ROBOTICS

Operating manual

IRB 14000

Trace back information:
Workspace R18-1 version a11
Checked in 2018-03-25
Skribenta version 5.2.025

Operating manual

IRB 14000

RobotWare 6.07

Document ID: 3HAC052986-001

Revision: E

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

Overview of this manual

About this manual

This manual contains instructions for daily operation of the IRB 14000 robot system.

Usage

This manual should be used during operation.

Who should read this manual?

This manual is intended for:

- operators
- product technicians
- service technicians
- robot programmers

Prerequisites

The reader should:

- Be familiar with the concepts described in *Operating manual - Getting started, IRC5 and RobotStudio*.
- Be trained in robot operation.

References

References	Document ID
Product manual - Product.ProductName	*Document.ID-1*
Operating manual - Getting started, IRC5 and RobotStudio	*3HAC027097-001*
Operating manual - RobotStudio	*3HAC032104-001*
Operating manual - Troubleshooting IRC5	*3HAC020738-001*
Technical reference manual - System parameters	*3HAC050948-001*
Technical reference manual - RAPID overview	*3HAC050947-001*
Technical reference manual - RAPID Instructions, Functions and Data types	*3HAC050917-001*
Technical reference manual - RAPID kernel	*3HAC050946-001*
Application manual - Controller software IRC5	*3HAC050798-001*
Application manual - MultiMove	*3HAC050961-001*

Revisions

Revision	Description
-	Released with RobotWare 6.01.

Continues on next page

Revision	Description
A	Released with RobotWare 6.03. • Minor corrections • Value in *Coll-Pred Safety Distance on page 60* changed from 0.001 to 0.01. • The predefined signal `Collision_Avoidance` is added. • Added information to section *Collision avoidance on page 39*. • Added section *Lead-through on page 36*. • Applicable ESD-standards added.
B	Released with RobotWare 6.04. • The RAPID instruction `ContactL` has been moved from this manual. It is now described in *Technical reference manual - RAPID Instructions, Functions and Data types*. • Value in *Arm Check Point Speed Limit on page 62* changed from 1.0 to 0.75. • Added *Programming Move-instructions on page 42*. • Added the system parameter *Limit avoidance distance on page 65*. • Minor corrections.
C	Released with RobotWare 6.05. • Added the new section *Collision on page 41*. • Added the new parameter *Friction compensation lead through factor on page 67*. • Updated the parameter *Collision Detection Memory on page 66*. • Updated descriptions of stops in section *Protective stop and emergency stop on page 22*.
D	Released with RobotWare 6.06. • Updated the section *Collision Detection Memory on page 66*.
E	Released with RobotWare 6.07. • Updated the system parameter *Connection Timeout Multiplier on page 59*. • Updated the Limitations section of the system parameter *Arm-Angle Definition on page 63*. • Safety section restructured.

Product documentation

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 can be found via myABB Business Portal, www.myportal.abb.com.

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 corresponding figures (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, for example lubrication, the RAPID language, and system parameters.

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, software).
- How to install included or required hardware.
- How to use the application.
- Examples of how to use the application.

Continues on next page

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 troubleshooters.

1 Safety

1.1 About this chapter

Introduction to safety

This chapter describes safety principles and procedures to be used when a robot or robot system is operated.

It does not cover how to design for safety nor how to install safety related equipment. These topics are covered in the Product Manuals supplied with the robot.

Note

The integrator is responsible for the safety of the final application.

1.2 Applicable safety standards

Note

The listed standards are valid at the time of the release of this document. Phased out or replaced standards are removed from the list when needed.

Standards, EN ISO

The robot system is designed in accordance with the requirements of :

Standard	Description
EN ISO 12100	Safety of machinery - Basic concepts, general principles for design
EN ISO 13849-1	Safety of machinery, safety related parts of control systems - Part 1: General principles for design
EN ISO 13850	Safety of machinery - Emergency stop - Principles for design
EN ISO 10218-1 [i]	Robots for industrial environments - Safety requirements - Part 1 Robot
EN ISO 9787	Manipulating industrial robots, coordinate systems, and motion nomenclatures
EN ISO 9283	Manipulating industrial robots, performance criteria, and related test methods
EN ISO 14644-1 [ii]	Classification of air cleanliness
EN IEC 61000-6-4	EMC, Generic emission
EN IEC 61000-6-2	EMC, Generic immunity
EN IEC 60204-1	Safety of machinery - Electrical equipment of machines - Part 1 General requirements
IEC 60529	Degrees of protection provided by enclosures (IP code)
IEC 61340-5-1:2010	Protection of electronic devices from electrostatic phenomena - General requirements

i See *Deviations from ISO 10218-1:2011 on page 12*.
ii Only robots with protection Clean Room.

Deviations from ISO 10218-1:2011

ISO 10218-1:2011 was developed with conventional industrial robots in mind. Deviations from the standard are motivated for IRB 14000 in the table below. More information about ISO 10218-1 compliance is given in *technote_150918*.

Requirement	Deviation for IRB 14000	Motivation
§5.7.1 Mode selector which can be locked in each position.	The mode selector is implemented in software on FlexPendant.	Automatic and manual mode are usability features for IRB 14000, but not safety features. Locking the operating mode does not contribute to a necessary risk reduction. [i]

Continues on next page

Requirement	Deviation for IRB 14000	Motivation
§5.12.1 Limiting the range of motion by adjustable stops (§5.12.2) or by safety functions (§5.12.3).	IRB 14000 does not have adjustable mechanical stops or provisions to install non-mechanical limiting devices.	The IRB 14000 robot is intended for collaborative applications where contact between robot and the operator is harmless. Limiting the working range is then not necessary for risk reduction. Note that PPE (Personal Protective Equipment) may be required.

i The selector is replaced by a selection through software and user authorities can be set to restrict the use of certain functions of the robot (e.g. access codes).

European standards

Standard	Description
EN 614-1	Safety of machinery - Ergonomic design principles - Part 1: Terminology and general principles
EN 574	Safety of machinery - Two-hand control devices - Functional aspects - Principles for design

Other standards

Standard	Description
ANSI/RIA R15.06	Safety requirements for industrial robots and robot systems
ANSI/UL 1740	Safety standard for robots and robotic equipment
CAN/CSA Z 434-14	Industrial robots and robot Systems - General safety requirements
ANSI/ESD S20.20:2007	Protection of Electrical and Electronic Parts, Assemblies and Equipment (Excluding Electrically Initiated Explosive Devices)

1.3 Safety actions

1.3.1 Stop the system

Overview

Press the emergency stop button if you need to stop the robot and its external equipment to protect equipment or personnel.

The FlexPendant emergency stop button

xx1400001445

A	Emergency stop button

Other emergency stop devices

The plant designer may have placed additional emergency stop devices in convenient places. Refer to your plant or cell documentation to find out where these are placed.

1.3.2 Recover from emergency stops

Overview

Recovering from an emergency stop is a simple but important procedure. This procedure ensures that the manipulator system is not returned to production while maintaining a hazardous condition.

Reset the latch of emergency stop buttons

All push-button style emergency stop devices have a latching feature that must be released in order to remove the emergency stop condition of the device.

In many cases this is done by twisting the push-button as marked, but there are also devices where you pull the button to release the latch.

Reset automatic emergency stop devices

All automatic emergency stop devices also have some kind of latching feature that must be released. Please consult your plant or cell documentation to see how your manipulator system is configured.

Recover from emergency stops

	Action
1	Make sure the hazardous situation that resulted in the emergency stop condition no longer exists.
2	Locate and reset the device or devices that gave the emergency stop condition.
3	Press the Motors On button on the Quickset menu on the FlexPendant to recover from the emergency stop condition.

1.3.3 Extinguishing fires

Precautions

In case of a fire always make sure both you and your co-workers are safe before performing any fire extinguishing activities. In case of injury always make sure these are treated first.

Select fire extinguisher

Always use carbon dioxide extinguishers when extinguishing fires in electrical equipment such as the robot or the controller. Do not use water or foam.

1.4 Working in a safe manner

1.4.1 For your own safety

General principles

A few simple principles should be followed in order to operate the robot safely:

- Always operate the robot in manual mode if personnel are inside safeguarded space.
- Always bring the FlexPendant along when you enter safeguarded space so that robot control is in your hands.
- Watch out for rotating or moving tools such as milling cutters and saws. Make sure those are stopped before you approach the robot.
- Watch out for hot surfaces both on work pieces as well as on the robot. The robot motors and other parts can be hot.
- Watch out for grippers and objects gripped. If the gripper is opened the work piece could fall and cause injuries or damage equipment. The gripper can be very powerful and can also cause injuries if not operated in a safe manner.
- Watch out for hydraulic and pneumatic systems and live electric parts. Even with power off residual energy in such circuits can be very dangerous.

Disconnected FlexPendant

A disconnected FlexPendant should be stored in such a way that it cannot be mistaken for being connected to the controller.

Custom connections for FlexPendant

Any means of connecting the FlexPendant with other than the supplied cable and its standard connector must not render the emergency stop button inoperative.

Always test the emergency stop button to make sure it works if a custom connection cable is used.

Personal protective equipment

Always use suitable personal protective equipment, based on the risk assessment for the robot installation.

1.4.2 About the manual mode

What is the manual mode?

In manual mode the manipulator movement is under manual control.

The manual mode is used when creating and verifying programs, and when commissioning a manipulator system.

Active safeguard mechanisms

The safety stop mechanism and the Cartesian speed supervision are always active, both while operating in manual and automatic mode.

Operating speed

In manual reduced speed mode the movement is limited to 250 mm/s.

1.4.3 About the automatic mode

What is the automatic mode?

The automatic mode is used for running the robot program in production.

Active safeguard mechanisms

The safety stop mechanism and the Cartesian speed supervision are always active, both while operating in manual and automatic mode.

Coping with process disturbances

Process disturbances may not only affect a specific manipulator cell but an entire chain of systems even if the problem originates in a specific cell.

Extra care must be taken during such a disturbance since that chain of events may create hazardous operations not seen when operating the single manipulator cell. All remedial actions must be performed by personnel with good knowledge of the entire production line, not only the malfunctioning manipulator.

Process disturbance examples

A manipulator picking components from a conveyer might be taken out of production due to a mechanical malfunction, while the conveyer must remain running in order to continue production in the rest of the production line. This means, of course, that extra care must be taken by the personnel preparing the manipulator in close proximity to the running conveyor.

A welding manipulator needs maintenance. Taking the welding manipulator out of production also means that a work bench as well as a material handling manipulator must be taken out of production to avoid personnel hazards.

1.5 Safety terminology

1.5.1 Safety signals in the manual

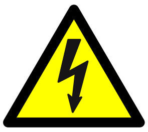
Introduction to safety signals

This section specifies all dangers that can arise when doing the work described in the user manuals. Each danger consists of:

- A caption specifying the danger level (DANGER, WARNING, or CAUTION) and the type of danger.
- A brief description of what will happen if the operator/service personnel **do not** eliminate the danger.
- Instruction about how to eliminate danger to simplify doing the work.

Danger levels

The table below defines the captions specifying the danger levels used throughout this manual.

Symbol	Designation	Significance
 xx0200000022	DANGER	Warns that an accident *will* occur if the instructions are not followed, resulting in a serious or fatal injury and/or severe damage to the product. It applies to warnings that apply to danger with, for example, contact with high voltage electrical units, explosion or fire risk, risk of poisonous gases, risk of crushing, impact, fall from height, and so on.
 xx0100000002	WARNING	Warns that an accident *may* occur if the instructions are not followed that can lead to serious injury, possibly fatal, and/or great damage to the product. It applies to warnings that apply to danger with, for example, contact with high voltage electrical units, explosion or fire risk, risk of poisonous gases, risk of crushing, impact, fall from height, etc.
 xx0200000024	ELECTRICAL SHOCK	Warns for electrical hazards which could result in severe personal injury or death.
 xx0100000003	CAUTION	Warns that an accident may occur if the instructions are not followed that can result in injury and/or damage to the product. It also applies to warnings of risks that include burns, eye injury, skin injury, hearing damage, crushing or slipping, tripping, impact, fall from height, etc. Furthermore, it applies to warnings that include function requirements when fitting and removing equipment where there is a risk of damaging the product or causing a breakdown.
 xx0200000023	ELECTROSTATIC DISCHARGE (ESD)	Warns for electrostatic hazards which could result in severe damage to the product.

Continues on next page

Symbol	Designation	Significance
 xx0100000004	NOTE	Describes important facts and conditions.
 xx0100000098	TIP	Describes where to find additional information or how to do an operation in an easier way.

1.5.2 Protective stop and emergency stop

Overview

The protective stops and emergency stops are described in the product manual for the robot.

2 Introduction to the IRB 14000 robot system

2.1 What is IRB 14000?

The IRB 14000 robot

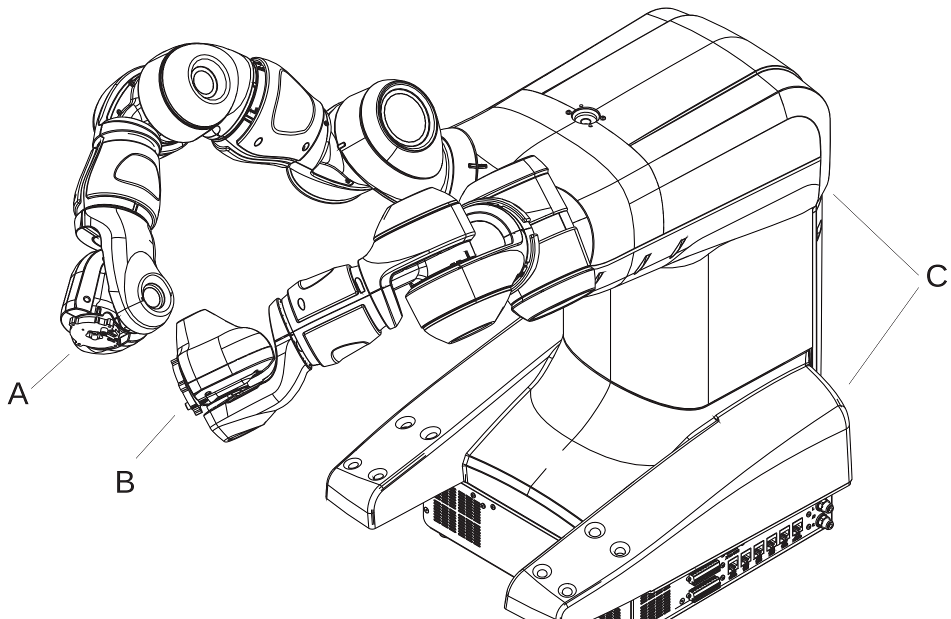
The IRB 14000 robot is a two-armed industrial robot with integrated controller. Each arm has seven axes, which gives an extra degree of freedom compared to traditional 6-axis robots.

The IRB 14000 integrated controller

The IRB 14000 integrated controller is based on the standard IRC5 controller, and contains all functions needed to move and control the robot.

The robot control software, RobotWare, supports every aspect of the robot system, such as motion control, development and execution of application programs, communication etc.

Illustration



xx1500000008

A	Right arm
B	Left arm
C	Integrated controller

2.2 What is a FlexPendant?

Introduction to the FlexPendant

The FlexPendant is a hand held operator unit used to perform many of the tasks involved when operating a robot system, such as running programs, jogging the manipulator, modifying robot programs and so on.

The FlexPendant consists of both hardware and software and is a complete computer in itself.

For more information on using the FlexPendant, see the section *Navigating and handling FlexPendant* in *Operating manual - IRC5 with FlexPendant*.

xx1500000702

2.3 What is RobotWare?

Introduction to RobotWare

RobotWare is a family of software products from ABB Robotics. The products are designed to make you more productive and lower your cost of owning and operating a robot. ABB Robotics has invested many years into the development of these products and they represent knowledge and experience based on several thousands of robot installations.

Product classes

Within the RobotWare family, there are different classes of products:

Product classes	Description
RobotWare-OS	This is the operating system of the robot. RobotWare-OS provides all the necessary features for fundamental robot programming and operation. It is an inherent part of the robot, but can be provided separately for upgrading purposes. For a description of RobotWare-OS, see *Product specification - Controller IRC5*.
RobotWare options	These products are options that run on top of RobotWare-OS. They are intended for robot users that need additional functionality for motion control, communication, system engineering, or applications. For more information on RobotWare options, see *Product specification - Controller software IRC5*. For more detailed information on RobotWare options, see *Application manual - Controller software IRC5*. Note that not all RobotWare options are described in this manual. Some options are more comprehensive and are therefore described in separate manuals.
Process application options	These are extensive packages for specific process application like spot welding, arc welding, and dispensing. They are primarily designed to improve the process result and to simplify installation and programming of the application. The process application options are all described in separate manuals. For more information, see *Product specification - Controller software IRC5*.
RobotWare Add-ins	A RobotWare Add-in is a self-contained package that extends the functionality of the robot system. Some software products from ABB Robotics are delivered as Add-ins. For example track motion IRBT, positioner IRBP, and stand alone controller. The purpose of RobotWare Add-ins is also that a robot program developer outside of ABB can create options for the ABB robot systems, and sell the options to their customers. For more information on creating RobotWare Add-ins, contact your local ABB Robotics representative at www.abb.com/contacts.

2.4 What is RobotStudio?

Overview

RobotStudio is an engineering tool for the configuration and programming of ABB robots, both real robots on the shop floor and virtual robots in a PC. To achieve true offline programming, RobotStudio utilizes ABB VirtualRobot™ Technology.

RobotStudio has adopted the Microsoft Office Fluent User Interface. The Office Fluent UI is also used in Microsoft Office. As in Office, the features of RobotStudio are designed in a workflow-oriented way.

With add-ins, RobotStudio can be extended and customized to suit the specific needs. Add-ins are developed using the RobotStudio SDK. With the SDK, it is also possible to develop custom SmartComponents which exceed the functionality provided by RobotStudio's base components.

For more information, see *Operating manual - RobotStudio*.

RobotStudio for real controllers

RobotStudio allows, for example, the following operations when connected to a real controller:

- Installing and modifying systems on RobotWare 6 controllers, using the **Installation Manager**.
- Text-based programing and editing, using the **RAPID Editor**.
- File manager for the controller.
- Administrating the User Authorization System.
- Configuring system parameters.

2.5 What is RobotStudio Online?

Introduction to RobotStudio Online

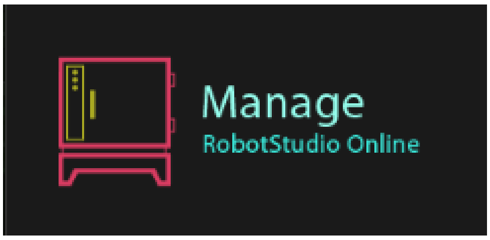
RobotStudio Online is a suite of **Windows Store** applications intended to run on **Windows 10** tablets. It provides functionality for the shop floor commissioning of robot systems.

> **Note**
>
> Some of the functionality requires use of a safety device such as the T10 jogging device or the JSHD4 three position safety device. For more information on T10, see *Operating manual - IRC5 with T10*.

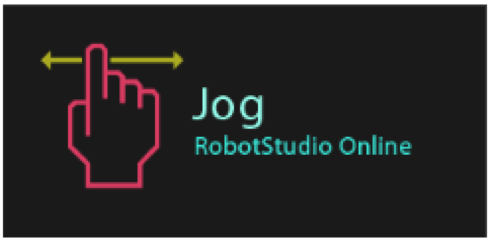
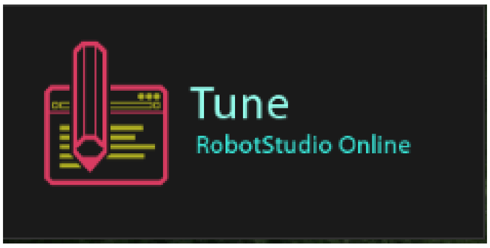
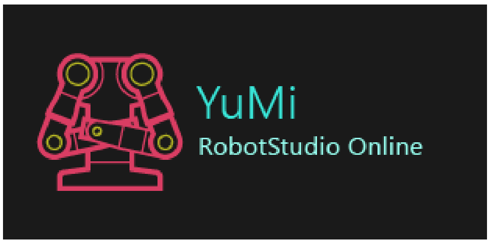
You can run these apps on a tablet that communicates with the robot controller wirelessly. To enable certain functionality, such as entering manual mode and enabling power to the mechanical unit motors, you need a safety device that is connected to the robot using the same plug that alternatively is used to connect the FlexPendant.

The following RobotStudio Online apps are available in the Microsoft Windows Store:

> **Note**
>
> You must have **Windows 8.1** to run these Apps.

RobotStudio Online Apps	Description
 xx1400002047	**Manage** is a tool to manage IRC5 controllers on a network.
 xx1400002049	**Calibrate** is a tool for calibration and definition of frames with IRC5 controllers.

Continues on next page

RobotStudio Online Apps	Description
 xx1400002048	**Jog** is a tool for manual positioning (moving or jogging) with IRC5 controllers.
 xx1400002050	**Tune** is a tool for shop floor editing of RAPID programs with IRC5 controllers.
 xx1400002511	**Operate** is a tool used in production to view the program code.
 xx1500000832	**YuMi** is a tool for programming of the new dual arm and collaborative robot YuMi, IRB 14000, from ABB. It will help the users to get a fast introduction to robot programming using lead-thru and graphical programming.

3 Using the IRB 14000

3.1 Axes and coordinate systems

What is a coordinate system?

A coordinate system defines a plane or space by axes from a fixed point called the origin. Robot targets and positions are located by measurements along the axes of coordinate systems.

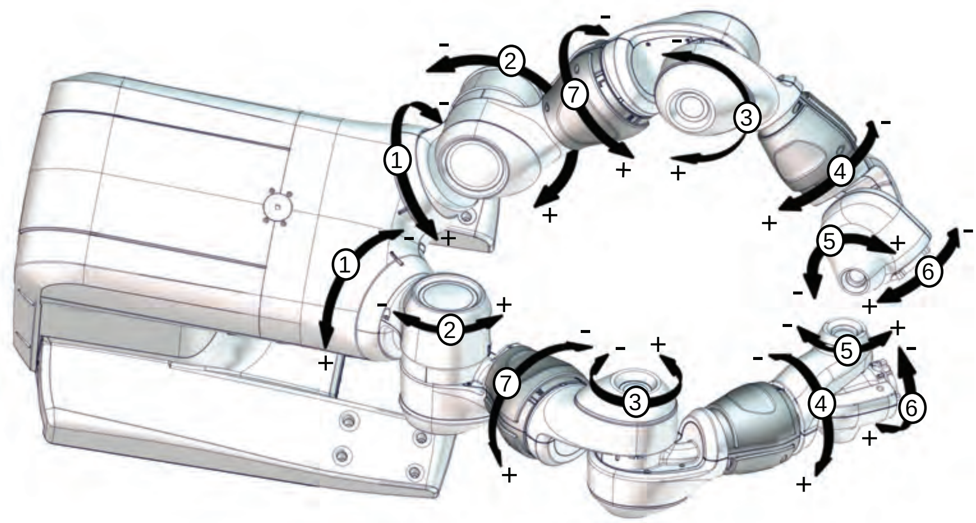
A robot uses several coordinate systems, each suitable for specific types of jogging or programming.

- The *base coordinate system* is located at the base of the robot. It is the easiest one for just moving the robot from one position to another.
- The *work object coordinate system* is related to the work piece and is often the best one for programming the robot.
- The *tool coordinate system* defines the position of the tool the robot uses when reaching the programmed targets.
- The *world coordinate system* that defines the robot cell, all other coordinate systems are related to the world coordinate system, either directly or indirectly. It is useful for jogging, general movements and for handling stations and cells with several robots or robots moved by external axes.
- The *user coordinate system* is useful for representing equipment that holds other coordinate systems, like work objects.

For more information on coordinate systems, see the *Jogging* section in *Operating manual - IRC5 with FlexPendant*.

Axes and joystick directions

The axes of the robot can be jogged manually using the joystick. The following illustration shows the location and movement patterns for each axis.



xx1500000254

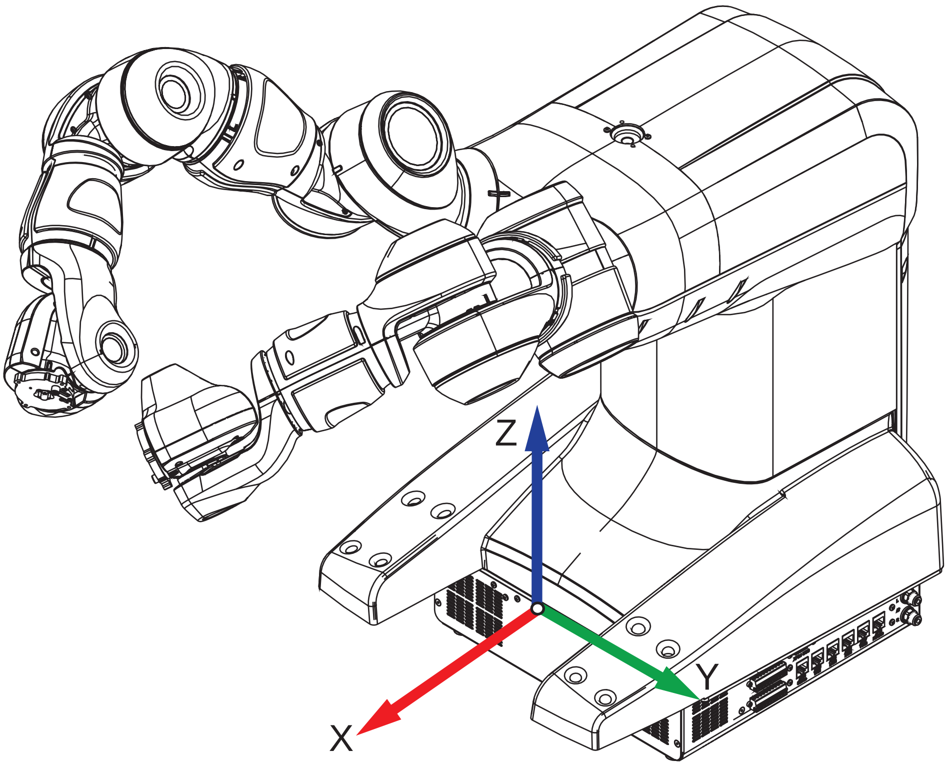
Continues on next page

> **Note**
>
> Note that axis 7 is located between axis 2 and axis 3.

The base coordinate system

The base coordinate system has its zero point in the base of the robot.

When you are standing in front of the robot and jog in the base coordinate system, pulling the joystick towards you will move the robot along the X axis, while moving the joystick to the sides will move the robot along the Y axis. Twisting the joystick will move the robot along the Z axis.



xx1500000007

3.2 Jogging

3.2.1 What is jogging?

Introduction

To jog is to manually position or move robots or external axes. You can only jog in manual mode, but not during program execution. Jogging is disabled in automatic mode.

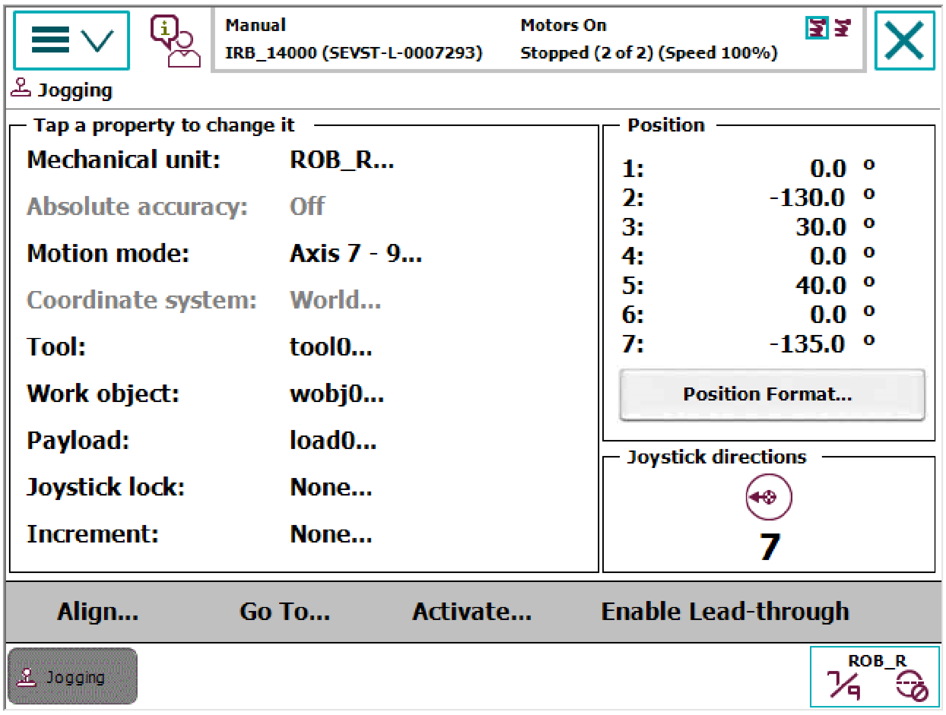
The selected motion mode and/or coordinate system determines the way the robot moves. For more information on how to jog robots, see the *Jogging* section in *Operating manual - IRC5 with FlexPendant*.

Note

This manual only describes the settings that are specific for the IRB 14000.

The jogging window

The jogging functions are found in the Jogging window. The most commonly used are also available under the Quickset menu.



xx1500000006

Tip

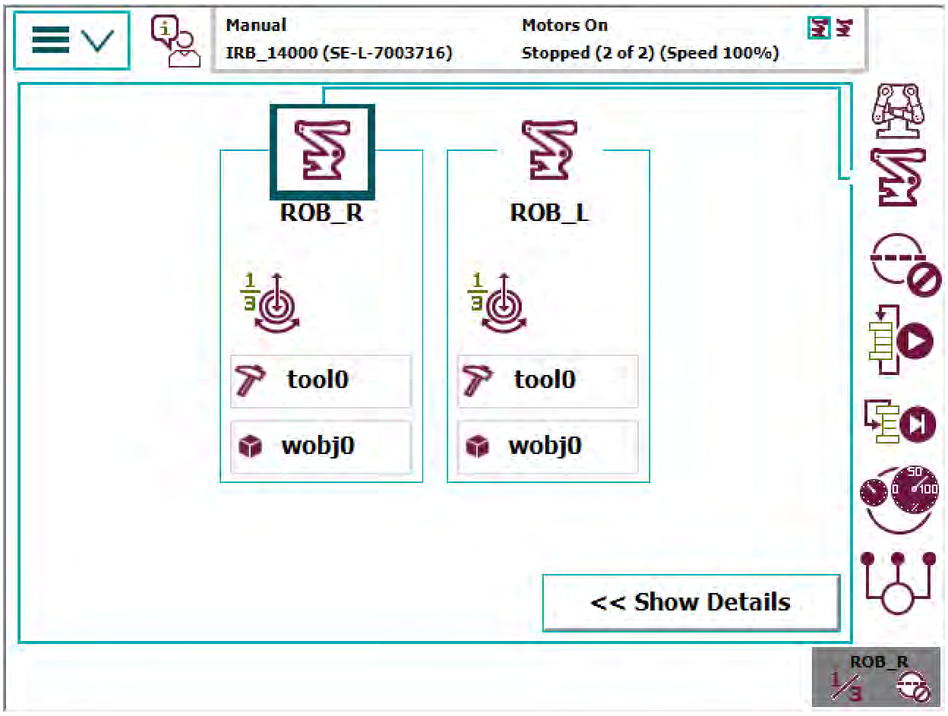
Use the hard buttons on the FlexPendant to toggle between the different motion modes.

Continues on next page

The Quickset menu

The Quickset menu provides a quicker way to change among other things jog properties rather than using the jogging window.

Each button on the menu shows the currently selected property value or setting.



xx1500000004

> **Note**
>
> Note that the operator panel is located under the Quickset menu. For more information on how to switch between manual and automatic mode, see *Operating modes on page 37*.

3.2.2 Motion modes

Standard motion modes

A traditional robot, with four to six axes, has three different motion modes modes: *axis-by-axis*, *linear*, and *reorientation* mode.

- *Axis-by-axis mode* moves one robot axis at a time. The tool center point, and the orientation of the tool, is not monitored. Axis-by-axis mode is used to manually position the robot before switching over to linear mode.
- In *linear motion mode*, the tool center point moves along a straight line in space, in a “move from point A to point B” fashion. The tool center point is monitored and moves in the direction of the selected coordinate system’s axes. The orientation of the tool is fixed throughout the motion.
- In *reorientation mode*, the tool center point is fixed in space and the orientation of the tool is changed. The tool center point is rotated around the direction of the selected coordinate system’s axes.

The arm mode

The IRB 14000 robot, with seven axes, has one additional motion mode, the *arm mode*. All of the other jogging settings are the same as for other robots.

- In *arm mode*, both the tool center point and the orientation of the tool is fixed in space and only the angle of the arm is changed. The tool center point is neither rotated nor moved. See *Jogging in arm mode on page 33*.

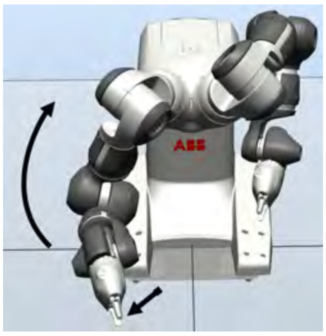
For more information on robot configuration and how the arm angle is calculated, see the data type `confdata` in *Technical reference manual - RAPID Instructions, Functions and Data types*.

Jogging in arm mode

In the pictures below, the robot is jogged in arm mode. Note that both the tool center point and the orientation of the tool are fixed in space and only the angle of the arm is changed.

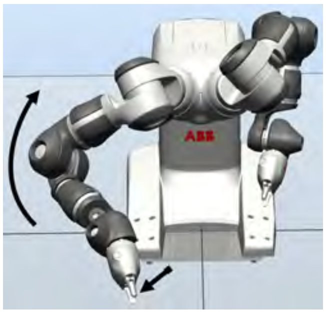
This is useful when programming to avoid singularity, to find the most natural way the robot can reach a given target, and also how to be able to proceed to the next target.

1.



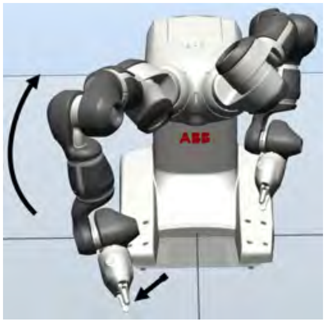
xx1500000009

2.



xx1500000010

3.



xx1500000011

3.2.3 Coordinated jogging

Introduction

The IRB 14000 robot is pre-installed with the RobotWare option *MultiMove coordinated,* which makes it possible to jog the two arms in coordinated mode.

Coordinated jogging has to be setup by creating a coordinated work object. The work object should be setup for the arm that is holding the work piece. The other arm is holding the tool.

When the arm moving the work object is jogged, the other arm that is currently coordinated with the work object will move so that it maintains its relative position to the work object.

For more information about MultiMove and coordinated jogging, see *Application manual - MultiMove*.

Tip

Use the MultiMove wizard in RobotStudio when setting up and programming MultiMove.

Setup coordinated jogging

Use this procedure to setup coordinated jogging.

	Action	Description
1	Create a work object for the arm that is to be coordinated. That work object will be held and moved by the other arm.	

2	Define the data of the work object.
	Set `hobhold` and `ufprog` to `FALSE`, set `ufmec` to the other arm.
		In this example the right arm is holding the work object and the left arm is coordinated to it:
		`PERS wobjdata wobjRight :=`
		`[`**`FALSE`**`, `**`FALSE`**`, `**`"ROB_R"`**`,`
		`[[0,0,0],[1,0,0,0]],`
		`[[0,0,0],[1,0,0,0]]];`

3	Optional, define x, y, z values for the work object and define a tool for the other arm.	
4	Activate coordinated jogging.	*Activate coordinated jogging on page 34*

Activate coordinated jogging

Use this procedure to activate coordinated jogging.

	Action	Description
1	Open the Quickset menu and select the arm that is to be coordinated.	*The Quickset menu on page 32*
2	Activate the previously created work object.	*Setup coordinated jogging on page 34*
3	Select the work object coordinate system.	
4	Select the other arm.	The coordinated arm is now indicated with a flashing frame.

Continues on next page

	Action	Description
5	Jog the arm, the other will follow.	

Deactivate coordinated jogging

Turn off coordination in one of following ways:

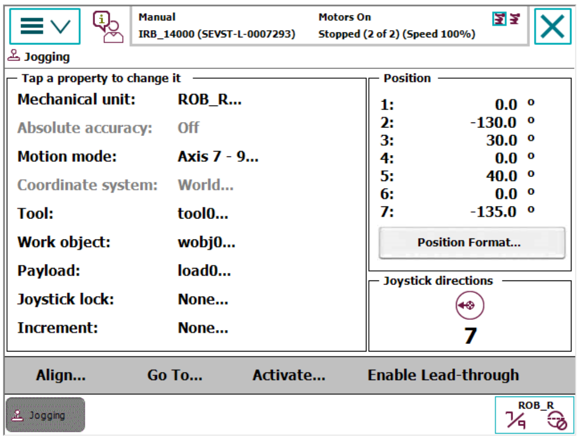
- Click the **Turn coordination off** button on the Quickset menu.
- Deactivate the work object.
- Deactivate the work object coordinate system.

3.3 Lead-through

What is lead-through?

Lead-through means that you can grab the robot arms and move them manually to the desired position, as an alternative to jogging.

Using lead-through

	Action	Note/illustration
1	In the jogging window, tap the button **Enable Lead-through**.	 xx1500000006
2	Gently pull the robot arm to the desired position.	You will feel if an axis reaches its end position. Do not try to force the axis beyond this position.

Note

Lead-through is enabled per robot arm. To enable lead-through for both arms, enable one arm and then select the other arm and enable that too.

Note

If the robot is in motors off mode, it will automatically go to motors on when enabling lead-through.

Note

If lead-through is enabled, it will be temporarily disabled during program execution and jogging. This means that it is possible to combine lead-through, jogging and testing the RAPID program without having to disable lead-through.

Note

When using lead-through, it is important that the load is correctly defined. If the load is heavier than defined, the effect will be the same as if you are pulling the robot arm downwards. If the load is lighter than defined, the effect will be the same as if you are pulling the robot arm upwards.

3.4 Operating modes

Introduction

The IRB 14000 has two operating modes, *Manual mode* and *Automatic mode*.

Note

Note that the IRB 14000 does not go to motors off when changing the operating mode.

Note

The IRB 14000 goes automatically to motors on when jogging or when tapping the play button or step button on the FlexPendant.

What is the manual mode?

In manual mode the manipulator movement is under manual control. The manual mode is used when programming and for program verification.

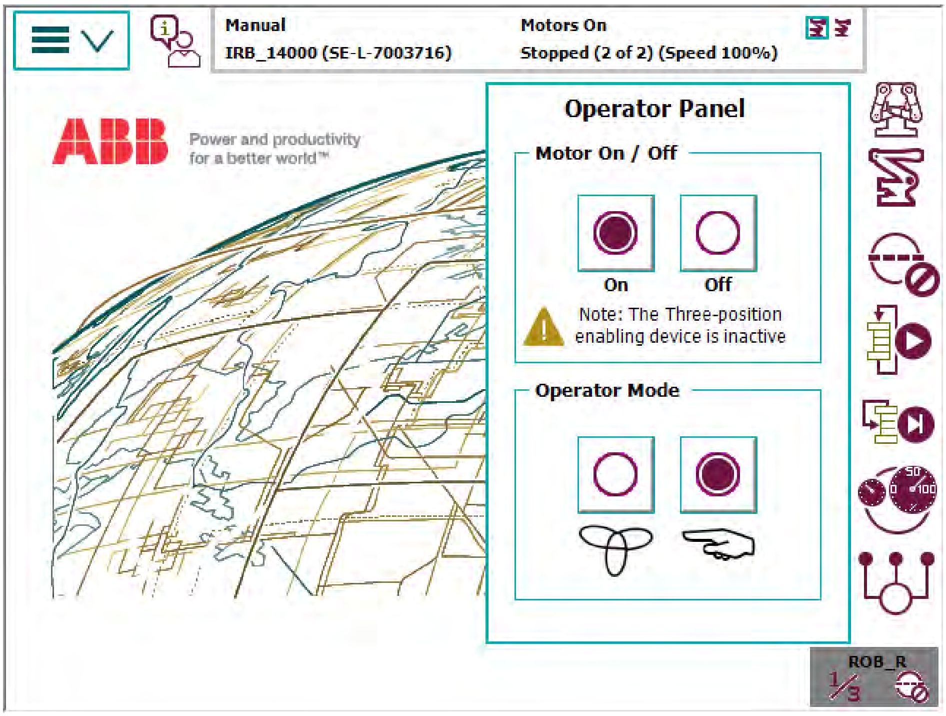
What is the automatic mode?

The automatic mode is the operating mode in which the robot control system operates in accordance with the task program, with functional safeguarding measures. This mode enables controlling the manipulator for example by using the I/O signals on the controller. An input signal may be used to start and stop a RAPID program, another to activate the motors on the manipulator.

Continues on next page

The operator panel

The operator panel is used to switch between *Manual mode* and *Automatic mode*. The operator panel is located under the Quickset menu, see *The Quickset menu on page 32*.



xx1500000005

> **Note**
>
> It is also possible to use the predefined I/O-signals to change and confirm the operating mode, see *I/O signals on page 45*.

3.5 Collision avoidance

Introduction

The IRB 14000 has a built in functionality called *Collision Avoidance* that is active both during jogging and when running programs. Collision avoidance monitors the geometrical models of the robot arms and the body, and the robot stops if any of the parts get too close to each other.

Collision Avoidance is no guarantee for avoiding collisions. It monitors the geometrical models of the robot arms and the body, but does not know the position of external equipment near to the robot.

CAUTION

The user must always be careful to avoid collisions with external equipment, since a collision could damage the mechanical structure of the arm.

System parameters

The default safety distance can be set by the system parameter *Coll-Pred Safety Distance*.

For more information, see the system parameter *Coll-Pred Safety Distance on page 60*.

False collision warning

If the robot arms move close to each other or close to the base, and the collision avoidance is too sensitive, a warning for collision may occur even if no collision will happen.

There are three different ways to lower the collision sensitivity to avoid warnings:

- Temporarily disable collision avoidance. See *Disabling collision avoidance on page 39*.
- Decrease the safety distance for the pair of robot arm links that trigger the false collision warning. See *Decrease sensitivity between links on page 40*.
- Decrease the general safety distance. See *Coll-Pred Safety Distance on page 60*.

Disabling collision avoidance

It is possible to disable the collision avoidance functionality. It may be necessary to temporarily disable collision avoidance if the robot has already collided or is within the default safety distance, or when the robot arms need to be very close and the risk of collision is acceptable.

Set the digital output signal `Collision_Avoidance` to 0 to disable collision detection. It is recommended to enable it (set `Collision_Avoidance` to 1) as soon as the work is done that required collision avoidance to be disabled.

Continues on next page

Decrease sensitivity between links

The sensitivity of collision avoidance can be decreased between individual robot arm links. This is useful if two links come close to each other, but the general safety distance should be maintained.

Open the file *irb_14000_common_config.xml* located in the folder *<SystemName>\PRODUCT\ROBOTWARE_6.XX.XXXX\robots\irb_14000*.

To decrease the safety distance between the left arm's link 3 and the right arm's link 4 to 1 mm, add the following row:

```
<Pair object1="ROB_L_Link3" object2="ROB_R_Link4"
      safetyDistance="0.001"/>
```

To decrease the safety distance between the left arm's link 5 and the robot base to 2 mm, add the following row:

```
<Pair object1="ROB_L_Link5" object2="Base" safetyDistance="0.002"/>
```

To disable collision avoidance between the left arm's link 2 and the right arm's link 3, add the following row:

```
<Pair object1="ROB_L_Link2" object2="ROB_R_Link3" exclude="true"/>
```

Note

The safety distance between two links can be decreased by adding a row to this XML file, but it cannot be increased to a higher value than defined by the system parameter *Coll-Pred Safety Distance*.

3.6 Collision

Overview

This section provides you an overview of the response of IRB 14000 robot system in the case of a collision.

Collision

If a collision is detected on an arm, the collided arm stops and the other arm keeps moving. But for a synchronized movement (`SyncMoveOn`), both arms stop in the case of a collision.

For more information about motion error handling see *Technical reference manual - RAPID kernel*.

Note

In RobotWare versions prior to 6.05 both arms stop in the case of a collision.

3.7 Programming and testing

Programming tools

You can use both the FlexPendant and RobotStudio for programming. The FlexPendant is best suited for modifying programs, such as positions and paths, while RobotStudio is preferred for more complex programming.

How to program using the FlexPendant is described in *Operating manual - IRC5 with FlexPendant*.

How to program using RobotStudio is described in *Operating manual - RobotStudio*.

Programming language

For more information about the RAPID language and structure, see *Technical reference manual - RAPID overview* and *Technical reference manual - RAPID Instructions, Functions and Data types*.

Programming Move-instructions

When programming a `jointtarget` for a 7-axis robot, the value of axis 7 (in degrees) is stored as the first external axis value in the `jointtarget` data.

Example:

```
! The value 135 is the angle of axis 7 in degrees:
CONST jointtarget myjointtarget := [[0,-130,30,0,40,0],
     [135,9E+09,9E+09,9E+09,9E+09,9E+09]];

MoveAbsJ myjointtarget \NoEOffs, v1000, fine, tool0;
```

When programming a `robtarget` for a 7-axis robot, the arm-angle is stored as the first external axis value in the `robtarget` data.

Example:

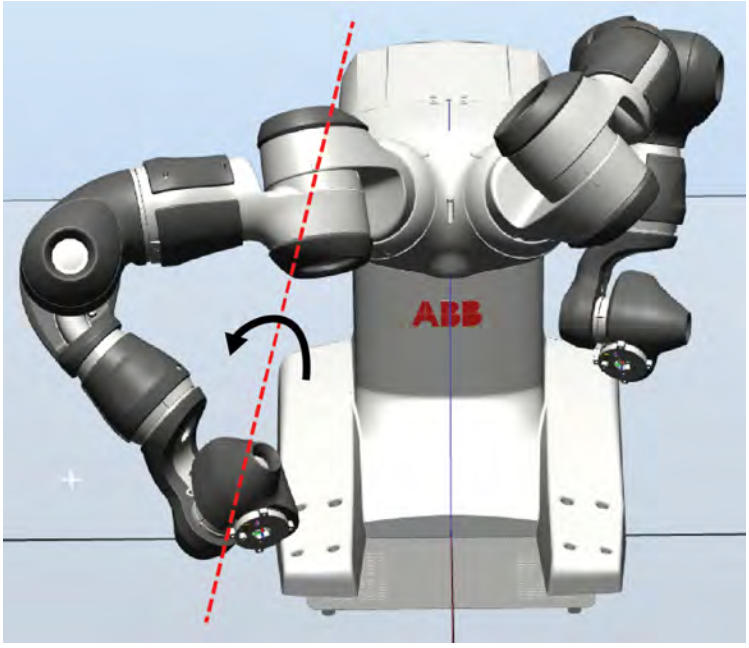
```
! The value 107.816 is the calculated arm-angle:
CONST robtarget myrobtarget :=
     [[-9,182,198],[0.0660087,0.84242,-0.111216,0.52307],[0,0,0,11],
     [107.816,9E+09,9E+09,9E+09,9E+09,9E+09]];

MoveL myrobtarget \NoEOffs, v1000, fine, tool0;
```

The arm-angle can, in a simplified way, be described as the angle of the arm seen from an axis that goes from the centre of axis 2 to the centre of the WCP (wrist

Continues on next page

center point). This value cannot be measured or calculated by the user since the underlying mathematical calculations are far more complex.



xx1600001072

Note

If necessary the arm-angle will be adjusted automatically during run time in order to avoid singularities and joint limits. The desired distance can be adjusted with the system parameter *Limit avoidance distance*, see *Limit avoidance distance on page 65*.

Coordinated programming using MultiMove

The IRB 14000 robot is pre-installed with the RobotWare option *MultiMove coordinated,* which makes it possible to program the two arms in coordinated mode.

For more information about MultiMove and coordinated jogging, see *Application manual - MultiMove*.

Tip

Use the MultiMove wizard in RobotStudio when setting up and programming MultiMove.

Configuration data

When programming linear movements it is important that the programmed positions have similar configurations, otherwise it will not be possible to move linearly between the positions.

This is important when programming all robots, but especially when programming 7-axis robots, since the *arm mode* adds more complexity.

Continues on next page

The data type `confdata` is used to define the configurations.

For more information about the data type `confdata`, see *Technical reference manual - RAPID Instructions, Functions and Data types*.

Contact applications

The RAPID instruction `ContactL` is designed to be used for contact applications, when the tool held by the robot has to press an object into place.

For more information, see *Technical reference manual - RAPID Instructions, Functions and Data types*.

3.8 I/O signals

Introduction

It is possible to connect different types of I/O signals to the IRB 14000, both digital I/O signals and different types of fieldbuses (industrial networks).

For more information about connecting I/O signals, see *Product manual - Product.ProductName* and *Circuit diagram - IRB 14000*.

Predefined signals

Operating mode signals

The following output signals are predefined in the system and can be used to change and confirm the operating mode.

Name	Type	Description
`VP_ENABLE`	output	Enable signal for manual mode.
`VP_MODEKEY`	output	Operating mode selector.
`VP_MOTOPB`	output	Motors On push button.

Collision avoidance signal

Name	Type	Description
`Collision_Avoidance`	output	Default is 1, meaning that collision avoidance is enabled. Setting this signal to 0 will disable collision avoidance. This is usable for tasks where you need the arms to be very close and the risk of collision is acceptable.

3.9 User authorization

Introduction

The data, functionality, and commands on a controller are protected by a User Authorization system (also called UAS). The UAS restricts the parts of the system the user has access to. Different users can have different access grants.

It is recommended to create different user groups for different types of users. For example *Operator*, *Engineer*, and *Service*. Operators should have very limited access to the system.

Note

Only users that are authorized to to modify safety functions should have access to the grants *Restore a backup* and *Modify configuration*.

For more information about configuring the User Authorization system and the different controller grants, see *Operating manual - RobotStudio*.

Changing safety related system parameters

When changing the safety related system parameters *Arm Check Point Speed Limit* or *Global Speed Limit* an event message will take focus on the FlexPendant after restart to notify the user of the change.

For more information about the system parameters, see *System parameters on page 57*.

4 Calibration

4.1 Introduction

General

This chapter includes information about when the robot system must be recalibrated. There are two types of calibration, to update the revolution counters or to do a fine calibration.

When to update the revolution counters

If the revolution counter memory is lost, the counters must be updated. This will occur when:

- The battery is discharged
- A resolver error occurs
- The signal between a resolver and measurement board is interrupted
- A robot axis is moved with the control system disconnected

The revolution counters must also be updated after the robot and controller are connected at the first installation.

To update the revolution counters is a simple procedure which can be performed by the operator, see *Updating revolution counters on page 50*.

When to do a fine calibration

The system must be fine calibrated when parts affecting the calibration position are replaced on the robot, for example motors or parts of the transmission.

A fine calibration should only be performed by a qualified service engineer. For more information see section *Calibration* in *Product manual - Product.ProductName*.

4.2 Calibration scale and correct axis position

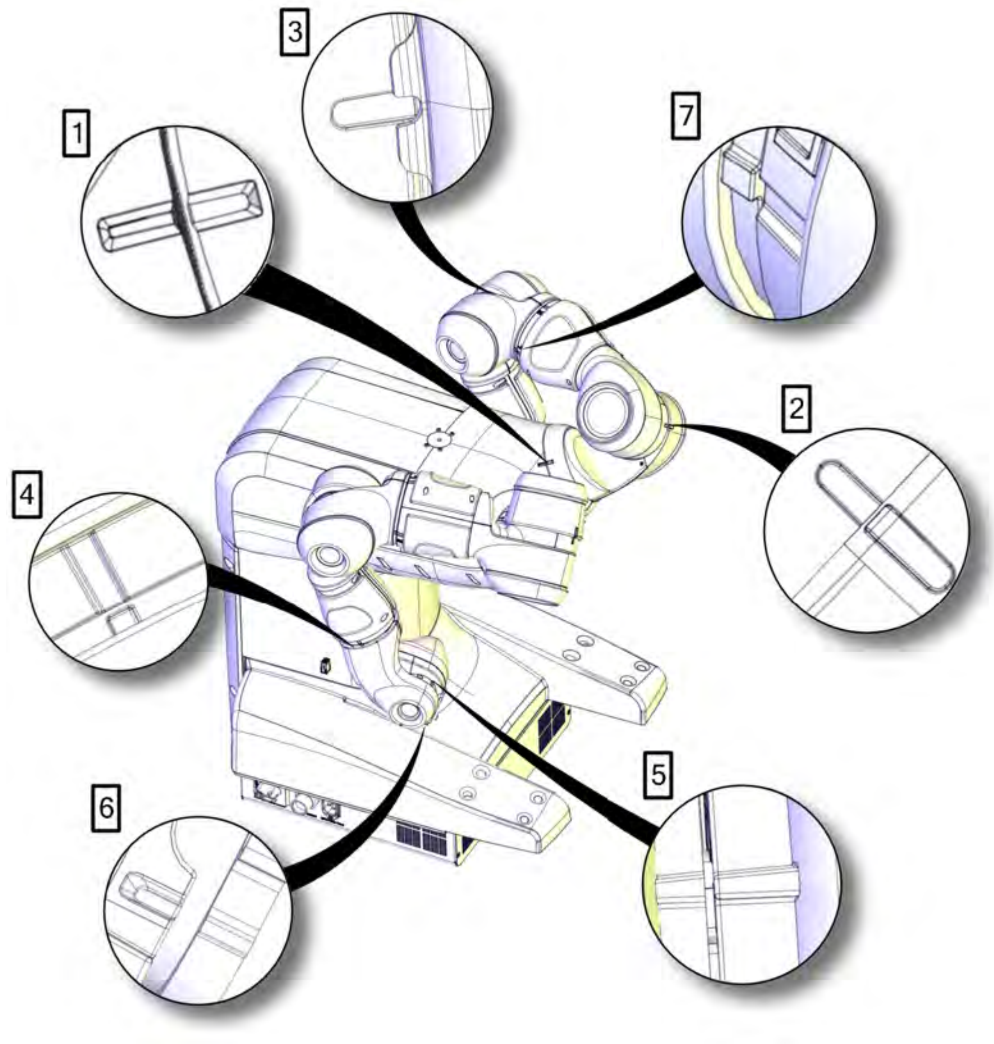
Introduction

This section specifies the calibration scale positions and/or correct axis positions.

Calibration scales/marks

This illustration shows the positions of the calibration scales and marks on the robot.

The number aside of the enlargement corresponds to the axis number.



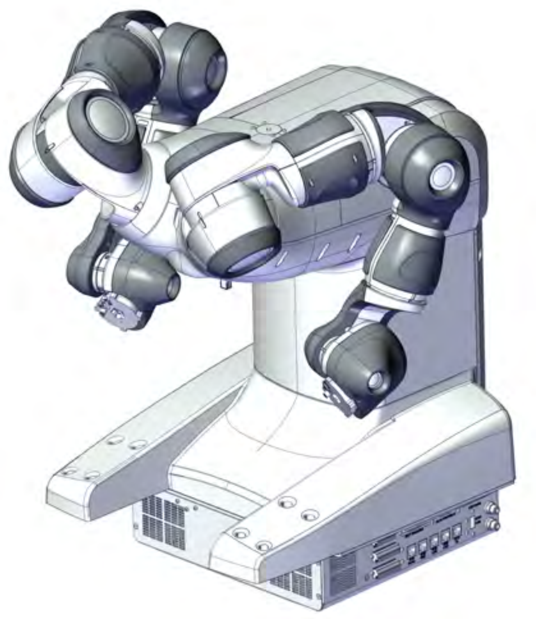
xx1500000526

Continues on next page

Calibration position

Illustration of robot in calibration position

The figure shows the robot in its calibration position.



xx1500000363

Exact axis positions in degrees

The table below specifies the exact axis positions in degrees.

Axis	IRB 14000 ROB_R	IRB 14000 ROB_L
1	0°	0°
2	-130°	-130°
3	30°	30°
4	0°	0°
5	40°	40°
6	0°	0°
7	-135°	135°

4.3 Updating revolution counters

Introduction

This section describes how to do a rough calibration of each robot axis, which updates the revolution counter value for each axis using the FlexPendant.

The procedure can be summarized accordingly:

1. Manually move the manipulator to the calibration position.
2. Select the Calibration with hall sensors (CalHall) routine.
3. Select the function Update of revolution counters.
4. Store the revolution counter setting.

Each step is described in detail in following sections.

Step 1 - Manually moving the manipulator to the calibration position

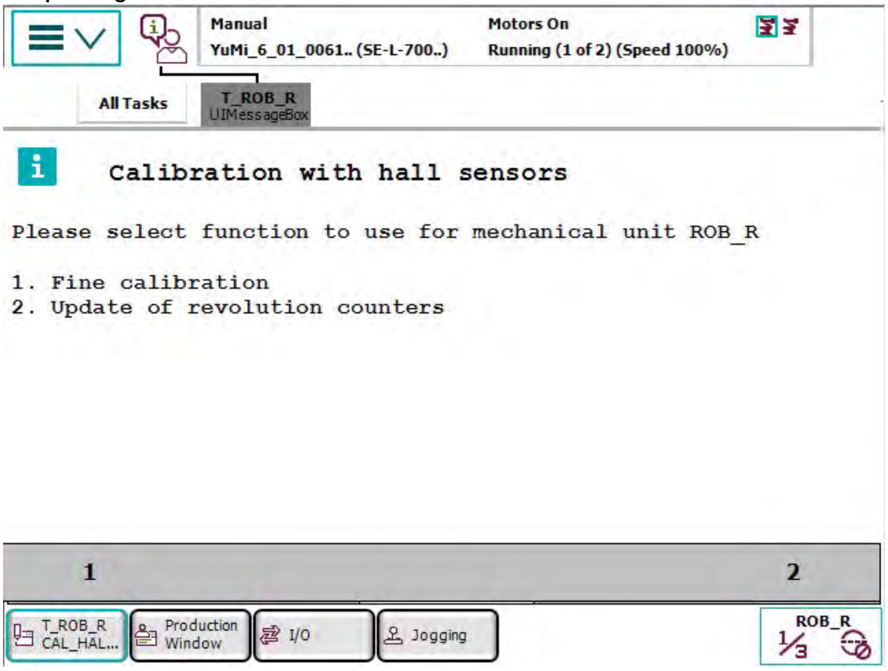
	Action	Note
1	**CAUTION** When releasing the holding brakes, the robot axes may move very quickly and sometimes in unexpected ways!	
2	Release the brakes of the robot arm to be calibrated and move the arm manually so that the synchronization mark of each joint is aligned. The robot now stands in its calibration position.	The synchronization marks are shown in *Calibration scale and correct axis position on page 48*. There is a tolerance for the joint position. The edge of a mark should be at least within the area of the opposite mark.

Step 2 - Selecting the Calibration with hall sensors (CalHall) routine

	Action	Note
1	Open the **Program Editor** on the FlexPendant.	
2	Select the task that corresponds to the robot arm to be calibrated. Tap **Open**.	
3	If necessary, create a new program. This needs to be done if no existing program is available.	
4	Select **Debug** and tap **PP to Main**.	
5	Select **Debug** and tap **Call Routine...**	
6	Select **CalHall**.	
7	Go to **Motor On** and press the **Start button**.	

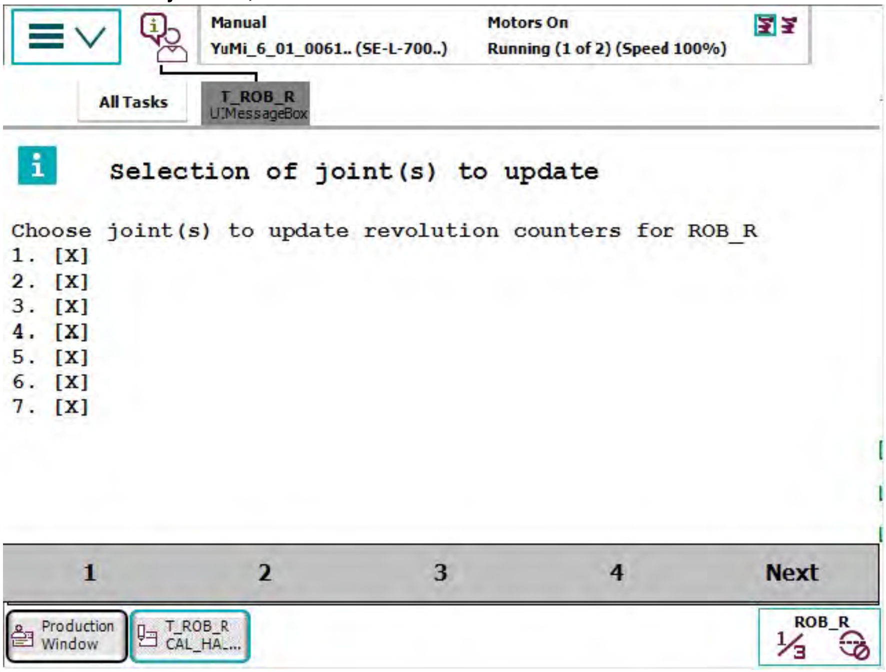
Continues on next page

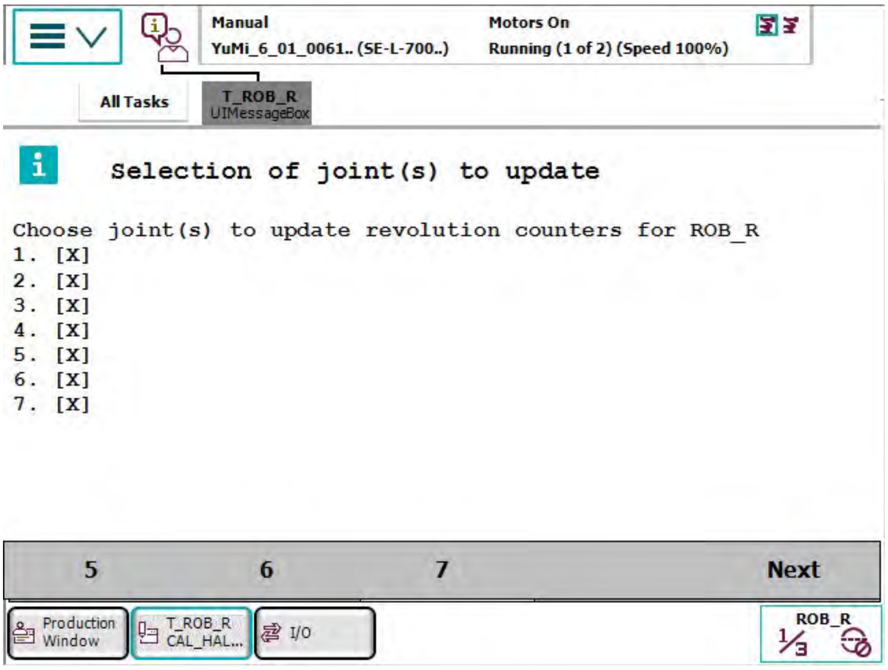
Step 3 - Selecting the function Update of revolution counters

<table>
<tr><th></th><th>Action</th></tr>
<tr><td>1</td><td>In the calibration with hall sensors routine (CalHall), tap 2 to select the function of updating the revolution counters.


xx1400002694</td></tr>
</table>

Continues on next page

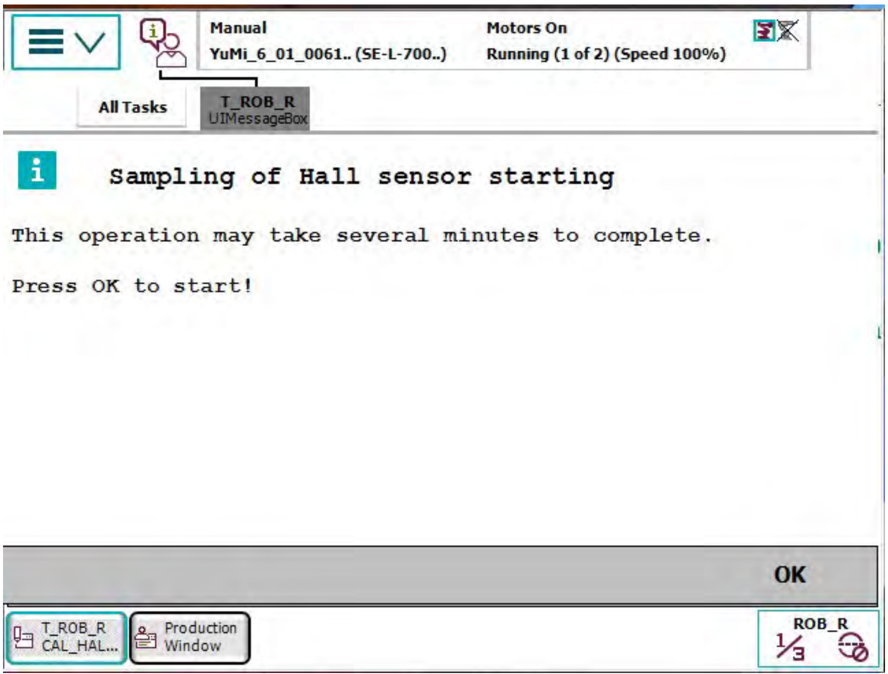
<table>
<tr><th></th><th>Action</th></tr>
<tr><td>2</td><td>Tap to select which joint(s) to update revolution counters for.
Joints 1, 2, 3 and 4 are selectable in the first window. Tap Next to open the second window where joints 5, 6 and 7 are selectable.


xx1400002695


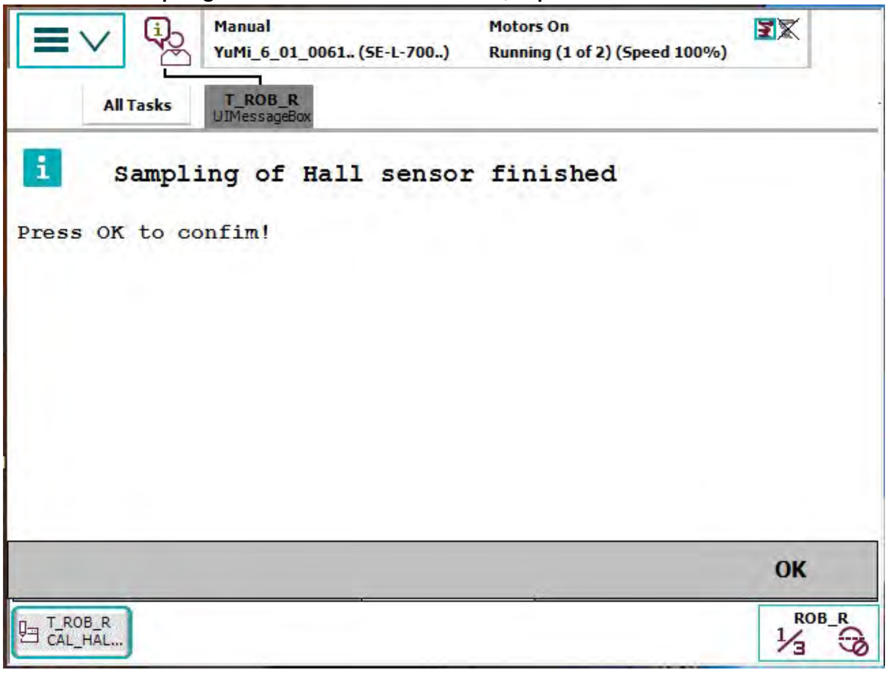
xx1400002696</td></tr>
</table>

Continues on next page

Step 4 - Storing the revolution counter setting

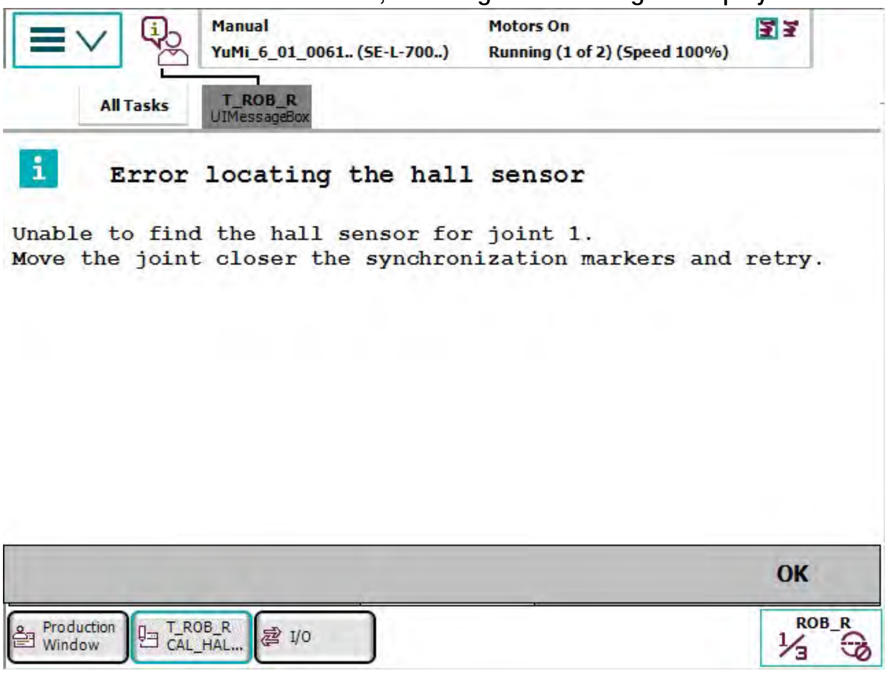
	Action
1	Tap **OK** to start the sampling procedure of the hall sensor. One at a time, each selected joint now rotates back and forth several times to calculate and find the exact position where the hall sensor is aligned with the magnet. Calibration order: axis 1-2-3-4-5-6-7. **Note** This procedure may take several minutes.  xx1400002697

Continues on next page

	Action
2	When the sampling of hall sensors is finished, tap **OK** to confirm.  xx1400002898

Continues on next page

<table>
<tr><th></th><th>Action</th></tr>
<tr><td>3</td><td>
If the hall sensor can not be found, following error message is displayed.



xx1400002698

There can be several causes for this error.

- The joints are not sufficiently aligned according to the synchronization marks. Move the joints so that the synchronization marks are aligned. Tap **OK** and restart the fine calibration procedure. The joints prior to the joint that stopped the calibration procedure are calibrated and do not need to be calibrated again. Joints coming after the joint that stopped the calibration procedure has not been calibrated. (Calibration order: axis 1-2-3-4-5-6-7.)
- The hall sensor is defect. Perform troubleshooting of I/O signals. Possible replacement of hall sensor may be required.
</td></tr>
</table>

4.4 Verifying the calibration position

Introduction

Verify the calibration position of the robot before beginning any programming of the robot system. This may be done:

- Using a `MoveAbsJ` instruction with argument according to calibration position degrees on all axes.
- Using the **Jogging** window on the FlexPendant.

Using a `MoveAbsJ` instruction

Use this procedure to create a program that runs all the robot axes to their calibration position.

	Action	Note
1	On ABB menu tap **Program editor**.	
2	Create a new program.	
3	Use **MoveAbsJ** in the **Motion&Proc** menu.	

4	Create the following program for the right arm:
	`MoveAbsJ [[0,-130,30,0,40,0], [-135,9E9,9E9,9E9,9E9,9E9]] \NoEOffs, v1000, fine, tool0;`
	Create the following program for the left arm:
	`MoveAbsJ [[0,-130,30,0,40,0], [135,9E9,9E9,9E9,9E9,9E9]] \NoEOffs, v1000, fine, tool0;`	

5	Run the program in manual mode.	
6	Verify that the calibration marks for the axes align correctly. If they do not, then update the revolution counters.	See *Calibration scale and correct axis position on page 48* and *Updating revolution counters on page 50*.

Using the jogging window

Use this procedure to jog the robot to the calibration position for all axes.

	Action	Note
1	On the **ABB** menu, tap **Jogging**.	
2	Tap **Motion mode** to select group of axes to jog.	
3	Tap to select the axis to jog, axis 1, 2, or 3.	
4	Manually run the robots axes to a position where the axis position value read on the FlexPendant, is equal to the calibration position degrees.	Degrees are specified in *Exact axis positions in degrees on page 49*.
5	Verify that the calibration marks for the axes align correctly. If they do not, then update the revolution counters!	See *Calibration scale and correct axis position on page 48* and *Updating revolution counters on page 50*.

5 System parameters

5.1 Introduction

About the system parameters

This section describes the IRB 14000 specific system parameters. The parameters are divided into the topic and type they belong to.

For information about other system parameters, see *Technical reference manual - System parameters*.

Topic I/O System

Parameter	For more information, see
Collision Avoidance	*Collision Avoidance on page 58*
Connection Timeout Multiplier	*Connection Timeout Multiplier on page 59*

Topic Motion

Parameter	For more information, see
Coll-Pred Safety Distance	*Coll-Pred Safety Distance on page 60*
Global Speed Limit	*Global Speed Limit on page 61*
Arm Check Point Speed Limit	*Arm Check Point Speed Limit on page 62*
Arm-Angle Definition	*Arm-Angle Definition on page 63*
Arm-Angle Reference Direction	*Arm-Angle Reference Direction on page 64*
Limit avoidance distance	*Limit avoidance distance on page 65*
Collision Detection Memory	*Collision Detection Memory on page 66*
Friction compensation lead through factor	*Friction compensation lead through factor on page 67*

5.2 Topic I/O System

5.2.1 Collision Avoidance

Parent

Collision Avoidance is an action value for the parameter *Action* that belongs to the type *System Input* in the topic *I/O System*.

Cfg name

CollAvoidance

Description

This parameter is currently applicable only to IRB 14000 (YuMi robot).

The action value *Collision Avoidance* is used to activate the Collision Avoidance functionality. By default this functionality is off in robots.

Collision Avoidance monitors a detailed geometric model of the robot. If two bodies of the model come too close to each other, the controller warns about a predicted collision and stops the robot. The system parameter *Coll-Pred Safety Distance* determines at what distance the two objects are considered to be in collision, see Coll-Pred Safety Distance.

5.2.2 Connection Timeout Multiplier

Parent

Connection Timeout Multiplier belongs to the type *Device*, in the topic *I/O System*.

Cfg name

`ConnectionTimeoutMultiplier`

Description

Connection Timeout Multiplier specifies the multiplier applied to the expected packet rate value to derive the value for the Inactivity/Watchdog Timer.

Usage

The *Connection Timeout Multiplier* is a number among 4, 8, 16, 32, 64, 128, 256. It is used together with RPI to calculate the timeout on connections. *RPI* multiplied by *Connection Timeout Multiplier* gives the maximum time before dropping the connection.

Note

For the IRB 14000 robot (YuMi) this parameter may have to be tuned depending on your network setup.

Prerequisites

The option *EtherNet/IP Scanner/Adapter* must be installed.

Allowed values

Allowed values are 4, 8, 16, 32, 64, 128, 256, 512.

Default value is 4.

5.3 Topic Motion

5.3.1 Coll-Pred Safety Distance

Parent

Coll-Pred Safety Distance belongs to the type *Motion System*, in the topic *Motion*.

Cfg name

coll_pred_default_safety_distance

Description

The parameter *Coll-Pred Safety Distance* determines at what distance two geometric objects (for example robot-links) are considered to be in collision.

Allowed values

A value between 0.001 to 1 meter.

Default value is 0.01 meter.

Related information

Collision Avoidance on page 58

5.3.2 Global Speed Limit

Parent

Global Speed Limit belongs to the type *Robot*, in the topic *Motion*.

Cfg name

Global_max_speed_limit_custom

Description

Global Speed Limit sets the speed limit in meters per second for the tool center point (TCP), the arm check point (ACP), and the wrist center point (WCP).

Note

This parameter is used to configure the safety function Cartesian speed supervision.

Note

When changing this safety related system parameter, an event message will take focus on the FlexPendant after restart to notify the user of the change. The user then has to verify that the intended setting was made.

Limitations

Global Speed Limit is only used for the following robots:

- IRB 14000

Setting this parameter for any other robot will not have any effect.

Global Speed Limit can only be used to lower the speed limit from maximum speed limit for each robot type. If a higher value is set, the maximum value for the robot type is used.

The maximum value for the robot types are:

Robot type	Maximum value
IRB 14000	1.5 m/s

Allowed values

A number between 0.1 and 20.

The default value is 20.

5.3.3 Arm Check Point Speed Limit

Parent

Arm Check Point Speed Limit belongs to the type *Robot*, in the topic *Motion*.

Cfg name

Global_max_speed_limit_acp_custom

Description

Arm Check Point Speed Limit sets the speed limit in meter per second for the arm check point (ACP).

Note

This parameter is used to configure the safety function Cartesian speed supervision.

Note

When changing this safety related system parameter, an event message will take focus on the FlexPendant after restart to notify the user of the change. The user then has to verify that the intended setting was made.

Limitations

Arm Check Point Speed Limit is only used for the following robots:

- IRB 14000

Setting this parameter for any other robot will not have any effect.

Arm Check Point Speed Limit can only be used to lower the speed limit from a maximum speed limit for each robot type. If a higher value is set, the maximum value for the robot type is used.

The maximum value for the robot types are:

Robot type	Maximum value
IRB 14000	0.75 m/s

Allowed values

A number between 0.1 and 20.

The default value is 0.75.

5.3.4 Arm-Angle Definition

Parent

Arm-Angle Definition belongs to the type *Robot*, in the topic *Motion*.

Cfg name

use_old_arm_angle_definition

Description

To completely specify the pose for a robot with 7 axes, an additional parameter called arm-angle is needed.

The parameter *Arm-Angle Definition* controls how the arm-angle is defined.

Users are advised to always use the new arm-angle definition. The old definition is kept only for backwards compatibility and can in some cases lead to non-optimal movements of the robot.

Limitations

Arm-Angle Definition is only applicable for 7-axis robots.

Note

Arm-Angle Definition parameter is not supported in RW 6.07 or later versions.

Allowed values

New or Old.

The default value is New.

Related information

Operating manual - IRB 14000

5.3.5 Arm-Angle Reference Direction

Parent

Arm-Angle Reference Direction belongs to the type *Robot*, in the topic *Motion*.

Cfg name

arm_angle_ref_dir

Description

Arm-Angle Reference Direction controls how the arm-angle property is calculated and affects the location of certain singularities for seven-axis robots.

Usage

In addition to position and orientation, seven-axis robots also depend on the arm-angle concept to fully specify a `robtarget`.

The calculation of the arm-angle depends on a chosen reference direction, and by default this reference direction is chosen as the line passing through axis 2 origin of the robot and being parallel with the Y-axis of the world frame. When the WCP is on the axis chosen as the reference direction, the arm-angle becomes undefined. Hence, the inverse kinematics is singular for all positions with the WCP on the line, and linear movement on and across this line will not work.

If linear movement in this area of the workspace is important for your application, then you can configure the robot to use another reference direction. The choices available are: the world Y-axis, the world Z-axis, and the line passing through axis 1 of the robot.

Note

A RAPID program created with one value for this parameter will behave differently or maybe not work at all if the parameter value is changed.

Allowed values

Arm-Angle Reference Direction can have the following values:

Value:	Name:	Description:
0	World Y	Reference direction parallel with the Y-axis of the world frame.
1	World Z	Reference direction parallel with the Z-axis of the world frame.
2	Axis 1	Reference direction parallel with a line passing through axis 1 of the robot.

The default value is 0.

Related information

Operating manual - IRB 14000

5.3.6 Limit avoidance distance

Parent

Limit avoidance distance belongs to the type *Robot*, in the topic *Motion*.

Cfg name

limit_avoidance_distance

Description

Limit avoidance distance controls the distance to the nearest singularity or joint limit when automatically adjusting the arm-angle.

Usage

The singularities that can be handled are where axis 2 or axis 5 is equal to zero.

Allowed values

A value between -1 to 100 radians.

The default value is 0.017453 radians.

Setting a negative value will disable the functionality.

Related information

Operating manual - IRB 14000

5.3.7 Collision Detection Memory

Parent

Collision Detection Memory belongs to the type *Motion Supervision*, in the topic *Motion*.

Cfg name

collision_detection_memory

Description

Collision Detection Memory defines how much the robot moves back on the path after a collision.

The parameter requires a restart of the controller when modified.

Usage

The movement of robot back on the path after a collision is specified in seconds. If the robot was moving quickly before the collision, it will move further back than if the speed was lower. For detailed information, see the *Application manual - Controller software IRC5*.

Allowed values

A value in the interval 0 to 0.5, specifying the movement in seconds.

For the IRB 14000 robot the default value is 0 s and hence the robot does not back off.

Setting the value to 0 s (disabling backing after collision) may leave the robot in a state with residual forces remaining after a collision. This could trigger new collisions when trying to move away from that position. To move away robustly after a collision, the following are some of the recommended solutions:

- Enable lead-through for a short period of time to release the tension.
- Set the value of `MotionSup\` to `Off` before executing the `move` instructions.
- Use `ContactL` instead of `MoveL`.

Related information

Refer the section "How to tune the motion supervision" in the manual *Technical reference manual - System parameters*.

Application manual - Controller software IRC5.

5.3.8 Friction compensation lead through factor

Parent

Friction compensation lead through factor belongs to the type *Robot*, in the topic *Motion*.

Cfg name

friction_comp_lead_through_factor

Description

Friction compensation lead through factor determines how soft a robot should be in lead through mode.

Usage

A higher value makes the robot softer in lead through mode and a lower value makes the robot less soft.

Setting a high value can make the robot sensitive to errors such as wrong payload in the tool definition. The robot can then start to drift by itself.

Setting the value to 0 removes all friction compensation in lead through mode.

> **Note**
>
> This parameter does not need a reboot to apply the changes. Hence the tests of different levels can be done directly after changing the parameter value.

Limitations

Friction compensation lead through factor is only used for the following robots:

- IRB 14000

Configuring this parameter in any other robot will not have any effect.

Allowed values

A value between 0.0 and 1.0.

Default value is 0.6.

This page is intentionally left blank

Index

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

abb.com/robotics

3HAC052986-001, Rev E, en