Type*
3	Select the topic *Motion* and type *Single Type* and specify the type of additional axis in the parameter *Mechanics*.	Example of values of the parameter *Mechanics*: • TRACK • FREE_ROT

Continues on next page

	Action	Info/Illustration
4	Select the topic *Motion* and type *Joint* and set the parameter *Logical Axis* to the logical axis number.	Example: Logical axis 10 will then correspond to the field `eax_d` in the RAPID data type `robtarget`.
5	Select the topic *Motion* and type *Arm* and specify the arm characteristics for the axis.	• *Upper Joint Bound* • *Lower Joint Bound*
6	Select the topic *Motion* and type *Acceleration Data* and specify the arm performance for the axis.	• *Nominal Acceleration* • *Nominal Deceleration*
7	Select the topic *Motion* and type *Transmission* and specify the following.	• *Transmission Gear Ratio* • *Rotating Move* • *Transmission High Gear* • *Transmission Low Gear*
8	Select the topic *Motion* and type *Motor Type* and specify the following.	• *Polepairs* • *ke Phase to Phase* (Vs/rad) • *Max current* (A rms) • *Phase resistance* (ohm) • *Phase inductance* (H)
9	Select the topic *Motion* and type *Motor Calibration* and define the calibration and commutation offsets.	• *Calibration Offset* • *Commutator Offset*
10	Select the topic *Motion* and type *Stress Duty Cycle* and define the torque and speed absolute max on the motor side.	• *Torque Absolute Max* (Nm) • *Speed Absolute Max* (rad/s)
11	Perform a warm start of the system from the FlexPendant or RobotStudio.	

Note

If *Torque Absolute Max* is too high it may result in a configuration error at restart.

Limitations

If *Torque Absolute Max* is to high it may result in a configuration error at restart. To avoid error do not set *Torque Absolute Max* higher than:

Torque Absolute Max < √ 3 * *ke Phase to Phase (Vs/rad)* * *Max Current*

where:

- *Max Current*, belonging to the type *Motor Type*, is the maximum current of the used drive module
- *ke Phase to Phase (Vs/rad)*, belonging to the type *Motor Type*, is a voltage constant

4.1.3 Minimal configuration of servo gun

Overview

This section describes how to configure a servo gun. It details the usage of important parameters, some of them servo gun specific, which need to be set up. Most of the advanced settings, such as relays, brakes and supervision, which are described in the following sections, are also valid for servo guns.

WARNING

Incorrect definition of system parameters for brakes or additional axes may cause damage to the robot or personal injury.

Load parameters

Use RobotStudio to perform the following instructions. See *Operating manual - RobotStudio*.

	Action
1	Right click on configuration icon in the system view, and select **Load Parameters.**
2	Select **Load parameters if no duplicates** and click **Open**.
3	Browse to the template files in the RobotWare installation, see *Template files on page 22*. *...\utility\AdditionalAxis\DriveSystem 09\ServoGun*
4	Select the configuration file for required axes and click **Open**.
5	Perform a restart of the system from the FlexPendant or RobotStudio.

Configure parameters

Use RobotStudio to perform the following instructions. See *Operating manual - RobotStudio*.

For parameter description, see *System parameters on page 135*.

	Action	Info/Illustration
1	Select the topic *Motion* and the type *Mechanical Unit* and define the following parameter:	• *Name*
2	Select the topic *Motion* and the type *Joint* and specify the logical axis number under parameter *Logical Axis*.	Example: Logical axis 10 will then correspond to the field `eax_d` in a RAPID data of the type `robtarget`.
3	Select the topic *Motion* and the type *Arm* and specify the arm characteristics for the axis.	• *Upper Joint Bound* • *Lower Joint Bound* *Lower Joint Bound* should be set to zero or a small negative value (e.g. -0.005 m) in order to protect the gun from collisions. The limit is not active during force control of the gun. For force control there is another positional limit, *Max Force Control Position Error*, in the type *Supervision*.

Continues on next page

	Action	Info/Illustration
4	Select the topic *Motion* and the type *Acceleration Data* and specify the arm performance for the axis.	• *Nominal Acceleration* • *Nominal Deceleration*
5	Select the topic *Motion* and the type *Transmission* and specify the following parameters:	• *Transmission Gear Ratio*
6	Select the topic *Motion* and the type *Motor Type* and specify the following parameters:	• *Pole Pairs* • *ke Phase to Phase* (Vs/rad) • *Max Current* (A rms) • *Phase Resistance* (ohm) • *Phase Inductance* (H)
7	Select the topic *Motion* and the type *Motor Calibration* and define the calibration and commutation offsets.	• *Calibration Offset* • *Commutation Offset*
8	Select the topic *Motion* and type *Stress Duty Cycle* and define the torque and speed absolute max on the motor side.	• *Torque Absolute Max* (Nm) • *Speed Absolute Max* (rad/s)
9	Perform a restart of the system from the FlexPendant or RobotStudio.	

Tuning

After configuration additional axis tuning needs to be performed. See *Tuning on page 107* for tuning of the system.

Configure servo gun parameters

After tuning, the servo gun specific parameters can be defined.

	Action	Info/Illustration
1	Select the topic *Motion* and the type *Supervision Type* and define the supervision limits during force control.	• *Max Force Control Position Error* • *Max Force Control Speed Limit*
2	Select the topic *Motion* and the type *SG Process* and define the process parameters specific for servo gun.	• *Sync check off* • *Close Time Adjust* • *Force Ready Delay* • *Max Force Control Motor Torque* • *Post-synchronization Time* • *Calibration Mode* • *Calibration Force High* • *Calibration Force Low* • *Calibration Time*

Continues on next page

Tip force

The relationship between the programmed tip force and the resulting motor torque is set up in the following parameters. The torques may be negative due to the sign of the gear ratio while the forces must always be positive. Before setting up this table, the parameters in the Force Master should be tuned. See *Application manual - Servo gun tuning*.

The easiest way to set up the table is by using a RAPID force calibration service routine.

Parameter	Description
Number of Stored Forces	Number of stored forces in the force vs motor torque table. The minimum value allowed is 2.
Tip Force 1	Gun tip force 1 (N)
Motor Torque 1	Motor torque 1 (Nm)
Tip Force 2	Gun tip force 2 (N)
Motor Torque 2	Motor torque 2 (Nm)
Tip Force 10	Gun tip force 10 (N)
Motor Torque 10	Motor torque 10 (Nm)

4.1.4 Minimal configuration of non-ABB robots

Overview

This section describes basic configuration of non-ABB robots.

> ⚠ **WARNING**
>
> Incorrect definition of system parameters for brakes or additional axes may cause damage to the robot or personal injury.

General approach

For each kinematic model a corresponding set of default configuration files are supplied with the additional option *Stand Alone Controller*. It is possible to configure system parameters by editing these configuration files directly with a text editor. The recommended way, however, is to use RobotStudio or the FlexPendant.

Configure system parameters

Use RobotStudio or the FlexPendant to configure the following system parameters for non-ABB robots. They all belong to the configuration topic *Motion*. For more information on how to do this see *Operating manual - IRC5 with FlexPendant* and *Operating manual - RobotStudio*. For more information about the parameters see *System parameters on page 135*.

	Action	Parameter name
1	Select the type *Robot* and specify name. **Note!** Naming a robot is optional but often convenient.	• Name
2	Select the type *Measurement Channel* and specify:	• Measurement Node
3	Select the type *Arm* and define the limits for the robot's working range. There is one set of parameters for each joint. Specify:	• Upper Joint Bound • Lower Joint Bound • Calibration Position
4	Select the type *Arm Type*. Depending on selected kinematic model, different parameters need to be configured.	See *Setting the Arm Type parameters on page 73*.
5	Select the type *Transmission* and specify:	• Transmission Gear Ratio • Rotating Move • Transmission Gear High • Transmission Gear Low
6	Select the type *Brake* and specify brake parameters.	See *Defining brake behavior on page 82*.
7	Select the type *Drive system* and specify:	• Use Drive Unit
8	Select the type *Motor* and specify:	• Use Motor Type

Continues on next page

	Action	Parameter name
9	Select the type *Motor Type* and specify: **Note!** Values for these parameters can be found in the motor specifications.	• Pole Pairs • Stall Torque • Ke Phase to Phase • Max Current • Phase Resistance • Phase Inductance
10	Select the type *Stress Duty Cycle* and specify:	• Speed Absolute Max • Torque Absolute Max
11	If the system uses MultiMove and has several mechanical units attached to the same drive module further configuration is needed.	See *Setting up a motion planner and a RAPID task on page 75*.
12	Check if any advanced configuration needs to be done.	See the chapter *Advanced settings on page 77* in this manual.
13	Fine calibrate the system.	On the FlexPendant tap **Calibration**, select a Mechanical Unit and tap **Fine Calibration**. For more information see *Operating manual - IRC5 with FlexPendant*, section *Calibrating*.
14	Tune the system before starting to use it.	See section *Tuning on page 107*, especially *Tuning of axes, complete procedure on page 110* and *Tuning of Nominal Acceleration and Nominal Deceleration on page 123*.

Setting the Arm Type parameters

Arm Type parameters need to be configured if any of the kinematic models below is used:

- Linear Gantry and Area Gantry with B-rotation
- TopLoader
- Doppin Feeder

Linear Gantry or Area Gantry with B-rotation

Parameters to be changed when using the kinematic model *Linear Gantry* or *Area Gantry with B-rotation*:

For arm...	Parameter name	Description
robx_6 in the default configuration file for kinematic models • XYZB(X) • XYZB(X)A(Z) • XYZC(Z)B(X)A(Z) • YZB(X) • YZB(Y) • YZB(X)A(Z) • etc.	`offset_z`	Length of arm robx_6 (in meter), see the selected *Kinematic models on page 28*.

Continues on next page

TopLoader

Parameters to be changed when using the kinematic model *TopLoader*:

For arm...	Parameter name	Description
robx_2	`length`	Length in meter according to the selected *Kinematic models on page 28*.
robx_3	`length`	Length in meter according to the selected *Kinematic models on page 28*.
robx_4	`offset_z`	Length in meter according to the selected *Kinematic models on page 28*.
robx_6	`offset_z`	Length in meter according to the selected *Kinematic models on page 28*.

Doppin Feeder

Parameters to be changed when using the kinematic model *Doppin Feeder*:

For arm...	Parameter name	Description

robx_2	`length`
	`offset_x`
	`offset_z`
		Length in meter according to the selected *Kinematic models on page 28*.

robx_3	`length`	Length in meter according to the selected *Kinematic models on page 28*.

Several mechanical units on the same drive module

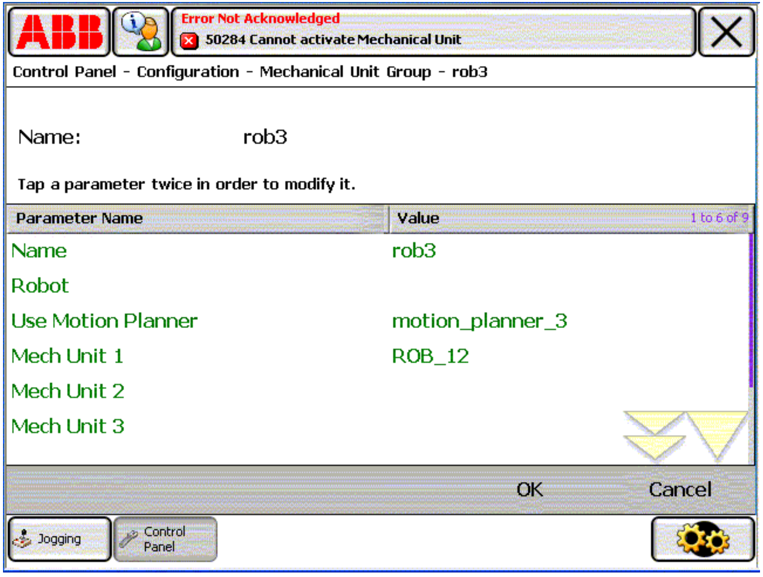
If the system has several mechanical units attached to the same drive module, the system will come up with the error message *50284* - *“Cannot activate Mechanical Unit”*. This is perfectly normal, as no motion planner or RAPID task has been defined by the system for a second or third mechanical unit on a drive module. This means you need to specify a RAPID task and a motion planner for all mechanical units that are not number one on a drive module. For information on how to do this see *Setting up a motion planner and a RAPID task on page 75*.

Continues on next page

Setting up a motion planner and a RAPID task

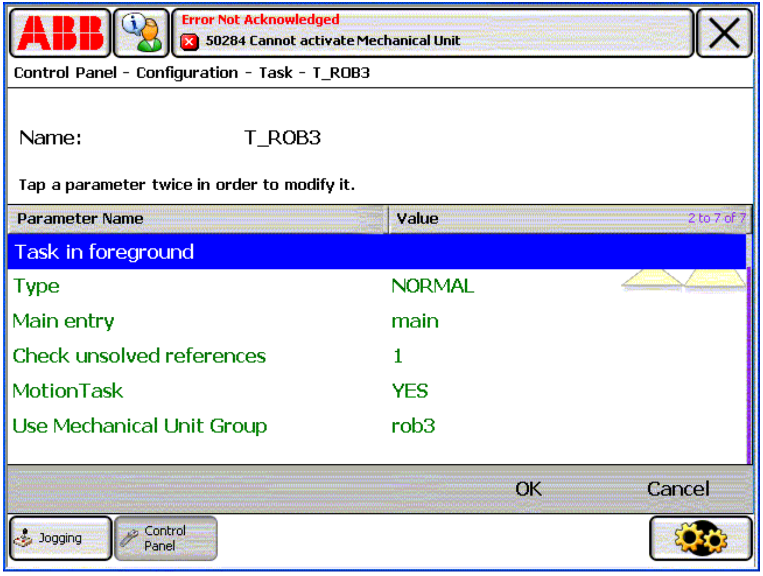
Suppose the system has two drive modules and three mechanical units, two of which are connected to the first drive module. At system setup motion planner 1 and motion planner 2 have been dedicated to the first mechanical units on the respective drive module. A motion planner for the second mechanical unit on drive module one must be configured manually, using either the FlexPendant or RobotStudio.

The following procedure shows how to do this using the FlexPendant.

	Action
1	Select the configuration topic *Controller* and add a new *Mechanical Unit Group*. Connect it to motion planner 3 and specify the second mechanical unit on drive module one (ROB_12) in the parameter *Mech Unit 1*.  Error Not Acknowledged 50284 Cannot activate Mechanical Unit Control Panel - Configuration - Mechanical Unit Group - rob3 Name: rob3 Tap a parameter twice in order to modify it. Parameter Name: Value (1 to 6 of 9) Name: rob3 Robot: Use Motion Planner: motion_planner_3 Mech Unit 1: ROB_12 Mech Unit 2: Mech Unit 3: en0600003035

Continues on next page

<table>
<tr><th></th><th>Action</th></tr>
<tr><td>2</td><td>To be able to program the mechanical unit you need to create a Rapid task. Select the configuration topic <i>Controller</i> and add a new <i>Task</i>. Attach it to the mechanical group created in step 1.



Error Not Acknowledged
50284 Cannot activate Mechanical Unit
Control Panel - Configuration - Task - T_ROB3
Name: T_ROB3
Tap a parameter twice in order to modify it.

<table>
<tr><th>Parameter Name</th><th>Value</th><th>2 to 7 of 7</th></tr>
<tr><td>Task in foreground</td><td></td><td></td></tr>
<tr><td>Type</td><td>NORMAL</td><td></td></tr>
<tr><td>Main entry</td><td>main</td><td></td></tr>
<tr><td>Check unsolved references</td><td>1</td><td></td></tr>
<tr><td>MotionTask</td><td>YES</td><td></td></tr>
<tr><td>Use Mechanical Unit Group</td><td>rob3</td><td></td></tr>
</table>

en0600003036</td></tr>
</table>

4.2 Advanced settings

4.2.1 Disconnect a servo motor

Overview

It is possible to disconnect and reconnect the motor of a deactivated axis if a certain deactivation mode is setup.

> **Note**
>
> If the axis is moved when disconnected, the position of the axis might be wrong after reconnecting, and this will not be detected by the controller. The position after reconnection will be correct if the axis is not moved, or if the movement is less than 0.5 motor revolutions. For servo guns, there is a RAPID calibration method available (the ToolChange calibration) that will adjust any positional error caused by gun movement during disconnection.

Configure parameters

Use RobotStudio to perform the following instructions. See *Operating manual - RobotStudio*

For parameter description, see chapter *System parameters on page 135*.

	Action	Info/Illustration
1	Select the topic *Motion* and type *Measurement Channel* and define the following in the parameter.	• *Disconnect at Deactive*

4.2.2 Servo Tool Change

Overview

With the option *Servo Tool Change* it is possible to disconnect the resolver and power cables from the motor of one external axis and connect them to the motor of another additional axis.

For details about *Servo Tool Change*, see *Application manual - Controller software IRC5*.

WARNING

It is important that no other mechanical unit used with one tool changer are activated, but the one corresponding to the currently connected servo gun! An activation of the wrong mechanical unit may cause unexpected movements and personal injury. See *Defining relays on page 80*.

Note

In case the Servo Gun is equipped with a brake, the 24V to the brake must be switched off before and during servo tool change. This is done via an I/O -signal and brake relay (e.g. by using the instruction `WaitTime` on the brake relay). See *Defining relays on page 80*, for defining of brake relays.

Considerations

The list below specifies special considerations when switching motors:

- The two (or more) additional axes sharing the same motor cables are configured as separate mechanical units.
- The additional axes are configured to use the same measurement node and drive unit node.
- If two servo guns are used with a tool changer, the template file `M7L1B1S_DM1.cfg` can be used for configuration of both guns (change the name of the instance in one of the files).
- A motor switch can be done only if all sharing axes are deactivated.
- The reconnected motor is activated and this activation will restore the position of the axis to the latest position.
- A new motor switch cannot be performed until this axis is deactivated.
- Always use the tool change tip calibration after activation.

Connection Relay

To make sure that the correct mechanical unit is active, some tool changers support I/O signals that specify which gun is currently connected.

It is also possible to lock unconnected mechanical units from activation by specifying a connection relay and connect it to a digital input (DI).

Continues on next page

Defining a connection relay

Use RobotStudio to perform the following instructions. See *Operating manual - RobotStudio*.

For parameter descriptions, see *System parameters on page 135*.

	Action	Parameter
1	Select the topic *Motion* and type *Mechanical Unit* and define the name of the relay, or check the name if this is already defined.	• *Use Connection Relay*
2	Select the topic *Motion* and type *Relay* and select the named relay, if this is defined.	
3	If the named connection relay is not defined, a new relay must be created.	
4	Change the name of the newly created relay to the same as the *Use Connection Relay* parameter. Define an activation lock signal.	• *Name* • *Output Signal* • *Input Signal*

4.2.3 Defining relays

Overview

The additional drive unit can be activated via signals from the robot controller. When a module is activated, e.g. by choosing the module in the Jogging window on the FlexPendant, the output signal is automatically set. A check is made later that the corresponding input signal from the relay is set.

Defining activating relays

Define the input and output signals for all connected relays. Use RobotStudio to perform the following instructions. See *Operating manual - RobotStudio*. For parameter description, see *System parameters on page 135*.

	Action	Parameters
1	Restart the controller to check that the additional axes can be activated from the I/O window on the FlexPendant.	
2	Select the topic *Motion* and type *Relay* and define the following parameters.	• *Name* • *Output Signal* • *Input Signal*
3	Select the topic *Motion* and type *Mechanical Unit* and specify the name of the activation relay.	• *Use Activation Relay*
4	Perform a restart of the system.	

Defining brake relays

If the additional mechanical units are equipped with brakes, these will automatically be activated when the unit is deactivated or when the robot system assumes the MOTORS OFF state. They will also be activated when the axes have been stationary for a certain time (*Brake on Time*) in the MOTORS ON state. For a MultiMove system, the largest value of the *Brake on Time* parameters define when the brakes are activated.

Note

Mechanical units that share brake relay with the robot must not be deactivated. Set the system parameters *Deactivation Forbidden* and *Activate at Start Up* to Yes.

Defining the input and output signals

Define the input and output signals for all connected relays. Use RobotStudio to perform the following instructions. See *Operating manual - RobotStudio*. For parameter description, see *System parameters on page 135*.

	Action	Parameters
1	Restart the controller to check that the brakes can be activated from the I/O window on the FlexPendant.	
2	Select the configuration topic *Motion* and the type *Relay* and define the following parameters:	• *Name* • *Output Signal* • *Input Signal*

Continues on next page

	Action	Parameters
3	Select the topic *Motion* and the type *Mechanical Unit* and specify the name of the brake relay.	• *Use Brake Relay*
4	Perform a restart of the system.	

4.2.4 Defining brake behavior

Overview

If the axis has a brake, parameters which control brake behavior should be configured. If the axis is affected by gravity, more accurate parameter settings are necessary.

This section describes how to set up brake behavior for additional axes and non ABB robots.

Brake behavior at emergency stop

When an emergency stop has been ordered, it will take about 50 to 300 ms before the mechanical brake is physically active. Meanwhile there is ramp retardation by motor. After a certain period of time, the speed of the axis will determine whether or not the electrical torque brake is to be used along with the mechanical brake.

Good brake behavior is characterized by low oscillation in speed during retardation. *Test Signal Viewer* can be used to verify this; study *speed* (code 6) and *torque_ref* (code 9).

Measures must be taken to prevent the axis from dropping due to gravitation. This will happen if the motor torque is turned off before the mechanical brake has become physically active.

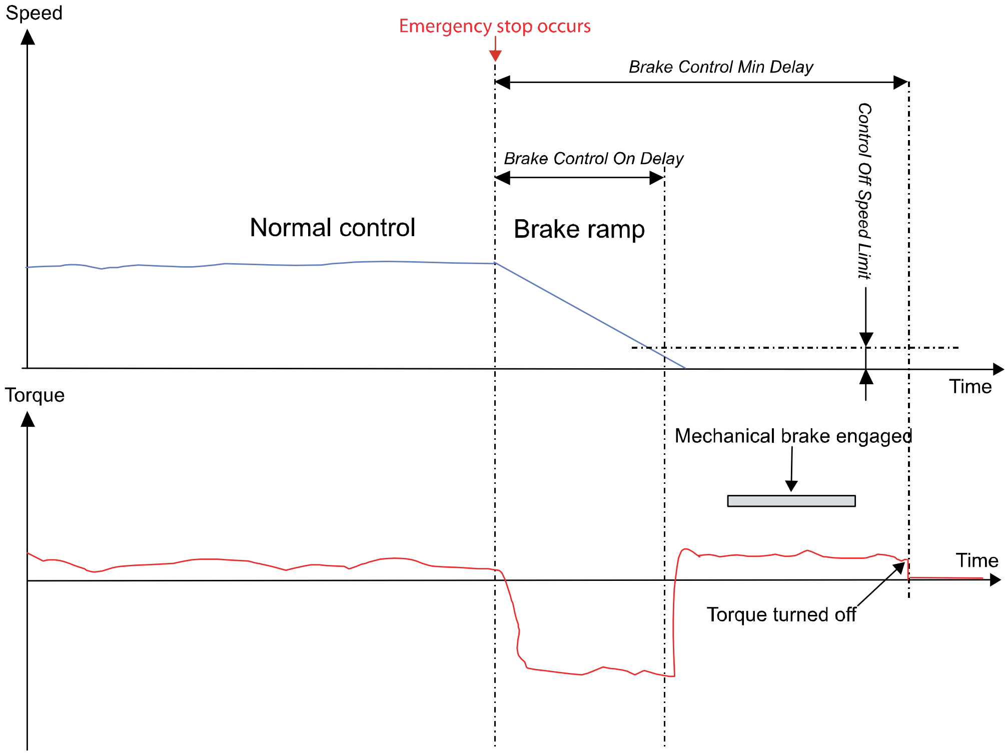
Continues on next page

Emergency brake algorithm

There are a few parameters which need to be configured to achieve good brake behavior. Understanding their role in the brake algorithm of the robot controller will simplify the task. The scenarios below illustrate how the brake parameters support the emergency brake algorithm of the robot controller.

Scenario 1

Axis almost at standstill after brake ramp, electrical torque brake not activated.



en0600003145

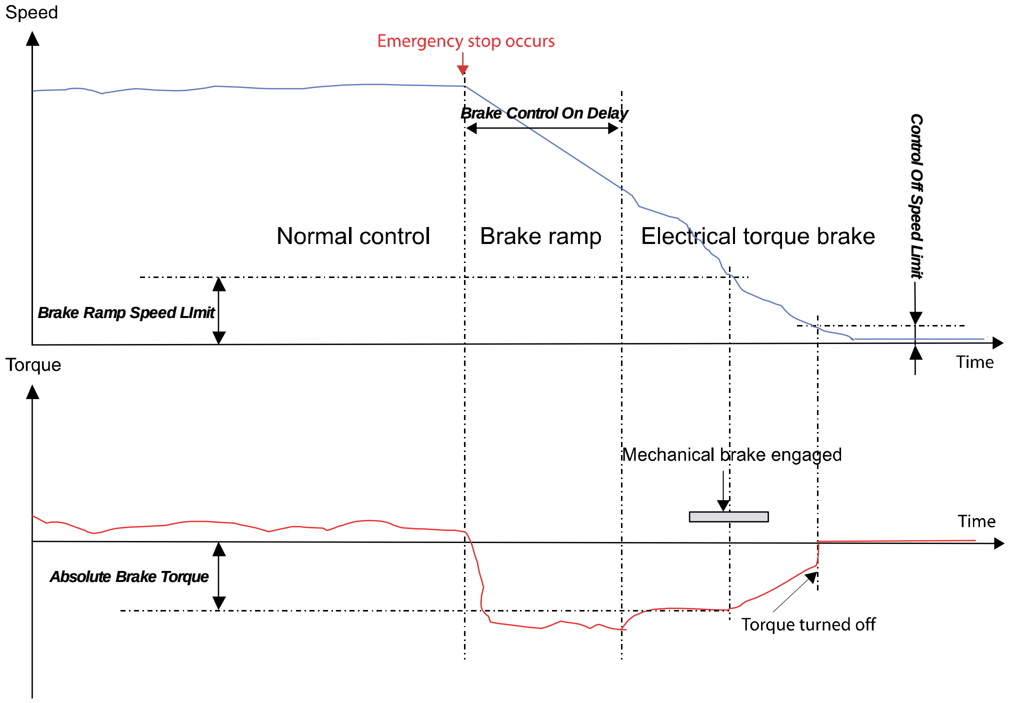
1 Emergency stops occurs. Immediately, ramp retardation by motor is started.

2 The axis has stopped when the *Brake Control On Delay* time has passed. The motor torque is used until the time *Brake Control Min Delay* has passed. This prevents the axis from falling before the mechanical brake is engaged.

Continues on next page

Scenario 2

Axis still moving after brake ramp, electrical torque brake activated.



en0600003146

1. Emergency stops occurs. Immediately, ramp retardation by motor is started.
2. As the axis is still moving (that is, speed exceeds *Control Off Speed Limit*) when *Brake Control On Delay* time has expired, the brake algorithm changes to Electrical torque brake.
3. The motor generates a brake torque specified by *Absolute Brake Torque*.
4. Torque reduction is started when the axis speed equals the value of *Brake Ramp Speed Limit*.
5. When the axis comes to a standstill the motor torque is turned off.

Defining brake parameters

Use RobotStudio or the FlexPendant to configure the brake parameters of the axis. See *Getting started with a simple brake configuration on page 85* for recommended start values for some of these parameters. All parameters belong to the type *Brake* in the configuration topic *Motion*.

	Action	Note
1	Define *Control Off Delay* in seconds.	Specifies for how long the control of the axis should be active. Time should be longer than it takes for the mechanical brake to become physically active, as to prevent the axis from dropping due to gravitation.
2	Define *Brake Control Min Delay* in seconds.	Should be set to the same value as *Control Off Delay*.

Continues on next page

	Action	Note
3	Define *Brake Control On Delay* in seconds.	Specifies the period of time during which retardation by motor is used. It should be set close or equal to the mechanical brake activation time, but must be long enough to damp mechanical oscillation. After the time has expired, the speed of the axis is measured against *Control Off Speed Limit*. If it is higher the electrical torque brake is activated.
4	Define *Absolute Brake Torque* in Nm.	Specifies max brake torque generated by the motor in the electrical torque brake phase. *Absolute Brake Torque* together with torque generated by the mechanical brake must not exceed max allowed torque for the arm, in order not to damage arm and gearbox.
5	Define *Brake Ramp Speed Limit* in rad/s.	Specifies the speed limit for torque reduction in the electrical torque brake phase and is typically set to zero.
6	Perform a restart of the controller.	

Getting started with a simple brake configuration

To facilitate brake configuration, this section provides initial values for some brake parameters. It is necessary, however, to adjust these parameter settings until good brake behavior is achieved.

The table below shows recommended initial values. All parameters belong to the type *Brake* in the configuration topic *Motion*.

Parameter	Start value
Control Off Delay	150% of mechanical brake activation time
Brake Control Min Delay	same as Control Off Delay
Brake Control On Delay	mechanical brake activation time
Absolute Brake Torque	0
Brake Ramp Speed LImit	0

Note

Do not modify *Control Off Speed Limit*! Its predefined ratio of max speed value defines zero speed.

4.2.5 Supervision

Overview

Supervision is used to avoid overload on the motors. To prevent misleading supervision errors due to influence forces, all axes with mutual influences shall be configured to the same influence group.

Description

If a manipulator mounted on a "Trackmotion" accelerates, the reaction (influence) forces affect the "Trackmotion". In the same way, if the "Trackmotion" accelerates, the manipulator is affected. Up to 10 different influence groups can be used (1-10). By default the manipulator belongs to the influence group no. 1.

Define influence groups

Define the parameter for supervision on the additional axes. Use RobotStudio to perform the following instructions. See *Operating manual - RobotStudio*. For parameter description see *System parameters on page 135*.

	Action	Parameters
1	Select the topic *Motion* and type *Supervision Type*.	
2	Select the additional axes to be grouped.	
3	Specify the following parameter. Default value: **0**.	*Influence Group*

4.2.6 Independent joint

Overview

With the options 610-1, Independent Axes, an additional axis (linear or rotating) can run independently of the other axes in the robot system.

Description

An axis is set in independent mode by executing an independent move instruction. Use the independent reset instruction to return to normal mode. Independent reset instruction can also be used in normal mode in order to change the logical position of the axis.

For more information about Independent Axes, see *Application manual - Controller software IRC5*.

Activate Independent Joint

Define the parameter for supervision on the additional axes. Use RobotStudio to perform the following instructions. See *Operating manual - RobotStudio*. For parameter description see section *System parameters on page 135*.

	Action	Parameter
1	Select the topic *Motion* and type *Arm*.	
2	Double click the axis to be activated	
3	Select the parameter *Independent Joint* in the appearing list.	
4	Set the Independent Joint to value **On.**	• *Independent Joint* • *Independent Upper Joint Bound* • *Independent Lower Joint Bound*

Defining transmission ratio

For external axes, the transmission ratio must be defined as normal with the parameter *Transmission Gear Ratio*, but also with its nominator and denominator values in order to get exact value (no rounding off). Use RobotStudio to perform the following instructions. See *Operating manual - RobotStudio*. For parameter description see section *System parameters on page 135*.

	Action	Parameter
1	Select the topic *Motion* and type *Transmission*.	
2	Specify the following parameters.	• *Transmission High Gear* • *Transmission Low Gear*

4.2.7 Soft servo

Overview

Soft servo can be activated for additional axes which are configured with *Lag Control Master 0*. The behavior of movements with the soft servo activated is described in the *Technical reference manual - RAPID Instructions, Functions and Data types*.

Description

There are four system parameters to consider when the soft servo is used for an additional axis. The parameter are set to default values.

Set soft servo parameters

Define the parameter for soft servo on the additional axes. Use RobotStudio to perform the following instructions. See *Operating manual - RobotStudio*. For parameter description see section, *System parameters on page 135*.

	Action	Parameters
1	Select the topic *Motion* and type *Lag Control Master 0*.	
2	Select the lag control master corresponding to the external axis.	
3	Select the desired parameter and change its value.	• *K Soft Max Factor* • *K Soft Min Factor* • *Kp/Kv Ratio Factor* • *Ramp Time*
4	Click **OK** to confirm.	
5	Perform a restart for the changes to take effect.	

4.2.8 Activate force gain control

Overview

Force gain control is used in cases when heavy load, high friction and low speed makes it difficult for an additional axis to reach its end point.

All axes that affect force gain control must be within a certain position range from the end point before forced gain control is enabled. This position range is also specified in the configuration topic *Motion*, type *Supervision*.

Description

When activating forced gain control for an additional axis, two types under *Motion* must be considered. Decide which axes should have forced gain control in *Lag Control Master 0*, and decide which axes should affect forced gain control in *Supervision*.

Set force gain control parameters

Define the parameter for forced gain control on the additional axes. Use RobotStudio to perform the following instructions. See *Operating manual - RobotStudio*. For parameter description see section, *System parameters on page 135*.

	Action	Parameter
1	Select the topic *Motion* and type *Lag Control Master 0*.	
2	Select the lag control master corresponding to the additional axis.	
3	Select the parameter to be changed.	• *Forced Control Active* • *Forced Factor for Kp* • *Forced Factor for Ki* • *Rise Time for Kp*
4	Press: **OK** to confirm.	
5	Perform a restart for the changes to take effect.	

Set supervision parameters

Define the parameter for supervision on the additional axes. Use RobotStudio to perform the following instructions. See *Operating manual - RobotStudio*. For parameter description see section, *System parameters on page 135*.

CAUTION

Do not change supervision for the robot axes. Changes on these values could affect the service interval cycles and impair its performance.

	Action	Parameter
1	Select the topic *Motion* and type *Supervision*.	
2	Select the supervision corresponding to the axis that should or should not affect forced gain control.	

Continues on next page

	Action	Parameter
3	Select the parameter to be changed.	• *Affects Forced Control* • *Forced on Position Limit* • *Forced off Position Limit*
4	Click **OK** to confirm.	
5	Perform a restart for the changes to take effect.	

4.2.9 Defining parameters for general kinematics

Overview

It is possible to use general kinematics for most manipulators. A set of template configuration files can be found in *...\utility\AdditionalAxis\DM1\GeneralKinematics.*, see *Template files on page 22*.

Note

Definition is not possible via the FlexPendant or RobotStudio. PC editing of the configuration files is necessary.

General kinematics for robots

The following needs to be defined.

Type	Description
ROBOT_TYPE	• base_pose_rot_u0, base_pose_rot_u1, base_pose_rot_u2, base_pose_rot_u3 (Rotation between user defined robot base and internal base according to Denavit - Hartenberg definition). • no_of_joints = highest joint number • type **GEN_KIN**
TRANSMISSION	For each arm of the additional robot in question. • **rotating_move** if rotating axes, exclude otherwise
ARM_TYPE	For each arm of the external robot in question. • **length** • **offset_x** • **offset_y** • **theta_home_position** • **offset_z** • **attitude** For information about the parameters, see *Arm Type on page 138*.

4.3 Coordinated axes

4.3.1 About coordinated axes

Additional axes, general

All additional axes are handled in mechanical units. This means that before an additional axis may be moved, the mechanical unit to which it belongs, must be activated. Within a mechanical unit, the different axes will be given a logical name, from *a* to *f*. In the system parameters, these logical axes will be connected to the additional axes joints. For each joint a motor and a drive unit is defined. Different joints may share the same motor and drive unit.

Two or more mechanical units may be activated at the same time, as long as they do not have the same logical axes defined in their set of additional axes. However, two or more mechanical units may have the same logical axes, if they are not activated simultaneously. Two or more mechanical units may not be activated at the same time, if they share one or more drive units, even if they use separate logical axes. This means that two logical axes, each belonging to different mechanical units, may control the same drive unit, but not at the same time.

Coordination

A mechanical unit may be coordinated or not coordinated with the robot movements. If it is not coordinated, each axis will be moved independent of the robot movements, e.g. when jogging, only the separate axis will move. However during program execution, the additional axes will be synchronized to the robot movement, in such a way that both movements will be completed in the same time.

If the mechanical unit is coordinated, the TCP velocity in the object coordinate system, will be the programmed velocity irrespective of the movements of the additional axes. Two types of coordination categories exist. The first category of coordination is when the robot base is moved, e.g. the coordination to a gantry or track movement. This means that the robot is mounted on a gantry or a track, and may be moved along these axes. The world and user/object coordinate systems, however, will be fixed in the room, and the robot movements in these coordinate systems will be independent of simultaneous gantry or track movements. This coordination is automatically active, if the mechanical unit with the track motion is active.

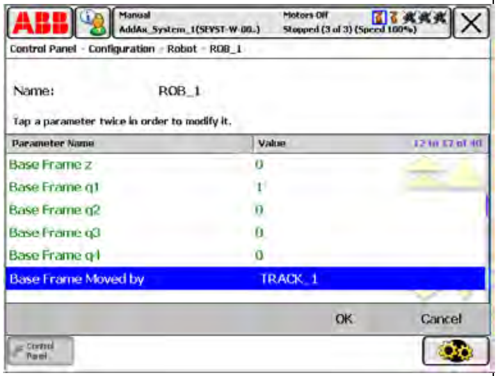
The second coordination category, is when the robot movements are coordinated to the movements of a user frame connected to a mechanical unit. E.g. a user frame may be placed on a turntable and connected to its movements. An ordinary work object may be used for this purpose, if it is marked with the name of the mechanical unit to be connected to, and that it should be moveable. The coordination will be active if the mechanical unit is active, and the coordinated work object is active. When such a coordinated work object is used, in jogging or in a move instruction, the data in the `uframe` component will be ignored and the location of the user coordinate system will only depend on the movements of the mechanical unit. However the `oframe` component will still work giving an object frame related to the user frame and also the displacement frame may be used.

4.3.2 Coordinated track motion

4.3.2.1 How to get started with a coordinated track motion

Coordination procedure

In the checklist below, the steps required to coordinate track motion are described. In each step, there may be a reference to another chapter in this manual, where more details of the specific actions to be taken will be found.

	Action	Info/illustration
1	Make sure the system parameter *Mechanics* in the type *Single Type* is set to TRACK.	
2	Calibrate the robot and the track motion, i.e. the zero position of the measuring system for both robot and track must be carefully determined.	See section *Calibrating* in *Operating manual - IRC5 with FlexPendant*.
3	Define the base frame of the robot. This defines the robot base frame relative to the world frame. The procedure is necessary only if the world frame is separate from the robot base frame. Please observe that the track must be in its calibration position when the robot base frame is defined.	See section *4 points XZ calibration* in *Operating manual - IRC5 with FlexPendant*.
4	Define the base frame of the track. This defines the rotation of the robot base relative to the track.	See *Defining the base frame for a track motion on page 95*.
5	Activate the base frame coordination by setting the system parameter *Base frame moved by* (topic *Motion* and type *Robot*) for the robot to the name of the track.	 Manual AddAx_System_1(SEYST-W-00..) Motors Off Stopped (3 of 3) (Speed 100%) Control Panel - Configuration - Robot - ROB_1 Name: ROB_1 Tap a parameter twice in order to modify it. Parameter Name / Value / 12 to 17 of 40 Base Frame z 0 Base Frame q1 1 Base Frame q2 0 Base Frame q3 0 Base Frame q4 0 Base Frame Moved by TRACK_1 OK Cancel Control Panel en0600002756 ⚠ **WARNING** The combined speed of the track motion and the robot must not exceed 250 mm/s in manual mode. This can only be ensured if the correct mechanical unit (the track motion) is set in the parameter *Base frame moved by* for the robot.
6	Create a backup of the system by tapping **ABB menu - Backup and Restore -Backup Current system.**	See *Back up and restore systems* in *Operating manual - IRC5 with FlexPendant*.

Continues on next page

	Action	Info/illustration
7	Activate the track unit in the jogging window and check that the coordination is working satisfactorily. This may be done by choosing **World** or **Work Object** in the field **Coordinate System** and then jogging the track axis. The robot TCP should not move, but be fixed relative to the object coordinate system.	

Note

If the robot base frame is rotated after the calibration of the track base frame, a new base frame calibration of the robot has to be done and also a new baseframe calibration of the track.

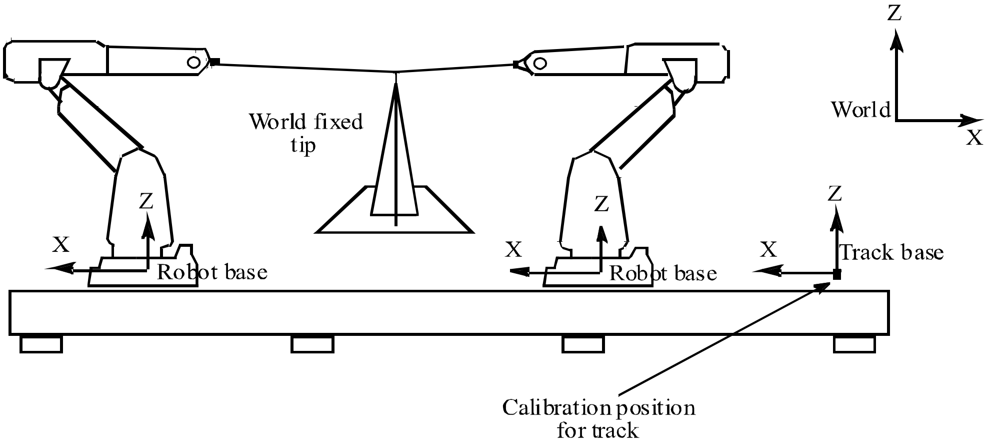
4.3.2.2 Defining the base frame for a track motion

Prerequisites

To make coordinated track motion possible it is necessary to define the base frame of the track. This frame is located in the calibration position of the track (see illustration below).

For the definition of a track base frame you need a world fixed tip within the robot's working range. The calibration procedure consists of a number of positionings of the TCP to the reference point (world fixed tip).

Please note that before the base frame of the track may be defined, the base frame of the robot must be defined with the track in the calibration position, that is robot base frame identical with track base frame.

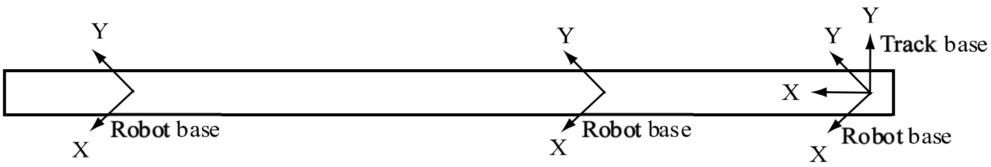


en0600002745

Definitions for track base coordinate system

The track's base coordinate system has its origin in the robot's base when the track is in its calibration position. The x direction is pointing along the linear track path and the z axis of the track's coordinate system is parallel with the z axis of the robot's base coordinate system.

The illustration below shows an example of how the base systems are oriented for a specific robot mounting. In this case the robot is mounted on the track at an angle of 45 degrees.



en0600002757

Base frame definition procedure

	Action
1	Tap the **ABB menu - Calibration.**

Continues on next page

	Action
2	Select the mechanical unit for the track.
3	Select **Base Frame** and **3 points**.
4	Activate the track unit and run it to the calibration position, that is zero position should be displayed on the FlexPendant.
5	Select **Point1**.
6	Jog the robot as close as possible to the world fixed tip.
7	Modify the position by tapping **Modify Position**.
8	Move the robot along the track and repeat the steps above for the points **Point 2** and **Point 3**.
9	Press **OK** to calculate the base frame for the track.

Result

The result of the calculation is displayed (expressed in the world coordinate system). The following values are shown:

Listed values	Description
Method	Displays the selected calibration method.
Max error	The maximum error for one positioning.
Min error	The minimum error for one positioning.
Mean error	The accuracy of the robot positioning against the tip.
Cartesian X	The x coordinate for the base frame. (x, y, z is the same as for the robot base frame).
Cartesian Y	The y coordinate for the base frame.
Cartesian Z	The z coordinate for the base frame.
Quaternion 1-4	Orientation components for the base frame.

If the estimated error is acceptable, press **OK** to confirm the new track base frame.

If the estimated error is unacceptable, press **Cancel** to redefine the calibration.

4.3.3 Coordinated positioners

4.3.3.1 How to get started with a coordinated (moveable) user coordinate system

Coordination procedure

In the checklist below, the steps required to coordinate a user coordinate system are described. In each step, there may be a reference to another chapter in this manual, where more details of the specific actions to be taken will be found.

	Action	Information
1	Calibrate the robot and the positioner, i.e. the zero position of the measuring system for both robot and positioner must be carefully determined.	See section *Calibrating* in *Operating manual - IRC5 with FlexPendant*.
2	Define the base frame of the robot.	See *4 points XZ calibration* in *Operating manual - IRC5 with FlexPendant*.
3	Define the user frame of the positioner.	See *Defining the user frame for a rotational single axis on page 98* or *Defining the user frame for a multi axes positioner on page 101*.
4	Create a backup of the system by tapping **ABB - Backup and Restore -Backup Current system**.	See *Back up and restore systems* in *Operating manual - IRC5 with FlexPendant*.
5	Create a new work object data and give it a name, e.g. wobj_turntable. In this work object, change the component `ufprog` to FALSE, indicating that the user object should be connected to a moveable mechanical unit. Also change the component `ufmec` to the name of the positioner (e.g. STN_1).	See section *Creating a work object* in *Operating manual - IRC5 with FlexPendant*.
6	If you want the object frame to be displaced relative to the user frame, you may write the displacement in the x, y, z values of the `oframe` component of the work object.	For more information about the object frame, see *Operating manual - IRC5 with FlexPendant*, section *What is a work object* and section *Defining the work object coordinate system*.
7	Activate the positioner in the jogging window and check that the coordination is working satisfactorily. This can be done by: • selecting **Work Object** in the field **Coordinate system** • selecting your work object, e.g. wobj_turntable, in the field **Work object** When jogging the positioner, the robot TCP should also move, following the moveable object coordinate system.	

Tip

When programming, it is important to have the coordinated work object, in this case wobj_turntable, programmed as an argument in each move instruction. This will be automatically added to the move instruction, if the work object is activated in the jogging window before starting the programming.

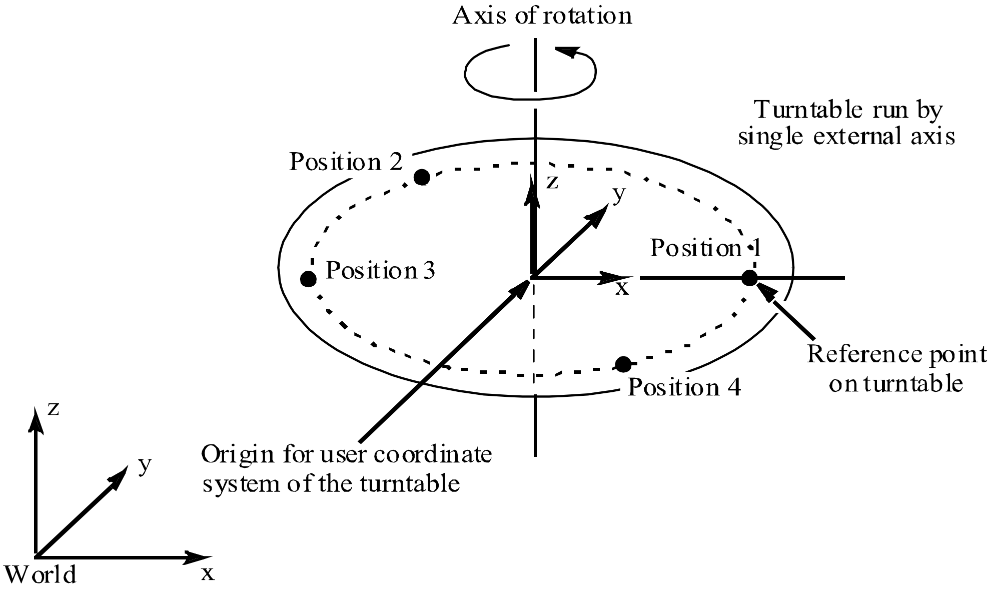
4.3.3.2 Defining the user frame for a rotational single axis

About defining the user frame

This method will define the location of the user coordinate system of a rotational single axis positioner, relative to the world coordinate system. As it is a single axis the base frame and user frame will coincide. This user coordinate system should be used when a coordinated work object is used.

Prerequisites

The definition of a user frame for a rotational additional axis requires that the turntable on the additional axis has a marked reference point. The calibration procedure consists of a number of positionings for the robot's TCP on the reference point when the turntable is rotated to different angles (see illustration below).

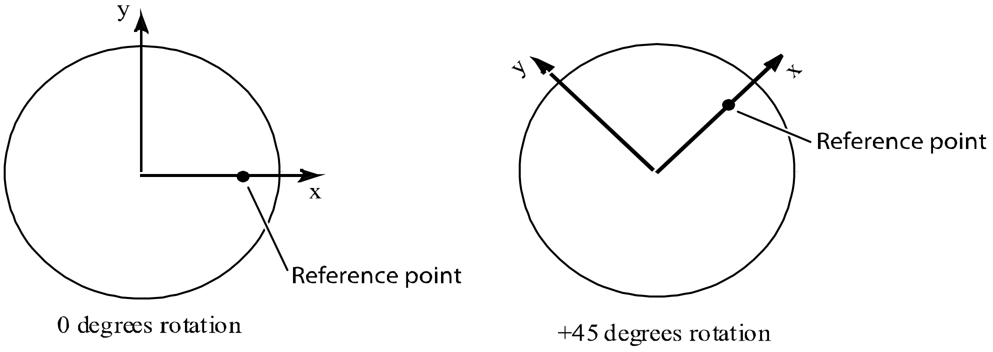

Axis of rotation
Turntable run by single external axis
Position 2
z
y
Position 1
Position 3
x
Reference point on turntable
Position 4
z
y
Origin for user coordinate system of the turntable
World
x

en0600002760

Continues on next page

Position and directions of the user frame

The user coordinate system for the rotational axis has its origin in the centre of the turntable. The z direction coincides with the axis of rotation and the x axis goes through the reference point. The illustration below shows the user coordinate system for two different positionings of the turntable (turntable seen from above).



en0600002761

Note

If it is intended to simulate the positioner in RobotStudio, it is recommended to define the user coordinate system of the rotational axis so that it coincides with the user coordinate system of the RobotStudio model.

User frame definition procedure

	Action
1	Tap the **ABB menu - Calibration.**
2	Select the rotational single axis positioner.
3	Select **Base Frame.**
4	Select the method **4 Point Z.**
5	If you have a MultiMove system, select which robot to use for the calibration. For non-MultiMove systems, go directly to the next step.
6	Select **Point 1**.
7	Jog the robot as close as possible to the reference point.
8	Modify the position by tapping **Modify Position.**
9	Move the rotational positioner to a new positive position and point out the new position with the robot.
10	Repeat the steps 6-9 for the points **Point 2**, **Point 3** and **Point 4.**
11	Jog the robot to a position where the tool center point (TCP) touches an imaginary extension of the desired positive z axis. In this case, this point should be along the rotational axis of the turntable (above the turntable). This is only to define the positive direction of the z axis. It is not used to increase the accuracy of the calibration. The exact direction of the z axis is defined as the normal of the xy plane.
12	Select **Elongator Point Z** and tap **Modify Position.**

Continues on next page

	Action
13	If you want, you can save the entered calibration data to a file. Tap **Positions** and then **Save**. Enter the name of the file and then tap **OK**. To restore this calibration, the file can be loaded from **Positions** - **Load**, instead of performing steps 6-12.
14	Press **OK** to calculate the user frame for the positioner.

Result

The result of the calculation is displayed (expressed in the world coordinate system). The following values are shown:

Listed values	Description
Method	Displays the selected calibration method.
Max error	The maximum error for one positioning.
Min error	The minimum error for one positioning.
Mean error	The accuracy of the robot positioning against the tip.
Cartesian X - Z	The x, y, z coordinates for the user frame.
Quarternion 1-4	Orientation components for the user frame.

If the estimated error is acceptable, press **OK** to confirm the new user frame.

If the estimated error is unacceptable, press **Cancel** to redefine the calibration.

4.3.3.3 Defining the user frame for a multi axes positioner

Parameter file required

It is possible to define positioners with more than one axis. To achieve the best possible performance from such a positioner, a set of data, describing its kinematic and dynamic properties (among other things), must be defined. This data cannot be defined in the system parameters via the FlexPendant or RobotStudio, but must be read from a parameter file. If no file was supplied with the manipulator, the manipulator cannot be coordinated with the robot. It can, however, be defined as a number of separate external axes.

Differences between one and multi axes positioner

The principles for defining a user frame for a multi axes positioner are the same as for a one axis positioner, see *Defining the user frame for a rotational single axis on page 98*. However, note that the axis must be moved in positive direction (see step 9 below).

For a positioner with more than one axis, a 4 point calibration is performed for each axis.

Note

If it is intended to simulate the positioner in RobotStudio, it is recommended to define the user coordinate systems of the rotational axes so that they coincide with the user coordinate systems of the RobotStudio model.

Define number of axes

The number of axes belonging to the positioner must be defined in the configuration file before coordinated motions are possible. The value should represent the number of axes connected in serial.

Parameter	Type	Description
no_of_error_model_joints	ROBOT_TYPE	Number of axes connected in serial belonging to the positioner. The maximum value is 6.

The parameter does not need to be set for ABB positioners.

User frame definition procedure

	Action
1	Tap the **ABB menu - Calibration**.
2	Select the multi axes positioner
3	Select **Base Frame**.
4	Select **4 Points for Axis 1**.
5	If you have a MultiMove system, select which robot to use for the calibration. For non-MultiMove systems, go directly to the next step.
6	Select **Point 1**.

Continues on next page

	Action
7	Jog the robot as close as possible to the reference point.
8	Modify the position by tapping **Modify Position**.
9	Move the first axis in positive direction to a new position (according to right hand rule).
10	Repeat the steps 6-9 for the points **Point 2**, **Point 3** and **Point 4**.
11	If you want, you can save the entered calibration data to a file. Tap **Positions** and then **Save**. Enter the name of the file and then tap **OK**. To restore this calibration, the file can be loaded from **Positions** - **Load**, instead of performing steps 6-10.
12	Select **4 Points for Axis 2** and repeat step 5-11 for the other axes belonging to the positioner.
13	Press **OK** to calculate the user frame for the positioner.

Result

The result of the calculation is displayed (expressed in the world coordinate system). The following values are shown:

Listed values	Description
Method	Displays the selected calibration method.
Max error	The maximum error for one positioning.
Min error	The minimum error for one positioning.
Mean error	The accuracy of the robot positioning against the tip.
Cartesian X - Z	The x, y, z coordinates for the user frame.
Quarternion 1-4	Orientation components for the user frame.

If the estimated error is acceptable, press **OK** to confirm the new user frame.

If the estimated error is unacceptable, press **Cancel** to redefine the calibration.

Note

When defining a work object for a coordinated motion, the user frame part of the work object is left empty (unit frame). Instead the user part is computed when needed using the kinematic model and the joint position for the mechanical unit.

5 Commutation

5.1 Commutate the motor

Overview

This chapter describes how to use the call service routine commutation, so the additional motor runs properly.

The service routine commutation is used to:

- Find a commutation value for a synchronous permanent magnet motor.
- Check motor phase order
- Verify the pole pair parameter value is correctly typed in.
- Check resolver connection

How to install a new motor

	Action	Note
1	Set the motor in safe mode by changing the system parameter *Current Vector On* (topic *Motion*, type *Drive System*) to Yes.	*Set the motor in safe mode/normal mode on page 103*.
2	Start the service routine Commutation.	See *Operating manual - IRC5 with FlexPendant* section *Programming and testing - Running a service routine*.
3	Check motor phase order connections.	*Check motor phase connections order on page 104*.
4	Check resolver connection.	*Check resolver connections on page 104*.
5	Move the motor to commutation position. • For the pre-Commutated motor: Check the motor phase connections. • For the none commutated motor: Commutate the motor by updating the commutation offset.	*Check the motor phase connections on page 104*. *Update commutation offset on page 104*.
6	The commutation is now finished and the motor is ready to use. When exiting, the program ask if the motor is to be set in normal mode. The motor can also be set to normal mode by changing the parameter *Current Vector On* to No.	See *Set the motor in safe mode/normal mode on page 103*.

Set the motor in safe mode/normal mode

The system parameters can be changed in the FlexPendant or in RobotStudio. On the FlexPendant tap **Control panel**/**Configuration**/**Topics**/**Motion**/**Drive System**. Then change the parameter *Current Vector On* to Yes or No in the *Drive System*. If the motor is to be set in safe mode set the parameter to Yes. If the motor is to be set in normal mode set the parameter to No.

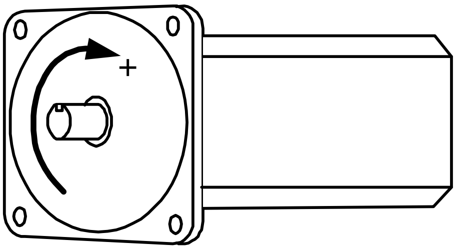
Continues on next page

CAUTION

If the motor is not properly installed, it can run away and destroy itself or other equipment. To avoid this set the motor in safe mode.

Check motor phase connections order

By stepping the motor in positive direction from the service routine, the motor shaft shall turn in counter clockwise direction, if the shaft is seen from the resolver side and clockwise from the drive shaft side.



xx0400001171

If the motor is turning in the wrong direction then the motor phases has been swapped. Try changing RST to SRT, RTS or TSR.

Check the pole pair parameter

Check if the pole pair parameter is loaded with the correct value by stepping the motor from the service routine. The motor shall turn 1/16 of a revolution for every step command.

Check resolver connections

From the service routine step the motor in positive direction. The resolver is connected correctly if the motor angle in the jogging window is increasing. Otherwise check the wiring of the resolver.

Check the motor phase connections

Step through the commutation angles to make sure that the cables are connected to the right phase. For best result commutate with a free mounted motor. There are a numbers of correct commutation angles (same as pole_pair parameter). If the difference is a number of 6.283185/pole_pair values - the commutation is ok. Otherwise all motor phases shall be moved one step forward or backward (same order! RST -> STR or TRS). A commutation value set by the motor manufacturer is normally more accurate than a value updated with this method.

Update commutation offset

To get a good commutation position the motor must not be affected by gravity or large friction from equipment connected to the motor. For best result commutate with a free mounted motor.

When the motor is aligned, the resolver commutation parameter can be set. When the parameter is set the database is also updated.

6 Tuning

6.1 Tuning the commutation offset

General

Before using an additional axis, you must tune the motors commutation offset. This requires that you connect a DC power source between two nodes and then measure the position of the motor.

Note

ABB motors are precommutated with the commutation value 1.5708. Therefore, an ABB motor does not require tuning of the commutation offset.

Prerequisites

The motor must comply with the specifications in *Motors on page 178*. The resolver must comply with the specifications in *Resolvers on page 184*.

Required material

This is a list of what you need to perform the tuning:

Material	Description
PC with Test Signal Viewer	The software Test Signal Viewer is delivered on the RobotWare DVD.
Power supply	24 V (DC). The power supply should be equipped with a relay that trips at short circuit. Otherwise a fuse will burn every time the power is applied. Check the motor data to see the current required from the power supply.
2 cable sets	Cables to brake release and motor phase. Each cable set includes one plus and one minus cable.
Motor documentation	Motor data sheet and electrical connection drawing.

Measuring procedure

This procedure describes how to measure the commutation position of a motor.

	Action
1	Deactivate the axis of the motor you want to tune.
2	Switch off the controller.
3	Disconnect the power cable to the motor.
4	Disconnect the motor from the gear (or in some other way make sure the motor is not affected by external torque and friction).
5	If the motor is using a brake, release it by connecting the power supply to the contact pins for the brake release. See motor specifications for max brake current, which contact are for the brake release and the polarity of the contacts (if any).
6	Ensure that the brake is released by manually turning the motor.

Continues on next page

	Action
7	Connect the power supply with the plus cable to the phase S (V) and the minus cable (0 V) to the to the phase T (W). A short pulse is enough to move the motor to its commutation position. Disconnect the power after the voltage pulse.
8	Connect the power to give another voltage pulse to the motor. If the motor is already in its commutation position it should not move this time.
9	Disconnect the power supply from the brake release, so that the motor brake is on.
10	Reconnect the power cable from the drive module to the motor.
11	Activate the axis. Do not move any mechanical unit.
12	Start the controller.
13	Configure Test Signal Viewer, selecting mechanical unit and the signal *Resolver_angle*. Zoom in on the signal so you can read at least 2 decimals. Note that the number of commutation positions are equal to the number of pole pairs. E.g. a motor with 2 pole pairs have 2 possible values for this measurement. It does not matter which of the commutation points you are measuring.
14	Set the measured value to the parameter *Commutator Offset* in the type *Motor Calibration*.
15	Reconnect the motor to the gear.

6.2 Tuning

6.2.1 Introduction

Overview

The servo control parameters can be adjusted (tuned) to achieve the best possible motion performance.

Tuning with TuneMaster

The recommended way to perform the tuning is by using the software TuneMaster, which can be installed from the RobotWare DVD (`Tools\TuneMaster`).

Documentation for the TuneMaster program is available as online-help in the program. Follow the descriptions in the online-help and skip the rest of this section.

Tuning manually

If you want to do the tuning manually, without the help of TuneMaster, follow the instructions in this section. This section contains a complete description of how to tune the axes and is divided into the following sections:

- *Tuning of axes, complete procedure on page 110* details the complete procedure of tuning the axes, including necessary preparations and references to more detailed instructions of setting tuned values for the different parameters.
- Separate procedures that detail how to set the respective tuning parameters.

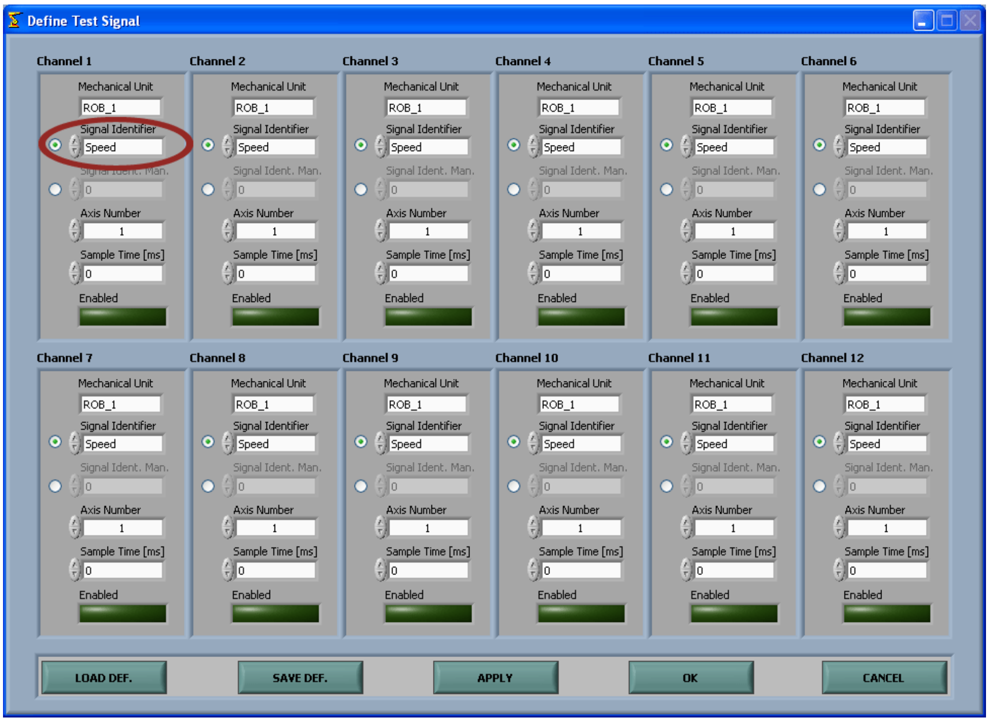
6.2.2 Defining signals in Test Signal Viewer

Defining test signals

Test Signal Viewer is used during tuning of the parameters. This section shows the signals that may be used, and how these are defined from the menu Commands - Define Test Signal.

Signal Identifier

Some of the signals may be defined from **Signal Identifier**, see the figure below:



	Channel 1	Channel 2	Channel 3	Channel 4	Channel 5	Channel 6
Mechanical Unit	ROB_1	ROB_1	ROB_1	ROB_1	ROB_1	ROB_1
Signal Identifier	● Speed	● Speed	● Speed	● Speed	● Speed	● Speed
Signal Ident. Man.	○ 0	○ 0	○ 0	○ 0	○ 0	○ 0
Axis Number	1	1	1	1	1	1
Sample Time [ms]	0	0	0	0	0	0
Enabled						

	Channel 7	Channel 8	Channel 9	Channel 10	Channel 11	Channel 12
Mechanical Unit	ROB_1	ROB_1	ROB_1	ROB_1	ROB_1	ROB_1
Signal Identifier	● Speed	● Speed	● Speed	● Speed	● Speed	● Speed
Signal Ident. Man.	○ 0	○ 0	○ 0	○ 0	○ 0	○ 0
Axis Number	1	1	1	1	1	1
Sample Time [ms]	0	0	0	0	0	0
Enabled						

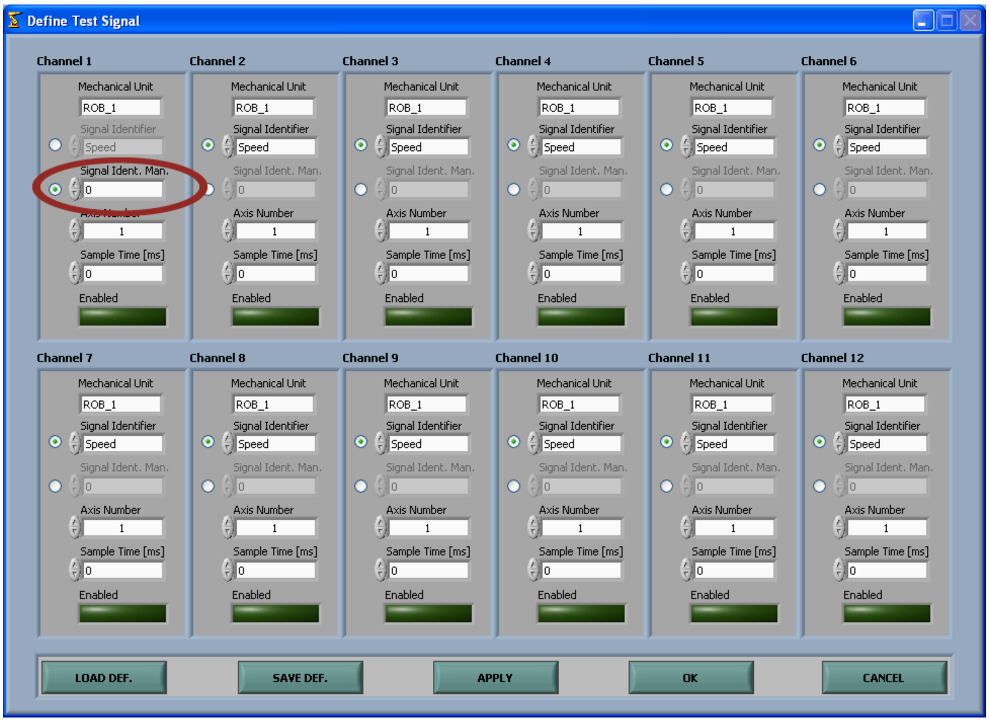
en0800000474

Signal number	Signal name
6	Speed
9	Torque Ref.

Continues on next page

Signal Identifier, Manual

Some of the signals may be defined manually by entering numbers in **Signal Ident. Man.**, see the figure below:



	Channel 1	Channel 2	Channel 3	Channel 4	Channel 5	Channel 6
Mechanical Unit	ROB_1	ROB_1	ROB_1	ROB_1	ROB_1	ROB_1
Signal Identifier	○ Speed	● Speed	● Speed	● Speed	● Speed	● Speed
Signal Ident. Man.	● 0	○ 0	○ 0	○ 0	○ 0	○ 0
Axis Number	1	1	1	1	1	1
Sample Time [ms]	0	0	0	0	0	0
Enabled						

	Channel 7	Channel 8	Channel 9	Channel 10	Channel 11	Channel 12
Mechanical Unit	ROB_1	ROB_1	ROB_1	ROB_1	ROB_1	ROB_1
Signal Identifier	● Speed	● Speed	● Speed	● Speed	● Speed	● Speed
Signal Ident. Man.	○ 0	○ 0	○ 0	○ 0	○ 0	○ 0
Axis Number	1	1	1	1	1	1
Sample Time [ms]	0	0	0	0	0	0
Enabled						

en0800000475

Signal number	Signal name
6	Speed
9	Torque Ref.
55	Positive Torque Limit
56	Negative Torque Limit
57	Torque Feed Forward

6.2.3 Tuning of axes, complete procedure

General

This section details the complete procedure for tuning of the axes, which is done in order to achieve the best possible motion performance of the robot system.

⚠ **WARNING**

The system is unstable and therefore dangerous during the tuning process, since bad parameters or parameter combinations may be used! The safety procedures of the robot system must be carefully followed throughout the tuning process.

How to set the parameter values

System parameters for tuning are preferably edited by using RobotStudio. They can however also be edited on the FlexPendant, and even directly in the configuration files by using a text editor. System parameters are listed in the section *System parameters on page 135*. More information, including parameter names in the configuration files, can be found in *Technical reference manual - System parameters*.

Preparations

The following preparations must be made before tuning a servomotor axis:

- Set up the Test Signal Viewer software according to the instructions in the pdf-file enclosed with the program.
- Make sure that the additional axis is commutated and calibrated. Any position may be defined as the calibration position.
- Select or use default tuning parameters so the axis may be jogged without stopping due to speed or torque supervision.

Tuning calibrated axes, overview

Tuning a calibrated axis means setting values for several parameters in the motion configuration type *Lag Control Master 0*. The procedure below is an overview of the complete tuning procedure and includes references to more detailed instructions of how to actually set each parameter.

	Action
1	Make the *Preparations on page 110*.
2	Use RobotStudio or the FlexPendant. Select the configuration topic *Motion* and the type *Lag Control Master 0*.
3	Set initially tuned values for the parameters *Kv*, *Kp* and *Ti*. This is further detailed in the section *Initial tuning of Kv, Kp and Ti on page 113*.

Continues on next page

	Action
4	Set a value for the parameter *FFW Mode* (feed forward mode) by choosing one of following modes: • **No**: This is the easiest configuration with no adjustments of the position lag. This mode is called **0** in the configuration file. • **Spd**: This is the recommended configuration. In this configuration the controller receives information about the desired speed of the axis. As a result, the position lag is considerably reduced compared to the **No** configuration. This mode is called **1** in the configuration file. • **Trq**: In this configuration the controller uses the desired speed and acceleration of the axis to calculate the desired motor torque. This requires knowledge of the mass moment of inertia of the axis, which must be supplied by the user. For this reason this configuration is more difficult to tune, specially for the axes affected by gravity. It is only recommended for experienced users. This mode is called **2** in the configuration file. The controller is driven by the position lag. This is the offset in time from the point of given command to the point of actual performance of the command. The lag may be adjusted with the different modes in the parameter *FFW Mode*. Depending on which mode is chosen for the *FFW Mode*, further tuning parameters can be set later on.
5	Set a value for the parameter *Inertia*, depending on the chosen *FFW Mode*: • *FFW Mode* = **No**: set *Inertia* to 0 • *FFW Mode* = **Spd**: set *Inertia* to 0 • *FFW Mode* = **Trq**: calculate and set *Inertia* according to the section *Specifying the inertia on page 119*.
6	If parameter *FFW Mode* is set to **Spd**, the following parameters are also available: • *Bandwidth*, should be left at its default value • *Delay*, should be left at its default value.
7	If parameter *FFW Mode* is set to **Trq**, the following parameters are also available: • *Bandwidth*, should be left at its default value. Adjustment is detailed in the section *Tuning of Bandwidth on page 120*. • *Delay*, should be left at its default value (0.004). In rare cases, increasing the value may reduce the speed overshot. • *Resonance frequency* (flexibility compensation filter), should initially be left at its default value. May be adjusted once the other parameters are set, if the *torque_ref* signal is oscillatory due to mechanical resonance. How to tune this parameter is detailed in the section *Tuning of resonance frequency on page 122*. • *Resonance damping* (flexibility compensation filter), should be left at its default value (0.01).
8	In the type *Acceleration Data*, set tuned values for *Nominal Acceleration* and *Nominal Deceleration* as detailed in the section *Tuning of Nominal Acceleration and Nominal Deceleration on page 123*.
9	In *Lag Control Master 0*, set finally tuned values for the parameters: • *Kv, Gain Speed Loop* • *Kp, Gain Position Loop* • *Ti Integration Time Speed Loop* This is further detailed in the section *Final tuning of Kp, Kv and Ti on page 125*.

Continues on next page

Tuning uncalibrated axes, overview

Tuning an uncalibrated axis means setting values for several parameters in the motion configuration type *Uncalibrated Control Master 0*. The procedure below is an overview of the complete tuning procedure and includes references to more detailed instructions of how to actually set each parameter.

	Action
1	Make the *Preparations on page 110*.
2	Use RobotStudio or the FlexPendant. Select the configuration topic *Motion* and the type *Uncalibrated Control Master 0*.
3	Set the initial tuning values of *Kv, Gain Speed Loop*, *Kp, Gain Position Loop*, and *Ti Integration Time Speed Loop*. Use tuned values, set during tuning of a calibrated axis. Use the values so that the axis is movable regardless of the position.
4	Set the tuning value of *Speed Max Uncalibrated*. Maximum speed for uncalibrated axis (rad/s on motor side).
5	Set the tuning value of *Deceleration Max Uncalibrated*. Maximum deceleration for uncalibrated axis (rad/s^2 on motor side). Recommended value: *Nominal Deceleration* * *Transmission Gear Ratio*. How to set the value for the nominal deceleration is detailed in the section *Tuning of Nominal Acceleration and Nominal Deceleration on page 123*.
6	Set the tuning value of *Acceleration Max Uncalibrated*. Maximum acceleration for uncalibrated axis (rad/s^2 on motor side). Recommended value: *Nominal Acceleration* * *Transm Gear Ratio*. How to set the value for the nominal acceleration is detailed in the section *Tuning of Nominal Acceleration and Nominal Deceleration on page 123*.
7	Set the final tuning values of *Kv, Gain Speed Loop*, *Kp, Gain Position Loop*, and *Ti Integration Time Speed Loop*, as detailed in the section *Final tuning of Kp, Kv and Ti on page 125*.

6.2.4 Initial tuning of Kv, Kp and Ti

General

This section details how to make the initial tuning of the parameters *Kv, Gain Speed Loop*, *Kp, Gain Position Loop* and *Ti Integration Time Speed Loop*.

The general strategy is to tune Kv first while keeping Kp constant and without integral effect (Ti is set to a high value, e.g. 10), then tune Kp to its maximum value without vibration/oscillation; finally tune Ti and other parameters.

Note

Check that the additional axes motor file contains the correct motor data.

Parameter description

The parameters *Kv, Gain Speed Loop*, *Kp, Gain Position Loop* and *Ti Integration Time Speed Loop* belongs to the type *Lag Control Master 0* and are further described in the section *Lag Control Master 0 on page 143*.

RAPID program

Program a back-and-forth motion of the axis. For the final tuning of the control parameters of the axis it is convenient to use the `TuneServo` command. Each procedure below includes an example of a RAPID program that can be used.

Initial tuning of Kv

The procedure below details how to make the initial tuning of the parameter *Kv, Gain Speed Loop*.

WARNING

The system is unstable and therefore dangerous during the tuning process, since bad parameters or parameter combinations may be used! The safety procedures of the robot system must be carefully followed throughout the tuning process.

	Action
1	Set the parameter *FFW Mode* to **No**.
2	Make following changes in the type *Lag Control Master 0*: • set the value of parameter *Kp, Gain Position Loop* to 5 • set the value of parameter *Ti Integration Time Speed Loop* to 10 (a big value in order to eliminate integral portion). Perform a restart of the controller for the changes to take effect.

Continues on next page

<table>
<tr><th></th><th>Action</th></tr>
<tr><td>3</td><td>

Increase the Kv value by 10% in each motion loop, until the axis starts to vibrate/oscillate or a clear vibration can be heard from the axis, either during motion or when stationary. The axis velocity supervision may also indicate speed failure.

The following RAPID program can be used to increase Kv by 10% of the default value set in the loaded configuration file:

```
MODULE Kv_tune
  PROC main()
    VAR num i;
    VAR num per_Kv;
    VAR num Kv;

    TuneReset;
    FOR i FROM 0 TO 40 DO
      per_Kv:=100+10*i;
      Kv:=1*per_Kv/100;
      TPErase;
      TPWrite "per_Kv = "\Num:=per_Kv;
      TPWrite "Kv = "\Num:=Kv;
      TuneServo M7DM1,1,100\Type:=TUNE_KP;
      TuneServo M7DM1,1,100\Type:=TUNE_TI;
      TuneServo M7DM1,1,per_Kv\Type:=TUNE_KV;
      MoveJ p1,v1000,z50,tool0;
      MoveJ p2,v500,z50,tool0;
      MoveJ p1,v1000,z50,tool0;
      WaitTime 1;
    ENDFOR
  ENDPROC
ENDMODULE
```

Note! The velocity data and test positions may be modified depending on the type of robot and axis to be tuned.

The RAPID instructions are described in the *Technical reference manual - RAPID Instructions, Functions and Data types*.

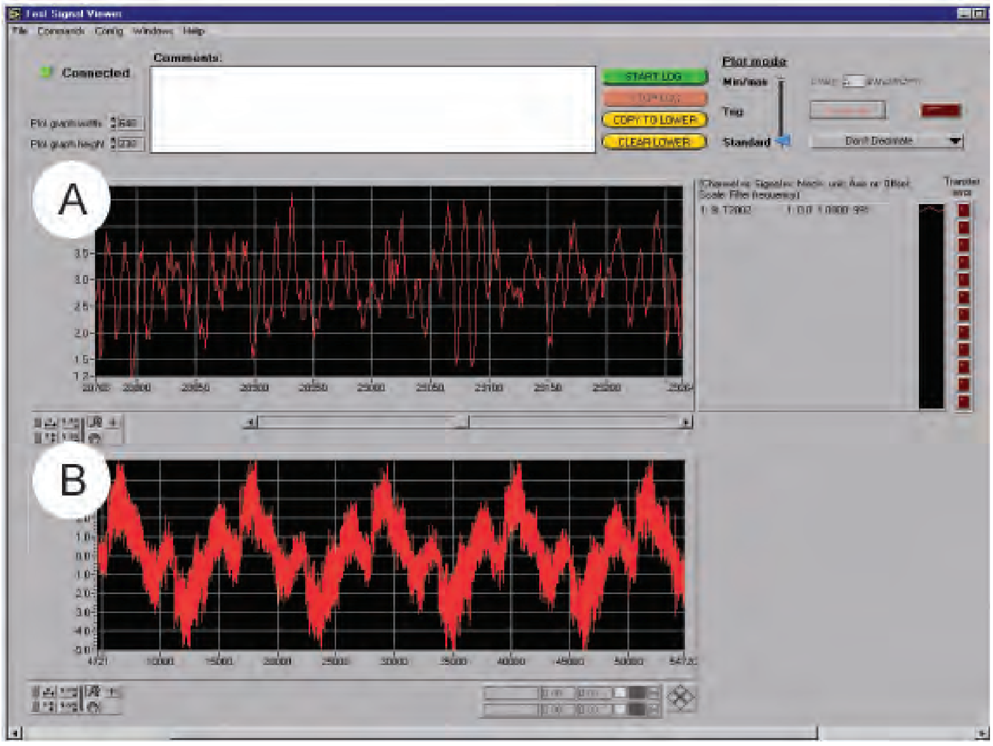
The *Torque Ref.* signal in the Test Signal Viewer may be used to evaluate the degree of vibration/oscillation. A typical plot of the signal is shown in the figure *Illustration, Torque Ref. plot on page 115*.

</td></tr>
<tr><td>4</td><td>Once an unstable point is reached, divide the current value of Kv by 2.</td></tr>
<tr><td>5</td><td>Enter the new value in the parameter <i>Kv, Gain Speed Loop</i>.</td></tr>
</table>

Continues on next page

Illustration, Torque Ref. plot

How to choose the *Torque Ref.* signal in Test Signal Viewer is detailed in section *Defining signals in Test Signal Viewer on page 108*.



xx0400000644

A	Normal plot
B	Unstable plot

Initial tuning of Kp

The procedure below details how to make the initial tuning of the parameter *Kp, Gain Position Loop*.

WARNING

The system is unstable and therefore dangerous during the tuning process, since bad parameters or parameter combinations may be used! The safety procedures of the robot system must be carefully followed throughout the tuning process.

	Action
1	Leave the initially tuned value of the parameter *Kv, Gain Speed Loop* and the earlier set default values of *Kp, Gain Position Loop* and *Ti Integration Time Speed Loop* unchanged.

Continues on next page

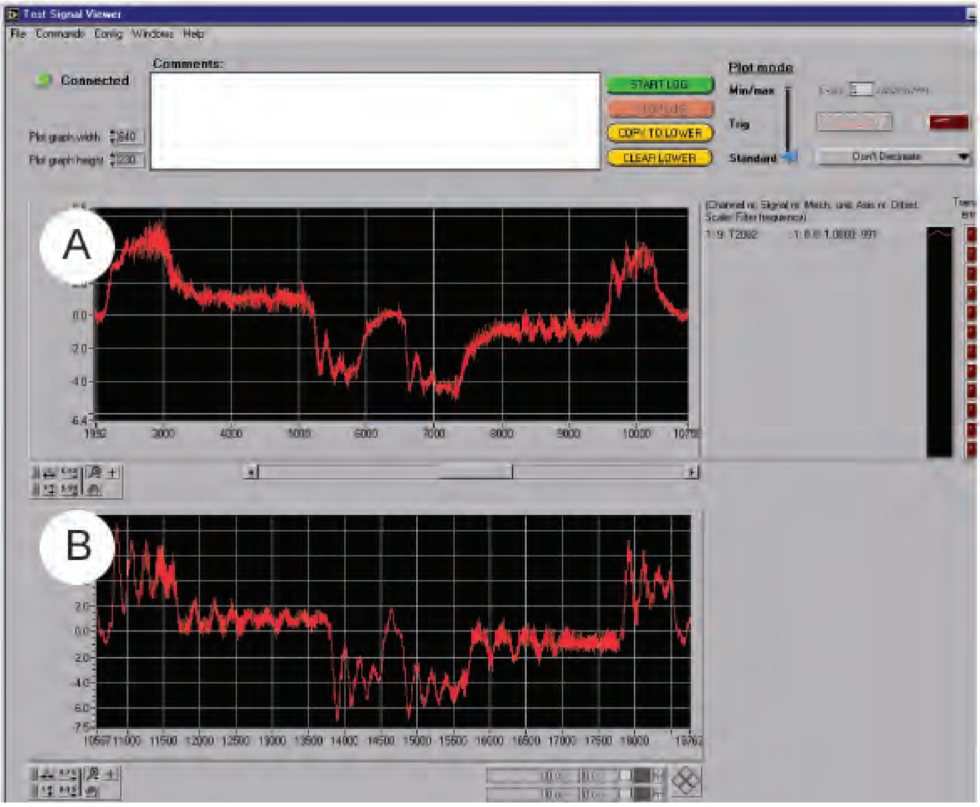
<table>
<tr><th></th><th>Action</th></tr>
<tr><td>2</td><td>Increase the Kp value by 10% in each motion loop, until the first signs of overshooting are observed in the velocity plot.
The following RAPID program may be used to increase Kp by 10% of the default value set in the loaded configuration file:
<pre>
MODULE Kp_tune
  PROC main()
    VAR num i;
    VAR num per_Kp;
    VAR num Kp;

    TuneReset;
    FOR i FROM 0 TO 20 DO
      per_Kp:=100+10*i;
      Kp:=5*per_Kp/100;
      TPErase;
      TPWrite "per_Kp = "\Num:=per_Kp;
      TPWrite "Kp = "\Num:=Kp;
      TuneServo M7DM1,1,100\Type:=TUNE_KV;
      TuneServo M7DM1,1,100\Type:=TUNE_TI;
      TuneServo M7DM1,1,per_Kp\Type:=TUNE_KP;
      MoveJ p1,v1000,z50,tool0;
      MoveJ p2,v500,z50,tool0;
      MoveJ p1,v1000,z50,tool0;
      WaitTime 1;
    ENDFOR
  ENDPROC
ENDMODULE
</pre>
<b>Note!</b> The velocity data and test positions may be modified depending on the type of robot and axis to be tuned.
The RAPID instructions are described in the <i>Technical reference manual - RAPID Instructions, Functions and Data types</i>.
A typical plot with speed overshot in Test Signal Viewer is shown in the figure <i>Illustration, speed plot on page 117</i>.</td></tr>
<tr><td>3</td><td>Once an overshot is observed, subtract 1 from the current value of Kp.
If an overshooting is observed at a later time, the Kp must be reduced to an even lower value.</td></tr>
<tr><td>4</td><td>Enter this new value in the parameter <i>Kp, Gain Position Loop</i>.</td></tr>
</table>

Continues on next page

Illustration, speed plot

How to choose the speed signal in Test Signal Viewer is detailed in section *Defining signals in Test Signal Viewer on page 108*.



Plot graph width 640

Plot graph height 230

Channel nr. Signal nr. Mech. unit Axis nr. Offset Scale Filter frequency

1: 9: T2002 1: 6: 0: 1.0000: 991

xx0400000648

A	Normal plot
B	Overshoot

Initial tuning of Ti

The procedure below details how to make the initial tuning of the parameter *Ti Integration Time Speed Loop*.

WARNING

The system is unstable and therefore dangerous during the tuning process, since bad parameters or parameter combinations may be used! The safety procedures of the robot system must be carefully followed throughout the tuning process.

	Action
1	Leave the initially tuned values of parameters *Kv, Gain Speed Loop* and *Kp, Gain Position Loop* unchanged.
2	Set the value of *Ti Integration Time Speed Loop* to 1.0 as default.

Continues on next page

<table>
<tr><th></th><th>Action</th></tr>
<tr><td>3</td><td>

Reduce the Ti value by 10% in each motion loop, until an overshot is observed on the velocity profile in Test Signal Viewer.

Looking at the *Torque Ref.* signal can also help to determine the critical value of Ti.

The following RAPID program may be used to reduce the Ti value by 10% of the default value set in the loaded configuration file:

```
MODULE Ti_tune
  PROC main()
    VAR num i;
    VAR num per_Ti;
    VAR num Ti;

    TuneReset;
    FOR i FROM 0 TO 10 DO
      per_Ti:=100-10*i;
      Ti:=1*per_Ti/100;
      TPErase;
      TPWrite "per_Ti = "\Num:=per_Ti;
      TPWrite "Ti = "\Num:=Ti;
      TuneServo M7DM1,1,200\Type:=TUNE_KV;
      TuneServo M7DM1,1,250\Type:=TUNE_KP;
      TuneServo M7DM1,1,per_Ti\Type:=TUNE_TI;
      MoveJ p1,v1000,z50,tool0;
      MoveJ p2,v500,z50,tool0;
      MoveJ p1,v1000,z50,tool0;
      WaitTime 1;
    ENDFOR
  ENDPROC
ENDMODULE
```

Note! The velocity data and test positions may be modified depending on the type of robot and axis to be tuned.

The RAPID instructions are described in the *Technical reference manual - RAPID Instructions, Functions and Data types*.

</td></tr>
<tr><td>4</td><td>Once an overshot is observed, stop reducing and instead increase the tuned value by 5-10%, in order to remove the effect.</td></tr>
<tr><td>5</td><td>When the effect is removed, enter the new value in <i>Ti Integration Time Speed Loop</i>.</td></tr>
</table>

Result

The parameters *Kv, Gain Speed Loop*, *Kp, Gain Position Loop* and *Ti Integration Time Speed Loop* are now initially tuned. Continue with the tuning process as described in next section. The complete tuning procedure is detailed in section *Tuning of axes, complete procedure on page 110*.

6.2.5 Specifying the inertia

General

This section details how to specify the mass moment of inertia. (only recommended for experienced users).

Prerequisite

The parameter *FFW Mode* (in the type *Lag Control Master 0)* must be set to **Trq**, in order to have use of *Inertia*. If *FFW Mode* is set to **No** or **Spd**, the parameter *Inertia* is not used, but should be set to 0 anyway.

Specifying the inertia

The procedure below details how to calculate the mass moment of inertia and how to enter this value.

	Action
1	The inertia is given by: Inertia = $\text{Inertia}_{\text{axis}}$ / (*Transmission gear ratio*)^2 + $\text{Inertia}_{\text{motor}}$ + Inertia $_{\text{brake}}$
2	Enter this value for the parameter *Inertia* in the type *Lag Control Master 0*.

Result

The parameter *Inertia* is now tuned. Continue with the tuning procedure.

6.2.6 Tuning of Bandwidth

General

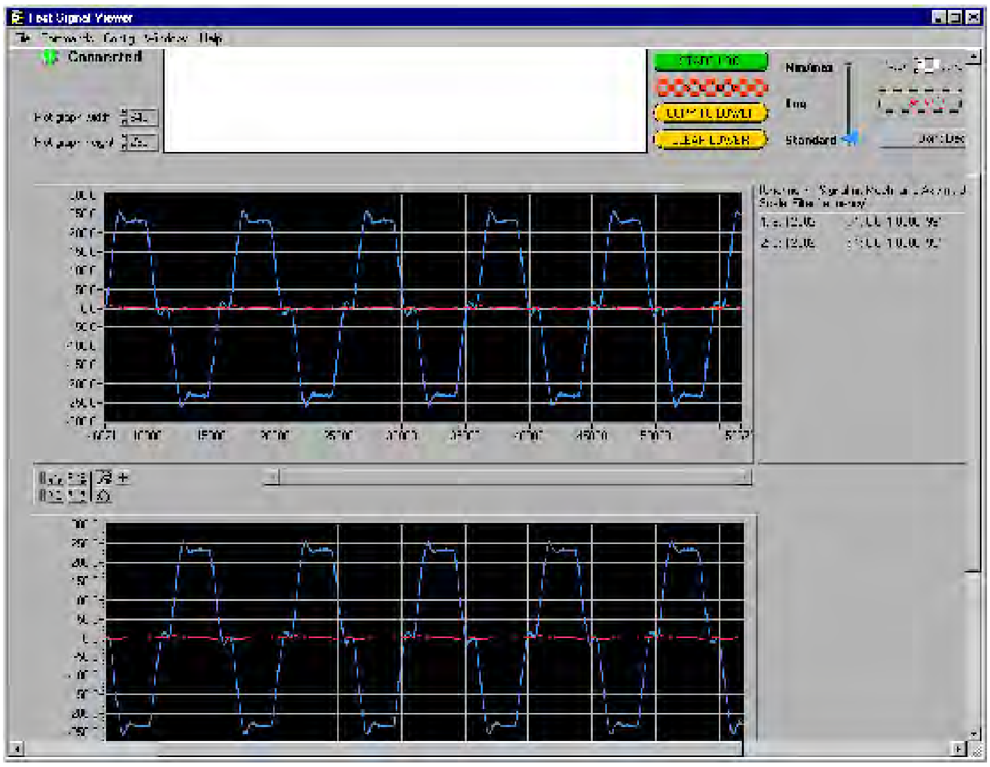
This section details how to set the tuned value for the parameter *Bandwidth*.

Parameter description

The parameter *Bandwidth* is found in the data group *Lag Control Master 0* and is further detailed in the section *Lag Control Master 0 on page 143*.

Illustration, speed overshot, bandwidth tuning

The figure below shows a typical plot in Test Signal Viewer when tuning the parameter *Bandwidth*.



xx0400000652

Tuning of bandwidth

The procedure below details how to tune the parameter *Bandwidth*.

	Action
1	Use properly tuned values for the parameters *Kv, Gain Speed Loop*, *Kp, Gain Position Loop* and *Ti Integration Time Speed Loop*.
2	Use the default value (25) for the parameter *Bandwidth*.

Continues on next page

	Action
3	In order to verify and tune the default *Bandwidth* value, run a motion program to see if the speed overshot occurs. If the overshoot is out of the tolerance, reduce the value of *Bandwidth* under the system resonance frequency. The system resonance frequency may not always be identified. Therefore the value may be varying several times and it may be necessary to perform a check of the speed overshoot and of the following error between command position and actual position. Normally, the bigger the *Bandwidth* value, the smaller the following error. A typical speed overshoot plot is shown in the figure *Illustration, speed overshot, bandwidth tuning on page 120*.

Result

The parameter *Bandwidth* is now initially tuned. Continue with the tuning process described in the next section, and in section *Tuning of axes, complete procedure on page 110*.

6.2.7 Tuning of resonance frequency

General

This section details how to set the value of the system parameter *Df* (resonance frequency).

Parameter description

The parameter *Df* is found in the type *Lag Control Master 0* and is further detailed in the section *Lag Control Master 0 on page 143*.

Tuning of resonance frequency

The procedure below details how to tune the resonance frequency.

⚠ **WARNING**

The system is unstable and therefore dangerous during the tuning process, since bad parameters or parameter combinations may be used! The safety procedures of the robot system must be carefully followed throughout the tuning process.

	Action
1	Measure the distance between the resonance peaks (in points) on the plot of the *Torque Ref.* signal. Divide the sampling frequency with this value: resonance frequency = sampling frequency / distance between resonance peaks This calculation gives a rough estimate (in Hz). **Note!** The sampling frequency of the logged data depends on the sampling interval that has been selected in the configuration of Test Signal Viewer.
2	The value of the resonance frequency should be in the range 3 to 25 (default 100). To tune the resonance frequency, program a short back-and-forth motion of the axis at maximum speed. The axis should not be allowed to reach full speed before deceleration. Use the `TUNE_DF` argument of the `TuneServo` command to adjust the resonance frequency and examine the *Torque Ref.* signal. Adjust the resonance frequency until the oscillations in the *Torque Ref.* signal are damped out.
3	Enter the tuned value of the resonance frequency to the parameter *Df*.

Result

The parameter *Df* is now tuned. Continue with the tuning procedure as described in the next section, and in section *Tuning of axes, complete procedure on page 110*.

6.2.8 Tuning of Nominal Acceleration and Nominal Deceleration

General

This section details how to tune the parameters *Nominal Acceleration* and *Nominal Deceleration*.

Parameter description

The parameters *Nominal Acceleration* and *Nominal Deceleration* belongs to the type *Acceleration Data* and are further described in the section *Acceleration Data on page 135*.

Preparations

The following preparations must be made before performing the tuning:

- If an axis has a varying moment of inertia, *Nominal Acceleration*and *Nominal Deceleration* should be tuned with the maximum moment of inertia. For a description of how to specify the inertia see *Specifying the inertia on page 119*.
- If gravity has an influence on the axis, then *Nominal Acceleration* should be tuned with a motion accelerating upwards (against gravity). *Nominal Deceleration* should be tuned with a stopping motion (deceleration) while moving downwards (in gravity direction).
- Program two test points for acceleration and two test points for deceleration with the following requirements:
  - Velocity: choose a velocity that is approximately 50% of the maximum speed of the additional axis.
  - Distance: the distance should be chosen to ensure that the axis stabilizes at the programmed velocity before deceleration starts.

Tuning of Nominal Acceleration and Nominal Deceleration

The procedure below details how to tune the parameters *Nominal Acceleration* and *Nominal Deceleration*.

Before beginning the tuning, observe the following warnings:

⚠ **WARNING**

The system is unstable and therefore dangerous during the tuning process, since bad parameters or parameter combinations may be used! The safety procedures of the robot system must be carefully followed throughout the tuning process.

⚠ **WARNING**

Kp can affect the torque level (torque_ref). Therefore, acceleration/deceleration tuning should be verified again if Kp is changed more than 10%.

Continues on next page

⚠ **WARNING**

For low values of Kp (e.g. <5) the axis might never reach the acceleration/deceleration value and these values can be tuned to unrealistically high values without reaching the torque limit. This can cause problems during an emergency stop if brake_control_on_delay is >0 (motors used at emergency stop). Do not continue to increase acceleration/deceleration if the effect on *torque_ref* is small. Always verify *torque_ref* level at emergency stop.

	Action	Note
1	Use Test Signal Viewer software to record the values of *Positive/Negative Torque Limit* and *Torque Ref.* for the axis.	How to define these signals is detailed in section *Defining signals in Test Signal Viewer on page 108*.
2	Use test positions for *Nominal Acceleration* and do the following: 1 Run the motion and check the value of *Torque Ref.* for the *Positive/Negative Torque Limit*. 2 Adjust the value of *Nominal Acceleration* upwards or downwards in increments of 0.5 until the *Torque Ref.* signal shows that the axis approaches, but does not reach, the torque limit. 3 Reduce the final value by 10% to allow for variations in the mechanical system over a period of time.	
3	Use test positions for *Nominal Deceleration* and do the following: 1 Run the motion and check the value of *Torque Ref.* for the *Positive/Negative Torque Limit*. 2 Adjust the value of *Nominal Deceleration* upwards or downwards in increments of 0.5 until the *Torque Ref.* signal shows that the axis approaches, but does not reach, the torque limit. 3 Reduce the final value by 10% to allow for variations in the mechanical system over a period of time.	

Result

The parameters *Nominal Acceleration* and *Nominal Deceleration* are now tuned. Continue with the tuning process as described in the next section. The complete tuning procedure is detailed in section *Tuning of axes, complete procedure on page 110*.

6.2.9 Final tuning of Kp, Kv and Ti

General

This section details how to tune the final value for the parameters *Kv, Gain Speed Loop*, *Kp, Gain Position Loop* and *Ti Integration Time Speed Loop*.

Parameter description

The parameters *Kv, Gain Speed Loop*, *Kp, Gain Position Loop* and *Ti Integration Time Speed Loop* belongs to the type *Lag Control Master 0* and are further described in the section *Lag Control Master 0 on page 143*.

Preparations

The following preparations must be made before performing the final tuning:

- The parameters must be initially tuned as detailed in the section *Initial tuning of Kv, Kp and Ti on page 113*.
- If the axis has a varying moment of inertia, *Kv*, *Kp* and *Ti* should be tuned with the maximum value for the moment of inertia.
- Program two test points with the following requirements:
  - Velocity: choose a velocity that is approximately 25% of the maximum speed of the additional axis. The speed must be low enough to guarantee that the axis does not encounter the torque limit but high enough to prevent friction from affecting the result.
  - Distance: choose a distance that ensures that the axis stabilizes at the programmed velocity before deceleration starts.

RAPID program

The following RAPID program can be used during the final tuning of the parameters *Kv, Gain Speed Loop*, *Kp, Gain Position Loop*, and *Ti Integration Time Speed Loop*:

```
PROC main()
  ActUnit M7DM1;
  TuneServo M7DM1,1,TuneValueKp\Type:=TUNE_KP;
  TuneServo M7DM1,1,TuneValueKv\Type:=TUNE_KV;
  TuneServo M7DM1,1,TuneValueTi\Type:=TUNE_TI;
  FOR i FROM 1 TO 10 DO
    MoveJ p1,v_tune,fine,tool0;
    MoveJ p2,v_tune,fine,tool0;
    MoveJ p1,v_tune,fine,tool0;
  ENDFOR
  DeactUnit M7DM1;
ENDPROC
```

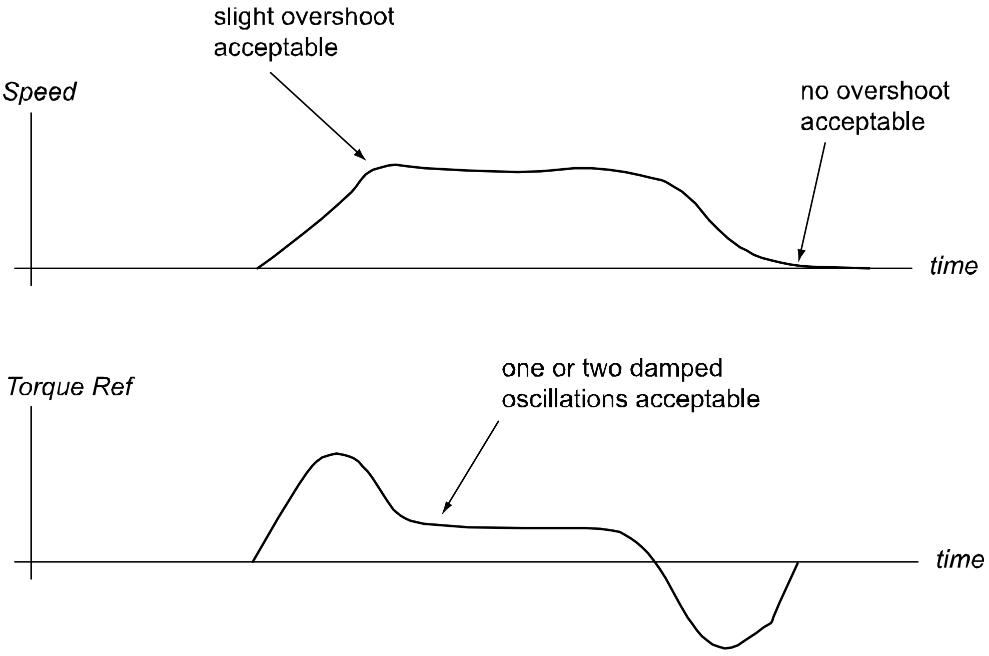
The arguments `TuneValueKp`, `TuneValueKv`, and `TuneValueTi` are percentage (1-500) of the set value in the loaded configuration file. 100% is the normal value.

Further descriptions of the RAPID instructions and arguments are found in *Technical reference manual - RAPID Instructions, Functions and Data types*.

Continues on next page

Illustration, final tuning of Kp

The figure below shows the desired profiles of *Speed* and *Torque Ref.* signals when making the final tuning of the parameter *Kp, Gain Position Loop*.



xx0400000649

Final tuning

The procedure below details how to make the final tuning of the parameters *Kv, Gain Speed Loop*, *Kp, Gain Position Loop*, and *Ti Integration Time Speed Loop*.

WARNING

The system is unstable and therefore dangerous during the tuning process, since bad parameters or parameter combinations may be used! The safety procedures of the robot system must be carefully followed throughout the tuning process.

	Action
1	Make the preparations detailed in the section *Preparations on page 125* before performing the final tuning.
2	Use the initially tuned values of the parameters *Kv, Gain Speed Loop*, *Kp, Gain Position Loop*, and *Ti Integration Time Speed Loop* as default values in the type *Lag Control Master 0*. Set the arguments `TuneValueKp`, `TuneValueKv`, and `TuneValueTi` in the RAPID program to 100.

Continues on next page

	Action
3	Make a final tuning of Kv by using the *RAPID program on page 125* with the following values: - Increase `TuneValueKv` in steps of 5% and observe the *Torque Ref.* in Test Signal Viewer. Stop when the axis starts to vibrate/oscillate. - Divide `TuneValueKv` by 2 and run the axis again, while observing the *Torque Ref.* There should be at most one or two damped oscillations after the acceleration stage. If *Torque Ref.* oscillates more than this, decrease its value somewhat. Kv is a critical parameter. A large value will result in a stiff axis and a fast response. If Kv is too small, Kp will also be limited, resulting in an under-utilized axis.
4	Make a final tuning of Kp by using the *RAPID program on page 125* and the following information: - Increase `TuneValueKp` slowly until the desired *Speed* and *Torque Ref.* profiles are achieved as shown in the figure *Illustration, final tuning of Kp on page 126*. The position error (lag) is inversely proportional to Kp thus a large value for Kp is desirable.
5	Make a final tuning of Ti by using the *RAPID program on page 125* with the following values: - Reduce `TuneValueTi` in steps of 5% until the effect can be seen on the plot of *Speed* as an increased overshoot. - Increase `TuneValueTi` by 5-10% until the effect is removed.
6	Calculate the final values of Kv, Kp and Ti by multiplying the initially tuned value in the configuration file (that was used as the default value) by the given `TuneValue` divided by 100. Default value * (TuneValue / 100)
7	Enter these new values to the parameters *Kv, Gain Speed Loop*, *Kp, Gain Position Loop* and *Ti Integration Time Speed Loop* in the type *Lag Control Master 0*.

Result

The parameters *Kv, Gain Speed Loop*, *Kp, Gain Position Loop*, and *Ti Integration Time Speed Loop* are now finally tuned.

6.3 Additional tuning

6.3.1 Tuning of the soft servo parameters

General

This section details how to tune the following parameters in the type *Lag Control Master 0*:

- *K Soft Min Factor*
- *K Soft Max Factor*
- *Kp/Kv Ratio Factor*
- *Ramp Time*

Tip

In most applications these parameters do not have to be trimmed and can be left at their default values.

Tuning of K Soft Min Factor

The procedure below details how to make the initial tuning of the parameter *K Soft Min Factor*.

Tip

The movements in this trim procedure should be done close to the point where the soft servo is activated, to minimize the risk of an axis collapsing.

	Action
1	Determine a maximum axis movement for which the axis should not move, when the softness is 100%. Such a movement can be 0.1 rad for a rotating axis.
2	Determine a minimum axis movement for which the axis should move, when the softness is 100%. Such a movement can be 0.2 rad for a rotating axis.
3	Activate the soft servo with softness 100% and perform the two movements.
4	If the axis moves for both movements, the axis is too stiff and *K Soft Min Factor* should be reduced. If the axis does not move for any movement, the axis is too soft and *K Soft Min Factor* should be increased.
5	Repeat step 3 and 4 until the axis does not move for the smaller movement but does move for the bigger movement.

Tuning of K Soft Max Factor

In most cases, *K Soft Max Factor* can be left at its default value (1.0).

If the axis is too stiff at 0% softness, reduce *K Soft Max Factor*. If the axis is too soft at 0% softness, increase *K Soft Max Factor*. The tuning can be made in a similar way as for *K Soft Min Factor*, but with smaller movements.

Continues on next page

Kp/Kv Ratio Factor

Kp/Kv Ratio Factor determines the stability margin for the axis. A value less than 1.0 increases the stability. It is not possible to set this parameter to a value larger than 1.0 since the stability of the axis would be jeopardized.

Ramp Time

If *Ramp Time* is changed, the duration of the activation and deactivation phase will change. A short ramp time can result in a twitch of the axis at activation.

6.3.2 Additional tuning for servo guns

Description in separate manual

The specifics for tuning a servo gun are described in *Application manual - Servo gun tuning*.

7 Error handling

7.1 Error management

General

This section details how to handle fault localization after having performed system configuration.

Fault localization is done in the following steps:

1. start the Hyper Terminal to get access to the log file containing information about the communication between the controller and the PC
2. check the log file and search for faults
3. edit the *.cfg file
4. reload the configuration file

For a description of how to use a Hyper Terminal see the section *How to use the Microsoft HyperTerminal on page 133*.

Handling errors - an iterative process

Fault localization is an iterative process that must go on until all errors are eliminated. The following steps describe the order of the process:

	Description
1	Search for error messages in the log file. Identify the first error in the file. Status 0 implies that the file is found and the operation has passed. A negative status value implies that an error has occurred. For an example of the status information see illustration in the section below.
2	Open the configuration file (internal or external configuration file) and correct the first error that was found in the log file. The illustration below shows a part of an internal configuration file. **Note!** To avoid new error messages only correct one error at a time.
3	Save the edited configuration file and load the file to the controller.
4	If files are loaded by install.cmd, restart the controller using the restart mode **Reset system**. If files are loaded manually, perform a restart.
5	Go back to step 1 and proceed with the four steps above until all errors are eliminated.

Illustration of error messages in log file

The following shows an example of the contents of the log file.

```
Status 0 from echo Installing XYZ on drivemodule 1
Status 0 from config
     /hd0a/SAC_5.XX.XXXX/GantryArea/XYZ_rob11/irbcfg/INT_XYZ_11.cfg
05-10-21 15:41:34 MCO :type = ERROR id = CFG code = 5
arg 0 : 177
arg 1 : INT_XYZ_11.cfg
arg 2 : pole_pairs
arg 3 : 0
arg 4 : 20
```

Continues on next page

Examples of error messages

The table below gives some examples of common error messages that can show up in the log file after system configuration.

Error message in log file	Description - cause of failure	How to correct the error
04-01-13 16:33:46 MC0: param.c 1286 cfg_get_named_instance for 'DRIVE_UNIT' ’M2DM1’ failed.	With regard to hardware drive unit position and node position; do not exist for axis rob11_2.	Check the hardware and set the correct node values for the parameter *Use Drive Unit*, in the INT_XYZ_11.cfg file. The parameter is described in *Technical reference manual - System parameters*.
04-01-13 17:43:59 MC0: cfg_file.c 2142 Mandatory attribute 'name' missing in line 180 Status -13 from config compr: */hd0a/SAC_5.XX.XXXX/GantryArea /XYZ_rob11/irb-cfg/INT_XYZ_11.cfg*	In this case, the sign for the line breaking, "\", is missing on line 179 in the INT_XYZ_11.cfg file. The error message indicates that the error exists in some of the lines before line 180.	Check the lines for missing or incorrect signs.
04-01-13 17:50:57 MC0: type = ERROR id = SYSTEM code = 136 arg 0: pole_pairs arg 1: 180 Status -12 from config compr: */hd0a/SAC_5.XX.XXXX/GantryArea /XYZ_rob11/irb-cfg/INT_XYZ_11.cfg*	The value of pole_pairs (in this case, on line 179) in the INT_XYZ_11.cfg file is not within the permitted limits. The error message indicates that the error exists in the parameter `pole_pairs`, in some of the lines before line 180.	Change the value of the parameter *Pole Pairs* in the INT_XYZ_11.cfg file. The parameter is described in *Technical reference manual - System parameters*.

Summary

Error management is necessary to secure that the right configuration file for a certain kinematic model is used. It is also important to check that parameter settings have been done with correct/allowed values.

Every time a new system configuration is done or axes are tuned, fault localization and error correction must also be done to ensure that any remaining errors will be eliminated.

7.2 How to use the Microsoft HyperTerminal

General

This section details how to connect the robot system to the Microsoft HyperTerminal.

The Microsoft HyperTerminal is a program that can be used to connect the controller to other computers, for example a PC.

How to use a Hyper Terminal

	Action
1	Connect the PC to the controller. Use the CONSOLE port on the computer module in the controller and the COM1 port in the PC.
2	Open the Microsoft HyperTerminal application. A Microsoft Windows application may be used: Start / Programs / Accessories / Communications / HyperTerminal.
3	Use UNIX commands to navigate in the controller software.

This page is intentionally left blank

8 System parameters

8.1 Acceleration Data

General

These parameters are applicable to each arm of the external robot in question.

Parameter description

The parameters belong to the configuration type *Acceleration Data* in the *Motion* topic.

Cfg name	Parameter name	Description
`name`	Name	Name of the *Acceleration Data* group. Max 32 characters.
`wc_acc`	Nominal Acceleration	Axis acceleration in rad/s^2. If the value is too high, the motor will reach the torque limit and result in poor path performance.
`wc_dec`	Nominal Deceleration	Axis deceleration in rad/s^2. If the value is too high, the motor will reach the torque limit and the axis will overshoot in fine points.

8.2 Arm

General

These parameters are applicable to each arm of the robot in question.

Parameter description

The parameters belong to the configuration type *Arm* in the *Motion* topic.

Common parameters

Cfg name	Parameter name	Description
`upper_joint_bound`	Upper Joint Bound	Upper bound for the axis work area (in radians or meters). The axis cannot be moved beyond this limit during jogging or program execution.
`lower_joint_bound`	Lower Joint Bound	Lower bound for the axis work area (in radians or meters). The axis cannot be moved beyond this limit during jogging or program execution.

Parameters for additional axes

Cfg name	Parameter name	Description
`independent_joint_on`	Independent Joint	Set parameter to value On in order to activate the possibility to use independent joint instructions. Default value is Off.
`ind_upper_joint_bound`	Independent Upper Joint Bound	Upper bound for the axis work area when operating in independent mode (in radians or meters.
`ind_lower_joint_bound`	Independent Lower Joint Bound	Lower bound for the axis work area when operating in independent mode (in radians or meters).

Parameters for non ABB robots

Cfg name	Parameter name	Description
`name`	Name	Name of the ARM data group, e.g. x.
`use_arm_type`	Use arm type	ID name for ARM_TYPE data group.
`use_acc_data`	Use acc data	ID name for ACC_DATA data group.
`use_arm_calib`	Use arm calib	ID name for ARM_CALIB data group.
`lower_joint_bound_min`	-	Minimum value for lower_joint_bound. The unit is radian or meters.
`upper_joint_bound_max`	-	Maximum value for upper_joint_bound. The unit is radian or meters.
`cal_position`	Calibration position	Calibration position. The unit is in radians or meters.

8.3 Arm Calib

General

These parameters are applicable to each arm of the external robot in question.

Parameter description

The following parameters belong to the topic *Motion* and the type *Arm Calib*.

Parameters for non ABB robots

Cfg name	Parameter name	Description
`name`	Name	Name of the ARM_CALIB data group.

8.4 Arm Type

General

These parameters are applicable to each arm of the robot in question.

Parameter description

The following parameters belong to the topic *Motion* and the type *Arm Type*.

Parameters for non ABB robots

Cfg name	Parameter name	Description
`name`	Name	Name of the ARM_TYPE data group.
`length`	-	Arm length (according to Craig's definition, see reference below). (Meter)
`offset_x`	-	Offset between 1st and 2nd axes of the arm, x direction. (Meter)
`Offset_y`	-	Offset between 1st and 2nd axes of the arm, y direction. (Meter)
`offset_z`	-	Offset between 1st and 2nd axes of the arm, z direction. (Meter)
`theta_home_position`	-	Axis angle of arm in home position (theta according to Craig's definition). (Radians)
`attitude`	-	Angle between the two axes of the arm (alpha according to Craig's definition). (Radians)

The Denavit-Hartenberg notation according to John J. Craig in *Introduction to Robotics*, *Mechanics and Control*

8.5 Brake

General

These parameters control the emergency brake behavior. They are applicable to each additional axis with a brake strong enough to hold against gravitation.

Parameter description

The parameters belong to the configuration type *Brake* in the *Motion* topic.

Cfg name	Parameter name	Description
`name`	Name	ID name of the brake.
`control_off_delay_time`	Control Off Delay	The motor torque is turned off after this delay time has expired. It must be long enough to ensure that the mechanical brake has started working, or the axis risk dropping toward the ground.
`brake_control_on_delay_time`	Brake Control On Delay	Period of time during which retardation by motor is used. It should be set close or equal to the mechanical brake activation time and must be long enough to damp mechanical oscillation. If the axis is still moving after this time has expired, the electrical torque brake is activated.
`brake_control_on_min_delay_time`	Brake Control Min Delay	Used by the brake algorithm much the same as *Control Off Delay*. Should be set to the same value as that parameter.
`absolute_brake_torque`	Absolute Brake Torque	Specifies max brake torque generated by the motor in the electrical torque brake phase. *Absolute Brake Torque* together with torque generated by the mechanical brake must not exceed max allowed torque for the arm, in order not to damage arm and gearbox.
`brake_ramp_speed_limit`	Brake Ramp Speed Limit	Specifies the speed limit for torque reduction in the electrical torque brake phase and is typically set to zero.

8.6 Force Master

General

Force Master is used to define how a servo gun behaves during force control. The parameters only affect the servo gun when it is in force control mode.

Parameter description

The following table contains the parameters that belongs to the topic *Motion* and the type *Force master*.

Parameters for additional axes

Cfg	Parameter name	Description
bandwidth_ramping	References Bandwidth	The frequency limit for the low pass filter for reference values.
ramp_time_switch	Use Ramp Time	Determines if the ramping of the tip force should use a constant time or a constant gradient.
ramp_torque_ref_closing	Ramp when Increase Force	Determines how fast force is built up while closing the tool when *Use ramp time* is set to No.
ramp_time	Ramp Time	Determines how fast force is built up while closing the tool when *Use Ramp Time* is set to Yes.
bandwidth_lp	Collision LP Bandwidth	Frequency limit for the low pass filter used for tip wear calibration.
alarm_torque	Collision Alarm Torque	Determines how hard the tool tips will be pressed together during the first gun closing of new tips calibrations and tool change calibrations.
col_speed	Collision Speed (%)	Determines the servo gun speed during the first gun closing of new tips calibrations and tool change calibrations.
distance_to_contact_position	Collision Delta Position (m)	Defines the distance the servo tool has gone beyond the contact position when the motor torque has reached the value specified in *Collision Alarm Torque*.
force_ready_detection_bandwidth	Force Detection Bandwidth	The feedback motor speed is filtered through a LP filter with this bandwidth. This is to avoid that variations in the speed will trigger the force detection too early.
force_ready_detection_speed	Force Detection Speed	When the feedback motor speed is below this value, it is considered that the ordered force is reached.
delay_ramp	Delay Ramp	Delays the starting of torque ramp when force control is started.

8.7 Force Master Control

General

These parameters are used to set the speed limit and speed loop gain as functions of the torque.

Parameter description

The following parameters belong to the topic *Motion* and the type *Force Master Control*.

Cfg name	Parameter name	Description
no_of_posts	No. of Speed Limits	The number of points used to define speed limit and speed loop gain as functions of the torque. Up to 6 points can be defined.
torque_1 - torque_6	Torque 1- Torque 6	The torque levels, corresponding to the ordered tip force, for which the speed limit and speed loop gain values are defined.
speed_lim_1 - speed_lim_6	Speed Limit 1-6	*Speed Limit 1* to *Speed Limit 6* are used to define the maximum speed depending on the ordered tip force.
Kv_1 - Kv_6	Kv 1-6	*Kv 1* to *Kv 6* are used to define the speed loop gain for reducing the speed when the speed limit is exceeded.

8.8 Joint

General

These parameters are used to identify individual axes.

Parameter description

The following parameters belong to the topic *Motion* and the type *Joint.*

Parameters for additional axes

Cfg name	Parameter name	Description
`logical_axis`	Logical Axis	Used by RAPID programs to identify individual axes. Robots from ABB normally use the values 1-6, while additional axes use 7-12. E.g. the value 7 of *Logical Axis* corresponds to eax_a in the data type `robtarget`, 8 corresponds to eax_b, etc.

8.9 Lag Control Master 0

General

The type *Lag Control Master 0* is normally used for regulation of axes without any dynamic model.

Parameter description

The following table contains the parameters that belong to the type *Lag Control Master 0* in the topic *Motion*.

Common parameters

Cfg name	Parameter name	Description
Kp	Kp, Gain Position Loop	The amplification of the position control, e.g. 15. A high value will give a stiff axis that quickly assumes its new position. The value should be large without inducing overshoot in the position or oscillations of the axis.
Kv	Kv, Gain speed loop	The amplification of the velocity control, e.g. 2. A high value gives better high frequency stiffness, better response speed and low overshoot. If the value is too high the axis will vibrate.
Ti	Ti, Integration Time Speed Loop	Integration time in the speed regulation loop. The lower the value of *Ti Integration Time Speed Loop*, the better tracking and disturbance rejection. Too low value may case oscillation or noise.
ffw_mode	FFW Mode	Feed Forward mode. Possible options are no ffw (0), speed ffw (1) or torque ffw (2). • No: The controller is driven by the position error (lag). • Spd: The controller receives information about the desired speed of the axis. Speed is the recommended configuration. • Trq: The controller uses the desired speed and acceleration of the axis to calculate the desired motor torque. This requires knowledge of the mass inertia of the axis, which must be supplied by the user. For this reason its more difficult to tune and is only recommended for experienced users.

Parameters for additional axes

Cfg name	Parameter name	Description
use_inpos_forced_control	Forced Control Active	Determines whether forced gain control is active for this joint. If set to Yes, *Affects Forced Control* in *Supervision* should normally also be set to Yes for this joint (see below).
Kp_forced_factor	Forced Factor for Kp	The forced factor for Kp, if forced gain control is active.
Ki_forced_factor	Forced Factor for Ki	The forced factor for Ki, if forced gain control is active.
Kp_raise_time	Rise time for Kp	The rise time for forced Kp.
bandwidth	Bandwidth	This parameter should be left at its default value.

Continues on next page

Cfg name	Parameter name	Description
`delay_time`	Delay	This parameter should be left at its default value.
`resonance_frequency`	Df	Dynamic factor. This parameter is only available in the Trq configuration. It can be used to damp oscillations of the axis due to mechanical resonance. Initially *Df* should be left at its default value. It can be adjusted once the other controller parameters have been fixed.
`inertia`	Inertia	Total mass moment of inertia at motor side.
`soft_servo_K_max_factor`	K Soft Max Factor	Determines the value of the product Kp*Kv when the soft servo is used with softness 0%. *K Soft Max Factor* should be in the range 0.001-1000 (default 1.0). When the soft servo is activated with 0% softness, the control parameters Kp and Kv are be tuned such that Kp*Kv = (Kp*Kv)normal**K Soft Max Factor*, where (Kp*Kv)normal is the product of Kp and Kv during normal operation.
`soft_servo_K_min_factor`	K Soft Min Factor	Determines the value of the product Kp*Kv when the soft servo is used with softness 100%. *K Soft Min Factor* should be in the range 0.001-1000 (default 0.01). When the soft servo is activated with 100% softness, the control parameters Kp and Kv are tuned such that Kp*Kv = (Kp*Kv)normal**K Soft Min Factor*.
`soft_servo_Kp_Kv_ratio_factor`	Kp/Kv Ratio Factor	Factor used to alter the Kp/Kv ratio during soft servo. *Kp/Kv Ratio Factor* should be in the range 0.001-1000 (default 1.0). In soft servo mode, Kp and Kv are tuned such that Kp/Kv = (Kp/Kv)normal**Kp/Kv Ratio Factor*.
`soft_servo_t_ramp`	Ramp time	Default time for activation of the soft servo. The default value is 0.5 s.

8.10 Measurement Channel

General

The type *Measurement Channel* provides parameters that are applicable to each axis of the robot in question.

Parameter description

The following table contains the parameters that belong to the type *Measurement Channel* in the topic *Motion*.

Parameters for additional axes

Cfg name	Parameter name	Description
disconnect_at_deactivate	Disconnect at Deactivate	The measurement channel for a deactivated motor can be disconnected (Yes/No).WARNING! If the axis is moved when disconnected, the position of the axis might be wrong after reconnecting, and this will not be detected by the controller. The position after reconnection will be correct if the axis is not moved, or if the movement is less than 0.5 motor revolutions. For servo guns, there is a RAPID calibration method available (the ToolChange calibration) that will adjust any positional error caused by gun movement during disconnection.

Parameters for non ABB robots

Cfg name	Parameter name	Description
name	Name	Name of the MEASUREMENT_CHANNEL data group, e.g. x.
use_measurement_board_type	Use Measurement Board Type	Measurement board type.
measurement_link	Measurement Link	The number of the measurement system. The number is 1 or 2. Default=1.
board_position	Board Position	The number of the board. The number is 1 or 2. Default=1.
measurement_node	Measurement Node	Measurement node. Node number: 1 to 7. Default=1.
memory_index	Memory Index	The index number on the measurement board where the data is saved.

8.11 Mechanical Unit

General

The type *Mechanical Unit* provides parameters that are used to define the Mechanical Unit.

Parameter description

The following table contains the parameters that belong to the type *Mechanical Unit* in the topic *Motion*.

Common parameters

Cfg name	Parameter name	Description
`name`	Name	Mechanical unit name.
`use_activation relay`	Use Activation Relay	ID name for activation relay.
`use_brake_relay`	Use Brake Relay	ID name for the brake relay.
`activate_at_start_up`	Activate at Start Up	Activate the mechanical unit at start-up.
`dactivation_forbidden`	Deactivation Forbidden	The unit is not allowed to be activated.

Parameters for additional axes

Cfg name	Parameter name	Description
`use_connection_relay`	Use Connection Relay	ID name of the relay that must be activated when the mechanical unit is activated.
`allow_move_of_user_frame`	Allow Move of User Frame	The unit can move a user frame, e.g. a work object.
`use_single_0` - `use_single_5`	Use Single 1 - Use Single 6	Defines which singles are part of the mechanical unit. Corresponds to the parameter *Name* in the type *Single*.

Parameters for non ABB robots

Cfg name	Parameter name	Description
`use_run_enable`	Use Run Enable	ID name for run enable input signal.

8.12 Motion Planner

Parameter description

The following table contains the parameters that belong to the type *Motion Planner* in the topic *Motion*.

Parameters for non ABB robots

Cfg name	Parameter name	Description
`name`	Name	Motion Planner name.
`brake_on_timeout`	Brake on Time	Brake activation time in motor on state (in seconds). Min.=0.3, Max.=3600000, Default=3600000.
`dynamic_resolution`	Dynamic Resolution	(Dynamic sample time)/0.024192. Min.=0.1667, Max.=1.0, Default=1.0.
`path_resolution`	Path Resolution	(Geometric sample time)/0.024192. If a very low programmed speed (less than 1 mm/s) is used, a small variation of the speed can be observed. This oscillation of the speed can be reduced by increasing `path_resolution`.
`std_servo_queue_time`	Queue Time	Standard servo queue time. Min.=0.004032, Max.=0.290304, Default=0.096768.
`perpendicular_acc_ratio`	-	Perpendicular acceleration ratio. Min.=0.1, Max.=1.5.
`dyn_ipol_decbuf_type`	-	OPTIMAL_TIME - original, OPTIMAL_PATH - less high torques.
`micro_ipol_type`	-	micro ipol type 0,.,n. Min.=0.
`cpu_load_added_to_dsp`	-	Used to verify CPU load margin in DSP.Min=0, Max=25, Default=0.
`motion_sup_max_level`	Motion Supervision Max Level	Maximum motion sup level and tune value. Min.=10, Max.=500, Default=300.

8.13 Motion System

Parameter description

The following table contains the parameters that belong to the type *Motion System* in the topic *Motion*.

Parameters for non ABB robots

Cfg name	Parameter name	Description
`name`	Name	Motion system name. Min.=-100, Max.=100.
`min_temp_ambient_cabinet`	Min Temperature Cabinet	Minimum ambient temperature for the cabinet. Min.=-100, Max.=100.
`max_temp_ambient_cabinet`	Max Temperature Cabinet	Maximum ambient temperature for the cabinet. Min.=-100, Max.=100.
`min_temp_ambient_robot`	Min Temperature Robot	Minimum ambient temperature for the robot. Min.=-100, Max.=100.
`max_temp_ambient_robot`	Max Temperature Robot	Maximum ambient temperature for the robot. Min.=-100, Max.=100.

8.14 Motor

General

The type *Motor* provides parameters that are applicable to each axis of the robot in question.

Parameter description

The following table contains the parameters that belong to the type *Motor* in the topic *Motion*.

Parameters for non ABB robots

Cfg name	Parameter name	Description
`name`	Name	Name of the MOTOR data group.
`use_motor_type`	Use Motor Type	ID name of the MOTOR_TYPE data group.
`use_motor_calib`	Use Motor Calibration	ID name of the MOTOR_CALIB data group.
`stator_cooling_factor`	-	Cooling factor for the stator, multiplied with attribute torque_0. Min.=0, Max.=10, Default=1.

8.15 Motor Calibration

General

The type *Motor Calibration* provides parameters that are applicable to each axis of the robot in question

Parameter description

The following table contains the parameters that belong to the type *Motor Calibration* in the topic *Motion*.

Common parameters

Cfg name	Parameter name	Description
`name`	Name	Name of the MOTOR_CALIB data group.
`com_offset`	Commutator Offset	The motor angle when voltage is applied between the phases S and T. For ABB motors *Commutator offset* should always be 1.5708.
`cal_offset`	Calibration Offset	Can be updated by moving the axes to their calibration positions and then fine calibrating.

Parameters for non ABB robots

Cfg name	Parameter name	Description
`valid_com_offset`	Commutator Offset Valid	Yes If `com_offset` is valid.
`valid_cal_offset`	Calibration Offset Valid	Yes if `cal_offset` is valid.

8.16 Motor Type

General

The type *Motor Type* provides parameters that are applicable to each axis of the robot in question.

Parameter description

The following table contains the parameters that belong to the type *Motor Type* in the topic *Motion*.

Common parameters

Cfg name	Parameter name	Description
`pole_pairs`	Pole Pairs	Number of pole pairs.
`ke`	ke Phase to Phase	Nominal voltage constant, induced voltage phase to phase (1V/1000rpm <=> 0.00955 Vs/rad) The unit is Vs/rad.
`i_max`	Max Current	Max. current without irreversible demagnetization. The unit is A rms.
`r_stator_20`	Phase Resistance	Stator phase resistance at 20 degrees Celsius. The unit is ohm.
`l_stator`	Phase inductance	Stator phase inductance at zero current. The unit is Henry.

Parameters for non ABB robots

Cfg name	Parameter name	Description
`inertia`	-	Motor and resolver inertia on motor side. The unit is kgm^2.
`torque_0`	Stall Torque	Stall torque, infinite time, `temp_stator_rise` to `temp_stator_max`. The unit is Nm.
`ke_temp_coef_20`	-	Temperature reduction coefficient for ke, a t 20 degrees. The unit is 1/K.
`ke_stability_coef_20`	-	Long-term stability reduction constant for ke after 4000 hours.
`ke_tolerance_min`	-	Minimum tolerance for ke (%/100) Min. ke= ke*(1+ke_tolerance_min).
`ke_tolerance_max`	-	Maximum tolerance for ke (%/100). Max. ke= ke*(1+ke_tolerance_max).
`ke_red_2i0`	-	Current dependant reduction of ke at two times rated current (%/100).
`torque_losses_at_speed1`	-	Total torque losses due to friction and iron losses at speed1 (cf.below). The unit is Nm.
`torque_losses_at_speed2`	-	Total torque losses due to friction and iron losses at speed2 (cf.below). The unit is Nm.

Continues on next page

Cfg name	Parameter name	Description
`torque_losses_at_speed3`	-	Total torque losses due to friction and iron losses at speed3 (cf.below). The unit is Nm.
`speed1`	-	The speed at which torque_losses_at_speed1 is defined in rad/s.
`speed2`	-	The speed at which torque_losses_at_speed2 is defined in rad/s.
`speed3`	-	The speed at which torque_losses_at_speed3 is defined in rad/s.
`temp_stator_max`	-	Maximum temperature for the stator winding. The unit is degrees Celsius.
`temp_stator_rise`	-	Maximum temperature rise for the stator winding. The unit is degrees Celsius.
`temp_rotor_max`	-	Maximum temperature for the rotor. The unit is degrees Celsius.
`temp_rotor_rise`	-	Maximum temperature rise for the rotor. The unit is degrees Celsius.
`r_stator_temp_coef_20`	-	Temperature coefficient for the stator resistance at 20 degrees Celsius.

8.17 Relay

General

The type *Relay* provides parameters that are used to define relay.

Parameter description

The following table contains the parameters that belong to the type *Relay* in the topic *Motion*.

Common parameters

Cfg name	Parameter name	Description
`name`	Name	Name of the relay: • For ABB axes: Must be the same as the *Use connection relay* parameter defined in the type *Mechanical Unit*. • For ABB axes: The name must be changed when more relays are added.
`out_signal`	Output Signal	Denotes the logical name of the output signal to the relay. The name must be identical (including upper and lower case letters) to the name used for the signal definition.
`in_signal`	Input Signal	Denotes the logical name of the input signal to the relay. The name must be identical (including upper and lower case letters) to the name used for the signal definition. The signal must be defined as "safety" and "INTERNAL".

8.18 Robot

Parameter description

The following table contains the parameters that belong to the type *Robot* in the topic *Motion*.

Parameters for non ABB robots

Cfg name	Parameter name	Description
`name`	Name	Name of the robot, for example master.
`use_robot_type`	Use Robot Type	Name off the Kinematic model according to gantry-type kinematic models, see *Kinematic models on page 28*.
`use_joint_0`	Use Joint 1	ID name of 1st axis, for example robx_1.
`use_joint_1`	Use Joint 2	ID name of 2nd axis, for example robx_2.
`use_joint_2`	Use Joint 3	ID name of 3rd axis, for example robx_3.
`use_joint_3`	Use Joint 4	ID name of 4th axis, for example robx_4.
`use_joint_4`	Use Joint 5	ID name of 5th axis, for example robx_5.
`use_joint_5`	Use Joint 6	ID name of 6th axis, for example robx_6.
`base_frame_pos_x`	Base Frame x	Base frame position in respect of world frame coordinate system, x - direction Min.=1000, Max.=1000, Default=0 (meters).
`base_frame_pos_y`	Base Frame y	Base frame position in respect of world frame coordinate system, y - direction (meters).
`base_frame_pos_z`	Base Frame z	Base frame position in respect of world frame coordinate system, z - direction (meters).
`base_frame_orient_u0`	Base Frame q1	Base frame orientation in respect of world frame coordinate system, first quaternion (q1). Min.=-1, Max.=1, Default=0.
`base_frame_orient_u1`	Base Frame q2	Base frame orientation in respect of world frame coordinate system, second quaternion (q2).
`base_frame_orient_u2`	Base Frame q3	Base frame orientation in respect of world frame coordinate system, third quaternion (q3).
`base_frame_orient_u3`	Base Frame q4	Base frame orientation in respect of world frame coordinate system, fourth quaternion (q4).
`rot_x_tol`	Orientation Tolerance about x	Orientation tolerance (in radians) Min.=0, Max.=4, Default=0.001.
`rot_y_tol`	Orientation Tolerance about y	Orientation tolerance (in radians). Min.=0, Max.=4, Default=0.001.
`rot_z_tol`	Orientation Tolerance about z	Orientation tolerance (in radians). Min.=0, Max.=4, Default=0.001.

8.19 SG Process

General

The type *SG Process* provides parameters that are applicable for servo guns.

Parameter description

The following table contains the parameters that belong to the type *SG Process* in the topic *Motion*.

Parameters for additional axes

Cfg name	Parameter name	Description
`sync_check_off`	Sync Check Off	By setting this parameter to 'Yes', it will be possible to close the gun without having done a tip calibration. This is useful during the tuning procedure of a servo gun or if running an application where tip calibrations are not used. When running the gun in production, it is recommended to always have the sync check active in order to prevent possible damage caused by closing an unsynchronized pair of gun tips.
`min_close_time_adjust`	Close Time Adjust	Constant time adjustment (s), positive or negative, of the moment when the gun tips reach contact during a gun closure. This value is normally zero. May be used to delay the closing slightly when the synchronized pre closing is used for welding.
`close_position_adjust`	Close Position Adjust	When the tool tips reach the position (plate thickness) ordered by the close instruction, the force control starts. This tool tip position can be adjusted with *Close Position Adjust* to make the force control start earlier.
`pre_sync_delay_time`	Force Ready Delay	Constant time delay (s) before sending the weld ready signal after reaching the programmed force.
`max_motor_torque`	Max Force Control Motor Torque	Maximum allowed motor torque (Nm) during force control. The parameter will protect the gun from too high programmed force, by reducing the resulting motor torque to this upper level. A warning will be logged whenever this happens. The value must not be set higher than the *Torque abs. max* (type *Stress duty cycle*) which defines the maximum output of motor torque during both force and position control.
`post_sync_time`	Post-synchronization Time	Release time anticipation (s) of the next robot movement after a weld. This parameter can be tuned to synchronize the gun opening with the next robot movement. The synchronization may fail if the parameters is set too high.
`calib_mode`	Calibration Mode	The number of closings performed during a Tipwear calibration. Normally 2 closings will be ok. An increase may improve the accuracy of thickness detection for some servo guns.

Continues on next page

Cfg name	Parameter name	Description
`calib_force_high`	Calibration Force High	The maximum tip force (N) used during a TipWear calibration. For best result of the thickness detection, it is recommended to use the max programmed weld force.
`calib_force_low`	Calibration Force Low	The minimum tip force (N) used during a TipWear calibration. For best result of the thickness detection, it is recommended to use the minimum programmed weld force.
`calib_time`	Calibration Time	The wait time (s) during a calibration before the positional gun tip correction is done. Recommended value ca: 0.5 s.
`no_of_active_db_posts`	Number of Stored Forces	Number of stored forces in the force VS motor torque table. The minimum value allowed is 2.

`squeeze_force_1`
-
`squeeze_force_10`
	Tip Forces 1 - 10	Gun tip force 1 (N) - Gun tip force 10 (N).

`squeeze_torque_1`
-
`squeeze_torque_10`
	Motor Torque 1 - 10	Motor torque 1 (Nm) - Motor torque 10 (Nm).

8.20 Single

Parameter description

The following table contains parameters that belong to the topic *Motion* and the type *Single*.

Parameters for additional axes

Cfg name	Parameter name	Description
`name`	Name	The name of the single. A single axis mechanical unit without kinematic model must have the name of single 1 set to the same name as the mechanical unit.
`use_single_type`	Use Single Type	Defines which single type to use.

8.21 Single Type

Parameter description

The following table contains the parameters that belong to the topic *Motion* and the type *Single Type*.

Parameters for additional axes

Cfg name	Parameter name	Description
`mechanics`	Mechanics	• TRACK - Linear motion. • FREE_ROT - Rotating additional axis. • SG_LIN - Linear servo gun motion.

8.22 Stress Duty Cycle

General

The type *Stress Duty Cycle* provides parameters that are applicable to each axis of the robot in question.

Parameter description

The following table contains the parameters that belong to the type *Stress Duty Cycle* in the topic *Motion*.

Common parameters

Cfg name	**Parameter name**	**Description**
`name`	**Name**	**Name of the STRESS_DUTY_CYCLE.**
`speed_absolute_max`	**Speed Absolute Max**	**The absolute highest motor speed to be used. (rad/s)**
`torque_absolute_max`	**Torque Absolute Max**	**The absolute highest motor torque to be used. (rad/sec)** • **For non ABB robots: If** `torque_absolute_max` **is to high it may result in a configuration error at restart. To avoid this make sure that:** `torque_absolute_max` **< sqrt(3)***`ke*i_max`**.**

8.23 Supervision

General

The type *Supervision* provides parameters that are applicable to each axis of the robot in question.

Parameter description

The following table contains the parameters that belong to the type *Supervision* in the topic *Motion*.

Parameters for additional axes

Cfg name	Parameter name	Description
`joint_affect_forced_Kp`	Affects Forced Control	Determines whether this joint effects forced gain control.
`Kp_forced_on_limit`	Forced on Position Limit	The upper position limit for forced gain control.
`Kp_forced_off_limit`	Forced off Position Limit	The lower limit for forced gain control.

Parameters for non ABB robots

Cfg name	Parameter name	Description
`name`	Name	Name of the SUPERVISION data group.
`use_supervision_type`	Use Supervision type	ID name of SUPERVISION_TYPE.
`power_up_position_on`	Power Up Position Supervision	Power up position supervision On, default is Off.
`counter_supervision_on`	Counter Supervision	Counter supervision On, default is Off.
`position_supervision_on`	Position Supervision	Position supervision On, default is Off.
`speed_supervision_on`	Speed Supervision	Speed supervision On, default is Off.
`load_supervision_on`	Load Supervision	Load supervision On, default is Off.
`jam_supervision_on`	Jam Supervision	Jam supervision On, default is Off.
`in_position_range`	In Position Range	-
`normalized_zero_speed`	Zero Speed (%)	-
`dsp_torque_limitation_zero_speed_width`	-	Deadband speed width (in rad/s on motor side).

8.24 Supervision Type

General

The type *Supervision Type* provides parameters that are applicable to each axis of the robot in question.

Parameter description

The following table contains the parameters that belong to the data group *Supervision Type* in the topic *Motion*.

Parameters for non ABB robots

Cfg name	Parameter name	Description
`name`	Name	Name of the SUPERVISION_TYPE data group.
`static_power_up_position_limit`	-	Static power up position error limit at zero speed.The unit is radians, Min.=0 and Max.=30.
`dynamic_power_up_position_limit`	Dynamic Power Up Position Limit	Dynamic power up position error limit at zero speed, the unit is radians.
`static_position_limit`	-	Position error limit at zero speed, the unit is radians on motor side.
`dynamic_postion_limit`	-	Position error limit at max speed, the unit is radians on motor side.
`static_normalized_speed_limit`	-	Speed error limit at zero speed. (% max. speed).
`dynamic_normalized_speed_limit`	-	Speed error limit at max speed (% max speed).
`normalized_influence_sensitivity`	-	Speed error influence sensitivity reduction. (% max. speed).
`speed_half_time`	-	Declination factor half time for supervision limits. The unit is seconds, Min=0 and Max.=5.
`max_jam_normalized_speed`	-	Speed limit for jam versus overload supervision. (% max. speed).
`max_overload_time`	-	Maximum overload time. The unit is seconds, Min.=0 and Max.=20.
`max_jam_time`	Max Jam Time	Max jam time. The unit is seconds, Min.=0 and Max.=20.
`teach_mode_speed_max_main`	Teach Max Speed Main	Maximum ordered speed ratio in teach mode (% max speed). Min.=0, Max.=1, Deafult=0.15.
`teach_mode_speed_max_dsp`	Teach Max Speed DSP	Maximum supervision speed ratio in teach mode for axis computer (% max speed). Min.=0, Max.= 1, Default=0.28.

8.25 Transmission

General

The type *Transmission* provides parameters that are applicable to each arm of the robot in question.

Parameter description

The following table contains the parameters that belong to the type *Transmission* in the topic *Motion*.

Common parameters

Cfg name	Parameter name	Description
`name`	Name	Name off the TRANSMISSION data group.
`transm_joint`	Transmission Gear Ratio	Gear ratio between motor and axis. For linear axis gear ratio is specified as motor rotation in radians per meter linear move (21.43 denotes that when the motor rotates 21.43 radians - the axis moves 1 m).
`rotating_move`	Rotating Move	Denotes whether the axis is of the rotating type (Yes) or linear type (No).
`high_gear`	Transmission Gear High	The integer value of the numerator of the transmission gear ratio. Only used for independent joints.
`low_gear`	Transmission Gear Low	The integer value of the denominators of the transmission gear ratio. Only used for independent joints.Example: For a rotating axis with high gear 100 and low gear 30, has a transmission gear ratio of 100/30=3.333333.

8.26 Uncalibrated Control Master 0

General

The type *Uncalibrated Control Master 0* provides parameters that are applicable to each axis of the robot in question.

Parameter description

The following table contains the parameters that belong to the type *Uncalibrated Control Master 0* in the topic *Motion*.

Common parameters

Cfg name	Parameter name	Description
`Kp`	Kp, Gain Position Loop	The amplification of the position control, e.g. 15. A high value will give a stiff axis that quickly assumes its new position. The value should be large without inducing overshoot in the position or oscillations of the axis.
`Kv`	Kv, Gain Speed Loop	The amplification of the velocity control, e.g. 2. A high value gives better high frequency stiffness, better response speed and low overshoot. If the value is too high the axis will vibrate.
`Ti`	Ti Integration Time Speed Loop	Integration time in the speed regulation loop. The lower the value of *Ti Integration Time Speed Loop*, the better tracking and disturbance rejection. Too low value may case oscillation or noise.
`speed_max_n`	Speed Max Uncalibrated	Max speed for uncalibrated axis (rad/s on motor side).
`acc_max_n`	Acceleration Max Uncalibrated	Max acceleration for uncalibrated axis (rad/s2 on motor side) Recommended value: *Nominal Acceleration * Transmission Gear Ratio*.
`dec_max_n`	Deceleration Max Uncalibrated	Max deceleration for uncalibrated axis (rad/s2 on motor side) Recommended value: *Nominal Deceleration * Transmission Gear Ratio*.

This page is intentionally left blank

9 Hardware

9.1 Configuration of the drive system

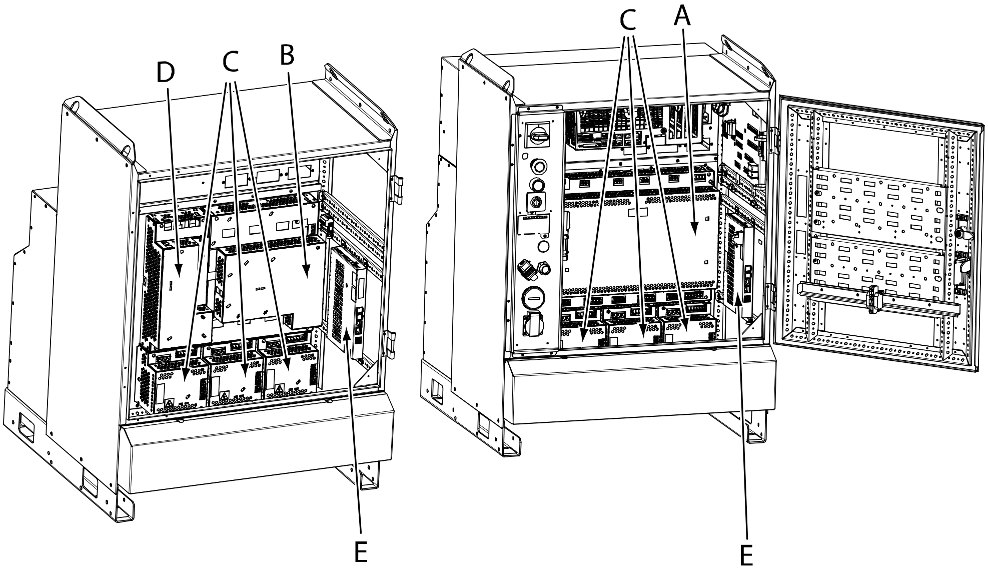
General

The IRC5 Controller contains one Main Drive Unit and up to three Additional Drive Units, and in some cases an Additional Rectifier Unit. The allowed combinations of these, depending on the robot type, is specified below.

The robot system may also be equipped with up to three additional drive modules, which are described in *Product manual - IRC5*.

Location

The drive system is located in the Single Cabinet Controller as shown below.



xx1000000001

A	Main Drive Unit MDU-790A (for large robots)
B	Main Drive Unit MDU-430A (for small robots)
C	Additional Drive Units (for additional axes)
D	Additional Rectifier Unit (only used for additional axes in combination with small robots)

DC bus cables

Between the units are fitted DC bus cables, which are specified below:

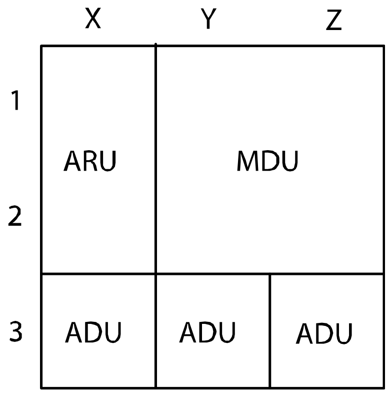
Description	Art. no.	Note
DC bus cable	3HAC032612-001	Between Main Drive Unit MDU-790A and Additional Drive Units.
DC bus cable	3HAC036612-001	Between Additional Rectifier Unit and first Additional Drive Unit.
DC bus cable	3HAC036612-002	Between Additional Rectifier Unit and second Additional Drive Unit.

Continues on next page

Description	Art. no.	Note
DC bus cable	3HAC036612-003	Between Additional Rectifier Unit and third Additional Drive Unit.

IRB 120, 140, 1410, 260, 360, 1600

The following illustration shows the drive units. The table specifies which units may be fitted in which positions.



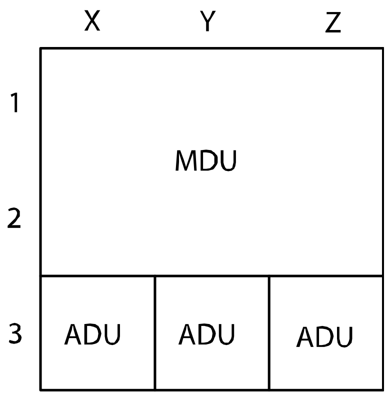
en0800000293

Pos.	Identification	Description	Art. no.	Note
Y1, Y2, Z1, Z2	DSQC 406	Main Drive Unit MDU-430A	3HAC035301-001	
X1, X2	DSQC 417	Additional Rectifier Unit ARU-430A	3HAC035381-001	Required if any Additional Drive Unit is used.
X3	DSQC 664	Additional Drive Unit ADU-790A	3HAC030923-001	For first additional axis
Y3	DSQC 664	Additional Drive Unit ADU-790A	3HAC030923-001	For second additional axis
Z3	DSQC 664	Additional Drive Unit ADU-790A	3HAC030923-001	For third additional axis

Continues on next page

IRB 2400, 2600, 4400, 4600, 6600, 6620, 6640, 6650, 6660, 660, 7600, 460, 760

The following illustration shows the drive units. The table specifies which units may be fitted in which positions.



en0800000292

Pos.	Identification	Description	Art. no.	Note
X1, X2, Y1, Y2, Z1, Z2	DSQC 663	Main Drive Unit MDU-790A	3HAC029818-001	
X3	DSQC 664	Additional drive unit ADU-790A	3HAC030923-001	For first additional axis
Y3	DSQC 664	Additional drive unit ADU-790A	3HAC030923-001	For second additional axis
Z3	DSQC 664	Additional drive unit ADU-790A	3HAC030923-001	For third additional axis

Drive unit connection

The following table shows the drive unit connection for each drive unit when using configuration template files for stand alone axes.

When using a template file, a power stage is connected to a physical output. The label of this output in the electrical circuit diagram is shown in the column "Designation in circuit diagram".

Main Drive Unit	Template file name (drive unit name)	Power stage	Designation in circuit diagram
MDU-430C DSQC 431 Used in IRC5 Compact for IRB 120	M1 (DMX) [a] M2 (DMX) M3 (DMX) M4 (DMX) M5 (DMX) M6 (DMX)	INV_14_20 INV_14_20 INV_6_8 INV_6_8 INV_14_20 INV_6_8	M3 (U,V,W) [b] M1 (U,V,W) M6 (U,V,W) M4 (U,V,W) M2 (U,V,W) M5 (U,V,W)

Continues on next page

Main Drive Unit	Template file name (drive unit name)	Power stage	Designation in circuit diagram
MDU-430A DSQC 406 Used for IRB 120, 140	M1 (DMX)[a] M2 (DMX) M3 (DMX) M4 (DMX) M5 (DMX) M6 (DMX)	INV_14_20 INV_14_20 INV_6_8 INV_6_8 INV_14_20 INV_6_8	M3 (U,V,W)[b] M1 (U,V,W) M6 (U,V,W) M4 (U,V,W) M2 (U,V,W) M5 (U,V,W)
MDU-430A DSQC 406 Used for IRB 1400, 1600	M1 (DMX)[a] M2 (DMX) M3 (DMX) M4 (DMX) M5 (DMX) M6 (DMX)	INV_14_20 INV_14_20 INV_6_8 INV_6_8 INV_14_20 INV_6_8	M1 (U,V,W)[b] M2 (U,V,W) M4 (U,V,W) M6 (U,V,W) M3 (U,V,W) M5 (U,V,W)
MDU-430A DSQC 406 Used for IRB 360	M1 (DMX)[a] M2 (DMX) M3 (DMX) - M5 (DMX) -	INV_14_20 INV_14_20 INV_14_20 - INV_6_8 -	M2 (U,V,W)[b] M1 (U,V,W) M3 (U,V,W) - M4 (U,V,W) -
MDU-430A DSQC 406 Used for IRB 260	M1 (DMX)[a] M2 (DMX) M3 (DMX) - - M6 (DMX)	INV_14_20 INV_14_20 INV_14_20 - - INV_6_8	M1 (U,V,W) M2 (U,V,W)[b] M3 (U,V,W) - - M6 (U,V,W)
MDU-790A DSQC 663 Used for IRB 2400	M1 (DMX)[a] M2 (DMX) M3 (DMX) M4 (DMX) M5 (DMX) M6 (DMX)	INV_31_54 INV_17_26 INV_31_54 INV_17_26 INV_31_54 INV_17_26	M1 (U,V,W)[b] M2 (U,V,W) M4 (U,V,W) M6 (U,V,W) M3 (U,V,W) M5 (U,V,W)
MDU-790A DSQC 663 Used for IRB 2600, 4400, 4600, 6600, 6620, 6640, 6650, 6660, 7600	M1 (DMX)[a] M2 (DMX) M3 (DMX) M4 (DMX) M5 (DMX) M6 (DMX)	INV_31_54 INV_17_26 INV_31_54 INV_17_26 INV_31_54 INV_17_26	M1 (U,V,W)[b] M4 (U,V,W) M2 (U,V,W) M5 (U,V,W) M3 (U,V,W) M6 (U,V,W)

Continues on next page

Main Drive Unit	Template file name (drive unit name)	Power stage	Designation in circuit diagram
MDU-790A DSQC 663 Used for IRB 660	M1 (DMX)[a] - M3 (DMX) - M5 (DMX) M6 (DMX)	INV_31_54 - INV_31_54 - INV_31_54 INV_17_26	M1 (U,V,W)[b] - M2 (U,V,W) - M3 (U,V,W) M6 (U,V,W)
ADU-790A	-	INV_30_55	-

[a] X= drive module number.

[b] Phase R,S,T (U,V,W).

For details about the connection pins, see the circuit diagram in *Product manual - IRC5*.

Note

The Main Drive Unit can handle a maximum of 3 axes for each of EXC1 and EXC2. Measurement nodes 1, 2 and 3 use EXC1 and measurement nodes 4, 5, 6 and 7 use EXC2. See *Serial measurement cables and connections on page 187*.

9.2 Transformers

Overview

The transformer is used to transform the incoming voltage to the voltage used in the cabinet. The selection of transformer depends on the selection of primary voltage and drive units.

Voltage alternative

The transformers are reversible to following primary voltage alternatives.

- 200 V
- 220 V
- 400 V
- 440 V
- 480 V
- 500 V
- 600 V

Technical data

The following table details the standard option transformers.

Robot type	Primary voltage (V)	Effect (kVA)
140, 260, 360, 1410, 1600, 2400, 2600 4400	200-600 V	4.2 kVA
4600, 66xx, 7600, 460, 760	< 400 V 480 V > 480 V	13 kVA 1.2 kVA, for electronics 13 kVA

9.3 Drive units

Overview

A Main Drive Unit (MDU) consists of 6 power stages.

An Additional Drive Unit (ADU) consists of one power stage.

Additional axes in combination with a low voltage Main Drive Unit, requires an Additional Rectifier Unit (ARU) to supply the Additional Drive Units.

Drive Unit voltage

The following table describes the input voltage and the Dc-bus voltage for the different drive units. See also *Requirements for high voltage motors on page 178*.

Drive Unit Type	Output voltage to motor (V_{rms}) [a]	Max dc bus voltage (V_{rms})
MDU-430A DSQC 406	• nominal 234 V • min 198 V	430 V
MDU-790A DSQC 663 for IRB 2400, 2600, 4400	• nominal 234 V • min 198 V	430 V
ARU-430A DSQC 417	• nominal 234 V • min 198 V	430 V
ADU-790A DSQC 664 for IRB 2400, 2600, 4400	• nominal 234 V • min 198 V	430 V
MDU-790A DSQC 663 for IRB 4600, 660, 66XX, 7600	• nominal 430 V • min 320 V	790 V
ADU-790A DSQC 664 for IRB 4600, 660, 66XX, 7600	• nominal 430 V • min 320 V	790 V

[a] defined as line to line.

Drive Unit Current

The following table describes the current for the different power stages. For a list of which power stages are used by which drive unit, see *Drive unit connection on page 167*.

Power stage	Rated current (A_{rms}) [a]	Time limited current (A_{rms}) [b]	Max current (A_{rms}) [c]	Time limit for max current (s) [d]
INV_6_8	6.0	8.25	8.3	unlimited
INV_14_20	13.5	17.4	19.6	30
INV_17_26	17	23	26	10
INV_31_54	31	48	54	8
INV_30_55	30	39	55	3

Continues on next page

[a] Max current for zero speed in indefinite time.

[b] Max current for zero speed in 3 seconds.

[c] Max current during acceleration or deceleration during a limited time (specified by [d]).

[d] Max time for max current during acceleration or deceleration.

9.4 Measurement System

Overview

This system can control up to nine axes at the same time, and measure another five axes.

Axis computer board

The drive module is equipped with one axis computer board. From a connector on the front of the cabinet, serial measurement links are connected to the axis computer.

Serial measurement links

Each drive module has two serial measurement links for measurement boards. The connectors in the front of the drive module are marked, **Measurement system 1** and **Measurement system 2.** These serial links are ring circuits, which means that if there is more than one board on the same link, the output from **Serial Measurement Board 1** is connected to the input on **Serial Measurement Board 2.** See *Serial Measurement Link examples on page 174*.

Serial Measurement Board

The standard SMB has seven resolver inputs. These inputs can be used as seven different nodes where the node number normally is equal to the axis number e.g. axis 1 to node 1.

Back-up battery

A back-up battery supplies the SMB with power during power failure. If an axis is moved a small distance during power off, the system is ready for operation, and no synchronization is needed after power on.

Features

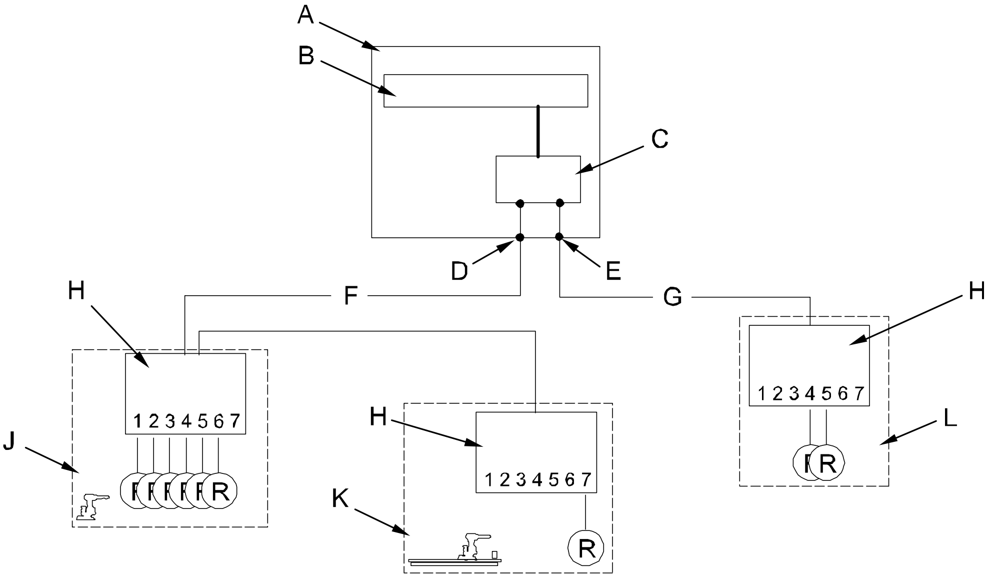
Specifications for the measurement system:

- Each drive module can handle up to four SMBs divided on two serial links.
- Each serial link can handle up to seven axes.
- Each node 1 - 7 may only be used once on each link.

9.5 Serial Measurement Link examples

1 + 2 Additional axes

The following is an example of a setup with three serial measurement boards on two measurement links, e.g *Trackmotion*.



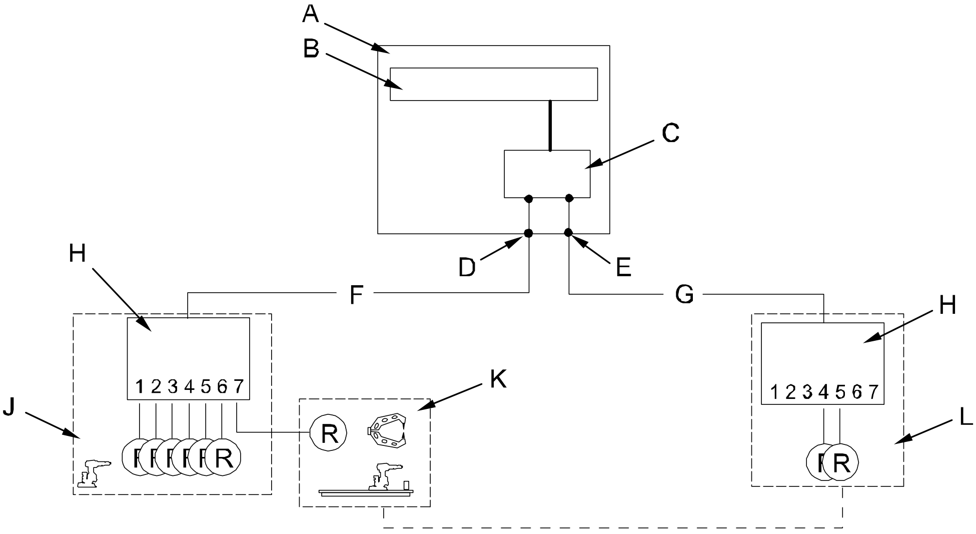
seriematslin

A	IRC5 Controller
B	Main Computer
C	Axis Computer
D	Serial Measurement Link 1 connector XS.2
E	Serial Measurement Link 2 connector XS.41
F	Serial Measurement Link 1
G	Serial Measurement Link 2
H	Serial Measurement Board
J	Six axes Robot system
K	Trackmotion
L	Axes 8-9
R	Resolvers

Continues on next page

1 + 2 Additional axes

The following is an example of a setup with two serial measurement boards on two measurement links, e.g *Servo Gun* or *Trackmotion*. If both Servo Gun and Trackmotion are to be used, the Trackmotion is connected to serial measurement link 2 and resolver node 1.



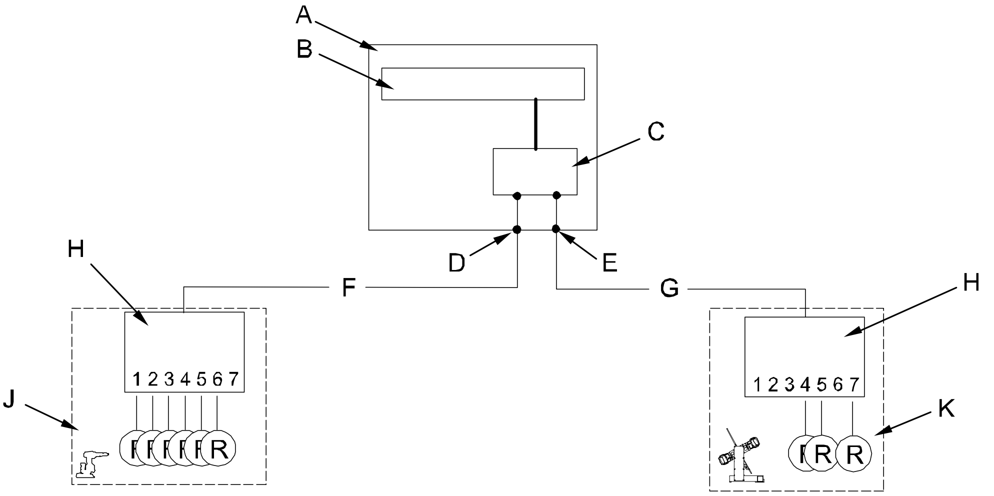
seriematslin

A	IRC5 Controller
B	Main Computer
C	Axis Computer
D	Serial Measurement Link 1 connector XS.2
E	Serial Measurement Link 2 connector XS.41
F	Serial Measurement Link 1
G	Serial Measurement Link 2
H	Serial Measurement Board
J	Six axes Robot system
K	Servo Gun
L	Axes 8-9
R	Resolvers

Continues on next page

Three Additional axes

The following is an example of a setup with two serial measurement boards on two measurement links, e.g *3 Axes Positioner*.



seriematslin

A	IRC5 Controller
B	Main Computer
C	Axis Computer
D	Serial Measurement Link 1 connector XS.2
E	Serial Measurement Link 2 connector XS.41
F	Serial Measurement Link 1
G	Serial Measurement Link 2
H	Serial Measurement Board
J	Six axes Robot system
K	Three axes Positioner
R	Resolvers

9.6 Equipment for additional axes

Overview

A number of parts needed to install and operate additional axes are available from ABB.

Motor units and gear units offer

The offer consists of:

- Motors
- Motors with gear boxes
- SMB boxes
- Cables
- Axis selectors

For more information, see *Product specification - Motor Units and Gear Units* and *Product manual - Motor Units and Gear Units*.

9.7 Motors

Overview

The motor units sold by ABB are specifically designed for ABB's robots and can be used for peripherals requiring power-steered motors that are synchronized with the robot movements. The motor units are designed for optimal performance and to facilitate installation and application.

Note

Before a motor is acquired, read also the information on how to calculate the correct motor data, see *Simple dimensioning of the motor on page 180*.

Motor description

Motor shall be a permanent magnet servo-motor of synchronous type intended for three-phase sinusoidal AC voltage, coupled in star (Y) connection.

- the motor should preferably be winded as class F according to IEC 85.
- dielectric strength minimum 1600 V. For low voltage motors (IRB 140 - 4400) connected to DM. For high voltage motors (IRB 6600 and 7600) connected to DM, see *Requirements for high voltage motors on page 178*
- Measurement signal cables must be separated from power cables and cables from temperature sensor and brake.

Requirements for high voltage motors

Third party driveline components used as external equipment on the IRB x6xx products must withstand the voltage stress levels as described in the following.

These data are valid for high voltage motors connected to the drive units:

- High Voltage Main Drive Unit DSQC 663
- High Voltage Additional Drive Unit DSQC 664

The maximum allowed cable length is 30m. Rise time is expressed as an indicative value at motor terminals.

Converter specifics	
Voltage (Pulse-Width Modulated)	400-480 VAC
DC link maximum voltage	790 VDC (including tolerance: 825 VDC)
Switching frequency	4 kHz
System specifics	
Rise time / dU/dt (indicative value)	0.2 microsec (as defined in IEC 60034-25) / 9 kV/microsec
Requirement for drive line components	
Insulation strength	According to IEC 60034 (i.e. >2000 V)
Voltage stress withstand capability (including PD deterioration effects)	Above withstand level B according to IEC 60034-25, Figure 17 Chapter 7

Continues on next page

Thermal protection

The temperature sensor normally used is of type PTC resistor. A high resistance or open circuit indicates that the temperature of the motor exceeds the rated level. If temperature sensor is not used, the circuit must be strapped. If more than one motor is used, all PTC resistors are connected in series.

The system input characteristics are:

- High temperature >3500 ohm
- Low temperature <3500 ohm

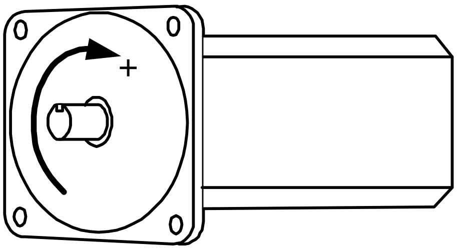
Note

For F class winding with maximum temperature of 155ºC, Siemens B59135-M155_A70 can be used.

Motor connection

Positive electric rotation R ->S ->T -> (U, V, W) results in positive mechanical rotation defined as **clock wise** direction, seen from the drive shaft side. See illustration below. For connection and cabling for the motor to the controller, for more information see

- *Product manual - IRC5*, chapter *Installation*
- *Product manual - Motor Units and Gear Units*, chapter *Installation*
- *Product manual - IRC5*, chapter *Circuit diagrams*



xx0400001171

Brake

Select a brake with minimum brake torque, sufficiently large to handle emergency stop when axis is moving downwards with maximum gravity. Check that maximum brake torque does not exceed allowed mechanical stress levels.

- Brake release voltage: 24 VDC +/- 10%.

Note

Check brake release voltage at maximum brake (motor) temperature and maximum allowed wear out for the brake.

Motor types

For more information about the recommended motor types from ABB, see section *Equipment for additional axes on page 177*.

9.8 Simple dimensioning of the motor

Overview

Before connecting a motor, read the general description for motors in chapter *Motors on page 178*

Note

This section is used as a rough dimensioning of the motor, so before installing the motor make sure that it is dimensioned by a professional.

Calculate system performance

Either the motor or the drive unit sets the limitations for the system performance.

Value	Description
Kt_{min}	Motor torque constant (Nm/A_{rms}).
I_{max} drive	Max current for the drive unit (A_{rms}). See *Drive units on page 171*.
I_{max} (motor)	Max current for the motor (A_{rms}).
T_0	Average motor torque (Nm).
I_0	Average drive unit current (A_{rms}). See *Drive units on page 171*.

Calculate T_{max} and $T_{average}$ for the drive unit and the motor, then choose the limiting torque.

Criteria	Calculate the minimum value
T_{max}(system)	= min(Kt_{min}*I_{max}(drive unit), Kt_{min}*I_{max}(motor))
$T_{average}$(system)	= min(T_0(motor), Kt_{min}*I_0(drive unit))

Check intermittence

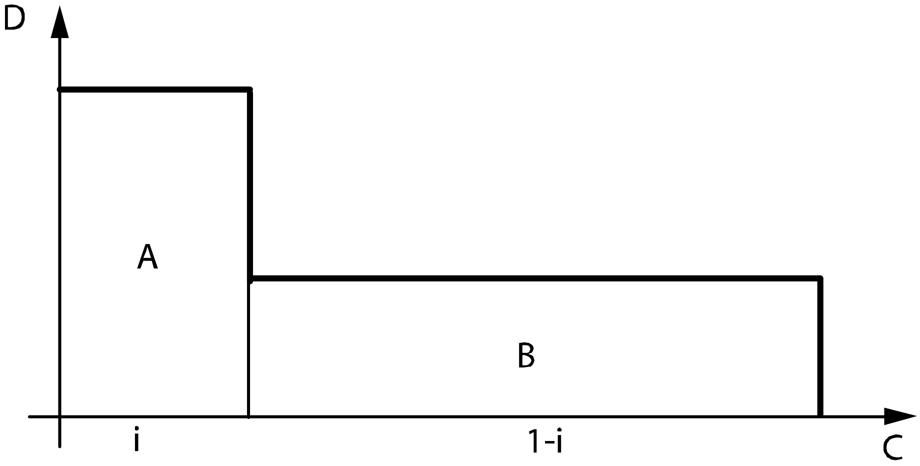
When T_{max} and $T_{average}$ for the system is found, check the thermal load factor. It could be of importance if the additional axis accelerates slowly or if the axis moves with short quick movements without stops. The motor, or the drive unit could be over heated. Observe the planned cycle and calculate the total acceleration time. The other time is treated as static load.

T_{stat} = friction torque + gravitational torque

Value	Description
i	Time in acceleration and deceleration divided by total time
T_{stat}	Static load
1-i	Time in constant speed and standing still (only friction and gravity influences the motor)

Continues on next page

Calculate: T_{rms} = sqrt(T_{max}^2 * i + T_{stat}^2 * (1-i))



xx0500002231

A	Max torque (T_{max})
B	Static torque (T_{stat})
C	Time
D	Torque

Dimensioning

T_{rms} should be lower than $T_{average}$. Otherwise reduce T_{rms} or change one of the components, drive unit or motor

Acceleration performance on arm side could now be calculated:

Acceleration = (T_{max} - GravitationalTorque - Friction) / (Inertia * Transmission)

Deceleration = (T_{max} - GravitationalTorque + Friction) / (Inertia * Transmission)

An alternative is to tune the acceleration and deceleration (parameters: *Nominal acceleration* and *Nominal deceleration*) directly on the external axis and find out if the assessable torque (T_{max}) gives desired performance.

If it is impossible to reach desired performance replace the motor or the drive unit.

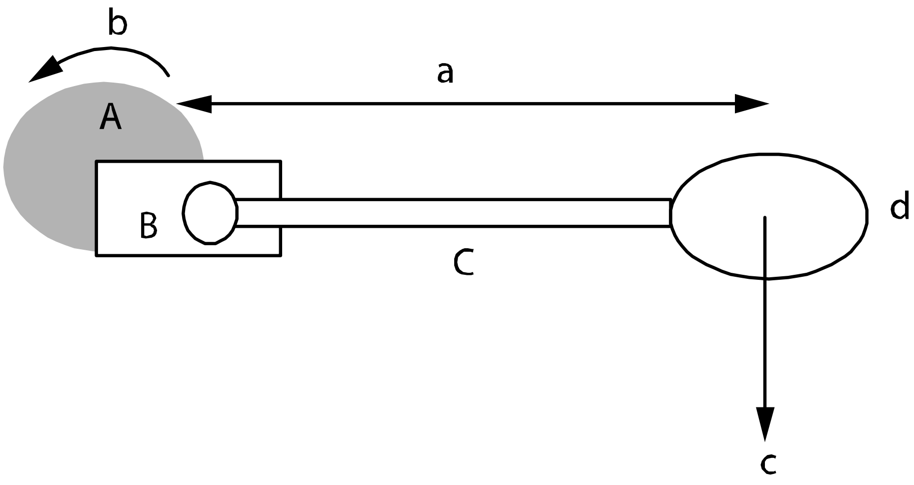
Example

In this example we use worst case performance which means acceleration against the gravity

T_0	5 (Nm)
Kt_{min}	1.0 (Nm/A)
I_{max} (motor)	15 (A)
I_{max} (drive unit)	10 (A)
I_0 (drive unit)	6 (A)
intermittence	0.1

Continues on next page

Transmission (n)	100
Mass (M)	20 (kg)
Friction (F)	2 (Nm)
Gravity constant (g)	9.81 (N/kg)
Length to mass (L)	1.0 (meter)
Motor inertia (J_m)	0.005 (kgm^2)



xx0500002230

a	Length to mass (L)
b	Motor inertia (J_m)
c	Mass (M) * GravityConstant (g)
d	Mass of Arm (M)
A	Motor
B	Gearbox
C	Arm

In this example acceleration needs to be 5 rad/s.

Calculations

Gravitational torque = (M*L*g)/n =(20*1*9.81)/100 = 1.96

T_{stat}= FrictionTorque + GravitationalTorque =2 + 1.96 = 3.96

T_{max} (system) = min (Kt_{min} * I_{max} (drive unit), Kt_{min} * I_{max}(motor) = min(1*10, 1*15)=10

$T_{average}$(system) = min((T_0(motor), Kt_{min}*I_0(drive unit)) = min(5.0, 1*6) = 5.0

T_{rms}= sqrt(T_{max}^2 * i + T_{stat}^2 * (1-i)) = (10^2*0.1+3.96^2*(1-0.1) $)^{0.5}$=4.9

T_{rms} is lower than average. No need to change motor or drive unit.

Total moment of inertia on motor side J= J_m+($M*L^2$)/n^2=0.005+($20*1^2$)/100^2=0.007

Continues on next page

Acceleration = (T_{max} - GravitationalTorque - Friction)/(J*n) = (10-1.96-2)/(0.007*100)=8.6

Deceleration = T_{max} - GravitationalTorque+Friction)/(J*n)=(10-1.96-2)/(0.007*100)=14.3

Both acceleration and deceleration are within the demand.

9.9 Resolvers

Overview

The resolver is integrated in the motors from ABB. The resolver must be approved by ABB for reliable operation.

Approved resolvers

The following resolvers are approved by ABB

Manufacturer	Article numbers
LTN Servotechnik GmbH	LTN RE-21-1-V02, size 21 LTN RE-15-1-V16, size 15
AG	V23401-U2117-C333, size 21
Tamagawa Seiki Co	TS 2640N141E172, size 21 TS 2640N871E172, size 21 TS 2620N871E172, size 15

Resolver specification

Data	Value	Unit
Single speed resolver		
Operating temperature	-25 to +120	°C
Rated input voltage	5	V_{RMS}
Frequency	4	kHz
Primary (EXC)	Rotor	
Secondary (X, Y)	Stator	
Nominal impedance - Primary (stator winding open) Z_{RO} at 4 kHz	>115	Ω
Nominal impedance - Secondary (rotor winding closed) Z_{SS} at 4 kHz	<440	Ω
Transformation ratio	0.5 ± 20%	
Phase shift out-in	0 ± 10	deg
Max error spread	≤ 10	arcmin
Resolver adjustment (COMOFF)	+90 ± 0.5	deg

The resolver has one rotor and two stator windings. The definition of the output signals are:

E(S1, S3) = 0.5 x E(R1, R2) x cos(resolver angle)

E(S2, S4) = 0.5 x E(R1, R2) x sin(resolver angle)

Note

The resolver must be tested together with a robot system to verify that the resolver also functions during battery mode.

Continues on next page

Considerations

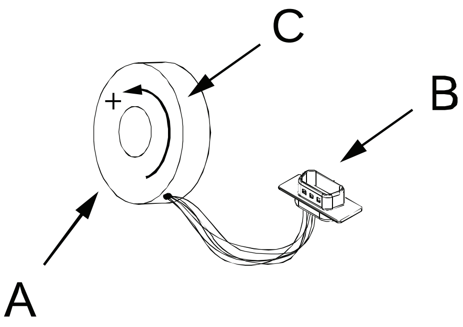
The following technical information must be considered before the installation:

- The maximum allowed resolver cable length is 30 m, from the resolver to the serial measurement board (SMB).
- The total length for all resolver cables using the same excitation must not exceed 70 m.
- A resolver cable consists of six wires. Two wires for excitation, and two wires each for the X and Y signals
- Use a shielded, AWG 24, max 55pF/m cable.
- To avoid disturbances in the signals due to magnetic fields generated by the brake it is recommended to use non-magnetic motor shaft.

Note

The unshielded part of the resolver cable must be as short as possible, less than 100 mm, and be well separated from the motor cables, more than 20 mm.

Resolver connection



xx0400001172

A	Resolver
B	9 pin D-sub
C	Positive motor direction

Normally in ABB motors, resolvers are connected to the internal cable in robot by a 9 pin D-sub connector, with pins at the resolver side.

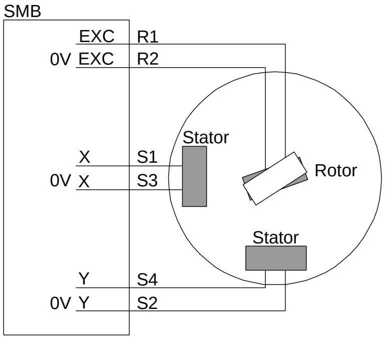
When the motor rotates in a positive direction, the resolver rotates mechanically in a negative direction, as the resolver is mounted at the opposite side of the drive shaft side.

To deliver electrically positive rotation the y-winding connection S2 and S4 has changed place.

9 pin D-sub	SMB input	Resolver connection	Color resolver wires
6	X	S1	Red
1	X 0V	S3	Black
7	Y	S4	Blue
2	Y 0V	S2	Yellow

Continues on next page

9 pin D-sub	SMB input	Resolver connection	Color resolver wires
3	EXC	R1	Red/White
8	EXC 0V	R2	Yellow/White



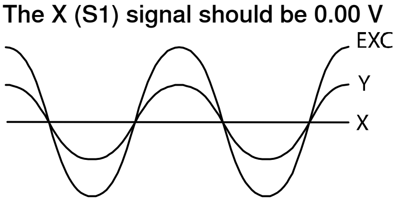
en0400000645

Resolver direction

Motor angle	X (S1)	Y (S4)
0	Maximum in phase with EXC	0
+ 90	0	Maximum in phase with EXC

Commutation

Commutation can be done in several ways. The following method is one of the possible methods.

	Action	Info/Illustration
1	Turn the motor to commutation by feeding positive current into power winding S with T connected to ground (R is not connected). For detailed description, follow the first part of the procedure in *Tuning the commutation offset on page 105*.	The number of different commutation positions that the motor can turn to is the same as the number of pole pairs.
2	Select a resolver commutation position enabling the resolver cables to be routed in the best possible way.	
3	Feed a 4 kHz sinus signal to the EXC (R1) input of the resolver.	
4	Connect an oscilloscope to EXC (R1), X (S1) and Y (S4).	
5	Adjust the commutation position to +90 degrees +/-0.5 degrees by turning the resolver.	The Y (S4) signal should be at max output and with the same phase as the EXC (R1) feeding signal. The X (S1) signal should be 0.00 V  xx0500001401

9.10 Serial measurement cables and connections

Overview

This section details the cables and connection between the resolver and the serial measurement board.

Signal Classes

The cabling must comply with a valid signal class "measurement signals" see *Serial measurement cables and connections on page 187*. The enclosure for external serial measurement board/boards must comply with enclosure class IP54, in accordance with IEC 144 and IEC 529.

Note

It is very important that the noise level on the measurement signals from the additional axes is kept as low as possible, to prevent bad performance, **(keep motor and resolver cables apart)**. Correct shielding and ground connections of cables, measurement boards and resolvers is essential.

Considerations

The **X**, **Y**, **0V X** and **0V Y**:

- Signals are used to connect resolvers to a serial measurement board.

The **EXC** and **0V EXC**:

- are used for common supply for all resolvers, parallel connected.

Resolver:

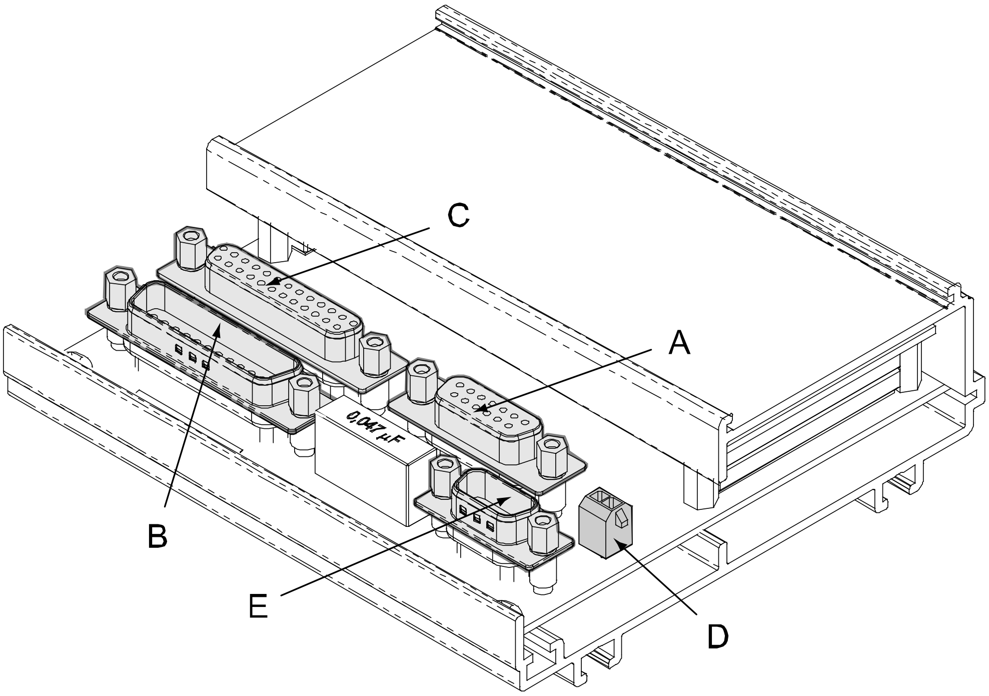
- 1 - 3, should always be connected to EXC 1.
- 4 - 7, should always be connected to EXC 2.

Note

Maximum allowed lenght on the serial measurement cable is 50 meters.

Continues on next page

Illustration



smb_irc5_en0

A	R2.SMB 1-2 (D-sub 15 socket)
B	R2.SMB 1-4 (D-sub 25 pin)
C	R2.SMB 3-6 (D-sub 25 socket)
D	R2.G
E	R2.SMB (D-sub 9 pin)

Connections to SMB DSQC 562

Contact point	R2.G	R2.SMB	R2.SMB 1-2	R2.SMB 1-4	R2.SMB 3-6
1	+BAT	GND	GND	GND	GND
2	0V BAT	-	0V EXC2	X1	X4
3		0V	0V EXC1	Y1	Y4
4		SDO-N	Y7	X2	X5
5		SDI-N	X7	Y2	Y5
6		-	Y1	0V EXC1	0V EXC2
7		+24V	X1	0V EXC1	0V EXC2
8		SDO	-	0V EXC1	0V EXC2
9		SDI	EXC2	X3	X6
10			EXC1	Y3	Y6
11			0V Y7	X4	X3
12			0V X7	Y4	Y3
13			0V Y1	0V EXC2	0V EXC1

Continues on next page

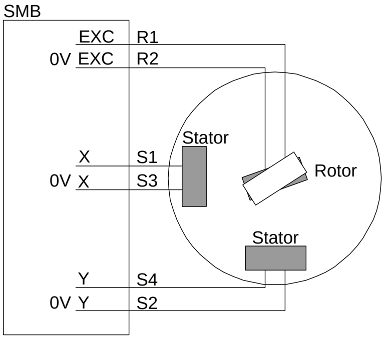
Contact point	R2.G	R2.SMB	R2.SMB 1-2	R2.SMB 1-4	R2.SMB 3-6
14			0V X1	0V X1	0VX4
15				0V Y1	0V Y4
16				0V X2	0V X5
17				0V Y2	0V Y5
18				EXC1	EXC2
19				EXC1	EXC2
20				EXC1	EXC2
21				0V X3	0V X6
22				0V Y3	0V Y6
23				0V X4	0V X3
24				0V Y4	0V Y3
25				EXC2	EXC1

Explanation

Term	Description
SDO	serial communication output
SDI	serial communication input
+BAT	Battery +
0V BAT	Battery 0V
BATLD	Not to be used
BATSUP	Not to be used
EXC1	excitation power to resolver 1, 2, 3
EXC2	excitation power to resolver 4, 5, 6, (7)
+24V	24 V power
0 V	0 V power
X1	Input x-stator node 1
Y1	Input y-stator node 1

Illustration

The connection point on the resolver corresponds to the connection table above.



en0400000645

Continues on next page

Example

To connect from resolver to SMB, use input 7 (i.e. node 7). Connect to contact R2.SMB 1-2.

Signals	Contact point SMB	Contact point Resolver
EXC 2	9	3
EXC 2, 0 V	2	8
X7	5	6
X7, 0 V	12	1
Y7	4	7
Y7, 0 V	11	2

Index

Contact us

ABB AB, Robotics
Robotics and Motion
S-721 68 VÄSTERÅS, Sweden
Telephone +46 (0) 21 344 400

ABB AS, Robotics
Robotics and Motion
Nordlysvegen 7, N-4340 BRYNE, Norway
Box 265, N-4349 BRYNE, Norway
Telephone: +47 22 87 2000

ABB Engineering (Shanghai) Ltd.
Robotics and Motion
No. 4528 Kangxin Highway
PuDong District
SHANGHAI 201319, China
Telephone: +86 21 6105 6666

ABB Inc.
Robotics and Motion
1250 Brown Road
Auburn Hills, MI 48326
USA
Telephone: +1 248 391 9000

www.abb.com/robotics